INVESTIGACIÓN DE LOS MÉTODOS ÁGILES PARA SU IMPLANTACIÓN EN UN LABORATORIO DE SEGURIDAD ELÉCTRICA

TESIS FINAL DE MASTER.

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Titular: Jose Javier Rodríguez Álvarez

Titulación: Máster en Ingeniería Electrónica (MEE)

Título: Investigación de los métodos ágiles para su implantación en un laboratorio de seguridad eléctrica

Director de la Tesis: Jaume Mussons Sellés

Departamento: Organización de empresas

Diciembre de 2013

Índice de capítulos

1. INTRODUCCION 7

1.1. OBJETIVO DE LA TESIS Y CLARIFICACIÓN DE LOS CONTENIDOS DE LA MISMA 7

1.2. DEFINICIÓN DE DESARROLLO ÁGIL DE PROYECTOS (PROYECTOS ÁGILES) 8

1.3. BREVE HISTORIA DE LOS MÉTODOS ÁGILES 9

1.4. IMPORTANCIA Y CONTENIDO DEL EXAMEN PMI-ACP 9

1.4.1. Requisitos para poder presentarse al examen 10

1.4.2. Supuestos de examen 11

1.4.3. Contenido del examen 11

2. ESTUDIO TEÓRICO DE LOS MÉTODOS ÁGILES 15

2.1. ESTRUCTURA ÁGIL 15

2.1.1. Justificación de la necesidad de los métodos ágiles 15

2.1.2. El manifiesto ágil 16

2.1.3. Principios guía 19

2.1.4. La declaración de interdependencia 26

2.1.5. Métodos Ágiles 29

2.1.5.1. Scrum 29

2.1.5.2. Extreme Programming (XP) 33

2.1.5.3. Desarrollo basado en las características (FDD) 37

2.1.5.4. Método de desarrollo dinámico de sistemas (DSDM) 38

2.1.5.5. Crystal 39

2.1.5.6. Desarrollo de software Lean 41

2.1.5.6. Desarrollo Kanban 42

2.1.6. Proceso de adaptación 44

2.1.6.1. Guías del proceso de aprendizaje 44

2.1.7. Ética y responsabilidad profesional 45

2.2. ENTREGAS BASADAS EN EL VALOR 47

2.2.1. Evaluación del valor 50

2.2.1.1. Rentabilidad de la inversión (ROI) 50

2.2.1.2. Valor actual neto 54

2.2.1.3. Tasa Interna de Rentabilidad (TIR) 55
2.2.1.4. Ejemplos de cálculo del VAN y de la TIR 55
2.2.2. Planificación del valor 59
2.2.2.1. Chartering 59
2.2.2.2. Mapa de flujo de valor 60
2.2.2.3. Priorización basada en el valor para cliente 65
2.2.2.4. Priorización relativa / Clasificación 68
2.2.2.5. Hoja de Ruta de producto 70
2.2.2.6. Cartera Ajustada al riesgo 71
2.2.2.7. Contratación ágil 76
2.2.3. Entregando valor 81
2.2.3.1. Tablas de tareas y Kanban 82
2.2.3.2. Límites de WIP 84
2.2.3.4. Entrega incremental 88
2.2.4. Confirmación de valor 88
2.2.4.1. Priorización basada en el valor para el cliente 89
2.2.4.2. Prototipos, simulaciones, demostraciones 90
2.2.5. Siguiendo y reportando el valor 91
2.2.5.1. Valor ganado ágil 91
2.2.5.2. Diagramas de Flujo Acumulado (CFDs) 94
2.2.5.3. Gráficos Risk Burn Down 99
2.2.5.4. Tablas de tareas/Kanban 104
2.3. IMPLICACIÓN DE LOS STAKEHOLDERS 107
2.3.1. Alineando la comprensión de los Stakeholders 109
2.3.1.1. Diagramas de estructura 110
2.3.1.2. “Personas” 110
2.3.1.3. Historias de usuario 112
2.3.1.4. Mapas de historias 117
2.3.1.5. Incorporando los valores de los Stakeholders 117
2.3.1.6. Gestión de los Stakeholders 117
2.3.1.7. Gestión de los proveedores 120
2.3.2. Comunicándose con los Stakeholders 121
2.3.2.1. Gestión de las comunicaciones 122
2.3.2.2. Radiadores de información 123

2.3.2.3. Gráficos “Burn down y Burn up” 123
2.3.2.4. Velocidad 125
2.3.2.5. Modelado ágil 126
2.3.3. Utilizando habilidades sociales criticas 126
2.3.3.1. Negociación 127
2.3.3.2. Escucha activa 128
2.3.3.3. Métodos de facilitación 130
2.3.3.4. Globalización, cultura y diversidad de equipos 131
2.3.3.5. Resolución de conflictos 132
2.3.3.6. Equipos distribuidos 136
2.3.3.7. Modelos de decisión participativos 136
2.3.4. Dirigir eficazmente 139
2.3.4.1. Liderazgo de servicio 140
2.3.4.2. Herramientas y técnicas de liderazgo 144
2.3.5. Ética y responsabilidad profesional 146
2.4. IMPULSAR LAS PRÁCTICAS DE RENDIMIENTO DEL EQUIPO 150
2.4.1. Comprender el rendimiento del equipo 150
2.4.1.1. Liderazgo adaptativo 152
2.4.1.2. Inteligencia emocional 155
2.4.1.3. Crear equipos capaces 157
2.4.1.4. Creando equipo de alto rendimiento 159
2.4.2. Prácticas de equipo 163
2.4.2.1. Reuniones diarias 163
2.4.2.2. Formar y asesorar 164
2.4.2.3. Técnicas de Brainstorming 165
2.4.2.4. Espacio del equipo 170
2.4.2.5. Equipos coubicados 170
2.4.2.6. Equipos distribuidos 172
2.4.2.7. Herramientas ágiles 175
2.4.3. Ética y responsabilidad profesional 177
2.5. PLANIFICACIÓN ADAPTATIVA 179
2.5.1. Planificación de conceptos 181
2.5.1.1. Timeboxing 182
2.5.1.2. Elaboración progresiva 183

2.5.1.3. Proceso de adaptación ... 185
2.5.1.4. Característica mínima comercializable (MMF) ... 185
2.5.1.5. Análisis basado en el valor ... 186
2.5.1.6. Descomposición y priorización basada en el valor ... 187
2.5.1.7. Juegos ágiles (juegos colaborativos y de innovación) ... 189
2.5.2. Estimación ... 193
2.5.2.1. Wideband Delphi y planificación póker ... 194
2.5.2.2. Tiempo ideal ... 197
2.5.2.3. Tamaño relativo / punto de historia ... 198
2.5.2.4. Estimación afín ... 202
2.5.2.5. Tiempo, presupuesto, y estimación de coste (vía tamaño, esfuerzo, y estimación de coste) ... 202
2.5.2.6. Principios de contabilidad en proyectos ágiles ... 205
2.5.3. Planes ágiles ... 208
2.5.3.1. Cartas ágiles ... 214
2.5.3.2. Caso de desarrollo de negocio ... 216
2.5.3.3. Planificación de la iteración y la entrega ... 218
2.6. DETECCIÓN Y RESOLUCIÓN DE PROBLEMAS ... 225
2.6.1. Identificar problemas ... 227
2.6.1.1. Tiempo de ciclo ... 227
2.6.1.2. Defectos escapados ... 230
2.6.1.3. Normas de proyecto y Calidad ... 232
2.6.1.4. Modos de fallo y alternativas ... 233
2.6.1.5. Análisis de varianza y tendencia ... 237
2.6.1.6. Límites de control ... 240
2.6.2. Resolviendo problemas ... 241
2.6.2.1. Integración continua ... 241
2.6.2.2. Puntas basadas en riesgo ... 243
2.6.2.3. Verificación y validación frecuentes ... 244
2.6.2.4. Desarrollo basado en el test (TDD) / Desarrollo basado en el primer test (TFD) ... 246
2.6.2.5. Aceptación del desarrollo basado en test (ATDD) ... 248
2.6.2.6. Resolviendo problemas ... 249
2.6.3. ¿Por qué centrarse en comprometer al equipo? ... 255
2.6.4. Ética y responsabilidad profesional ... 256

2.7. MEJORA CONTINUA 259
2.7.1. Prácticas de mejora continua 259
2.7.1.1. Retrospectivas 260
2.7.1.2. Compartir el conocimiento 265
2.7.1.3. Proceso de adaptación 268
2.7.1.4. Principios de sistemas del pensamiento 270
2.7.1.5. Análisis de procesos 271
2.7.1.6. Aplicar las nuevas Prácticas Ágiles 273
2.7.1.7. Código de conducta ética y profesional PMI 275
2.7.1.8. Procesos continuos de mejora 275
2.7.1.9. Autoevaluación 277
3. CASO PRÁCTICO 280
3.1. OBJETIVOS DEL CASO PRÁCTICO 280
3.2. VISIÓN GENERAL DE LA EMPRESA CASO DE ESTUDIO 280
3.2.1. Historia del Laboratori General d'Assaigs i Investigacions 281
3.2.2. Descripción de la actividad de la empresa objeto del caso de estudio 284
3.3. DESCRIPCIÓN DE LA ORGANIZACIÓN ACTUAL EN EL CENTRO DE SEGURIDAD ELÉCTRICA 288
3.3.1. Descripción de la organización de trabajo 289
3.3.2. Funciones de los diferentes puestos 290
3.3.2.1. Área comercial. 290
3.3.2.2. Director de operaciones 290
3.3.2.3. Product manager 290
3.3.2.4. Test manager 291
3.3.2.5. Técnicos de Laboratorio. 291
3.3.3. Flujo de trabajo 291
3.3.4. Tareas adyacentes al flujo de trabajo 294
3.4. IMPLANTACIÓN DE METODOLOGÍA ÁGIL 296
3.4.1. Trabajadores con el conocimiento para realizar los trabajos 296
3.4.2. Liderazgo de servicio 297
3.4.3. División del proyecto en iteraciones 299
3.4.3.1 Partes del proyecto y su importancia 299
3.4.3.2 División del proyecto en iteraciones 306
3.4.4. Reuniones diarias y por iteración 308
3.4.4.1. Reunión diaria 308

3.4.4.2. Reunión de final de iteración 309
3.4.4.3. Reunión mensual 309
3.4.4.3. Reunión inicial de proyecto 309
3.4.5. Implicación de los Stakeholders 310
3.4.5.1. Stakeholders 310
3.4.5.2. Técnicas para lograr la implicación de los Stakeholders 313
3.4.6. Técnicas para controlar el avance del proyecto 316
3.4.7. Resolución de conflictos 322
3.4.8. Ubicación de los equipos 322
3.5. GESTIÓN DEL LIDERAZGO Y DEL RENDIMIENTO DE LOS EQUIPOS 323
3.5.1. Liderazgo de servicio 323
3.5.2. Herramientas y técnicas de liderazgo 325
3.5.3. Rendimiento del equipo 326
3.5.3.1. Prácticas para aumentar la efectividad del equipo 328
4. CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN 331
4.1. CONCLUSIONES 331
4.2. LÍNEAS FUTURAS DE INVESTIGACIÓN 333
5. BIBLIOGRAFÍA 334
5.1 Libros consultados 334
5.2 Artículos consultados 334
5.3 Páginas web consultadas 335
6. ANEXO DE EXÁMENES PRÁCTICOS 336
6.1. Examen práctico capítulo 2.1: Estructura ágil 336
6.2. Examen práctico capítulo 2.2: Entregas basadas en el valor 342
6.3. Examen práctico capítulo 2.3: Implicación de los Stakeholders 350
6.4. Examen práctico capítulo 2.4: Impulsar las prácticas de rendimiento del equipo 357
6.5. Examen práctico capítulo 2.5: Planificación adaptativa 364
6.6. Examen práctico capítulo 2.6: Detección y resolución de problemas 371
6.7. Examen práctico capítulo 2.7: Mejora continua 377

1. INTRODUCCION

1.1. OBJETIVO DE LA TESIS Y CLARIFICACIÓN DE LOS CONTENIDOS DE LA MISMA

El objetivo del desarrollo de la presente Tesis Final de Master es la investigación y estudio de las prácticas ágiles para lograr su implantación en un entorno de gestión de proyectos que no las utiliza, en este caso un laboratorio de seguridad eléctrica.

Al mismo tiempo, se busca que el estudio llevado a cabo en la sección 2 de la Tesis Final de Máster, pueda utilizarse para preparar el examen para la Certificación ACP (Agile Certified Practitioner) del Project Management Institute (PMI).

En la sección 3, se detallará el caso práctico realizado, en el que se implantará la metodología ágil en un ambiente de trabajo que no la utiliza, de manera que se pueda comprobar cómo afectaría el uso de estas técnicas en el desarrollo de los proyectos. En primer lugar se realizará un estudio del funcionamiento actual de la empresa caso del estudio, viendo en que partes se podrían implantar los métodos ágiles. Se verá que técnicas de los métodos ágiles pueden ayudar a mejorar aún más la eficiencia del trabajo en general y conseguir aumentar la productividad y la calidad del trabajo realizado. A continuación se expondrá una amplia visión de la empresa, describiendo su historia y centrándonos en describir la actividad objeto del estudio, de manera que pueda establecerse una amplia comprensión.

En la sección 4 se detallarán las conclusiones generales de la Tesis Final de Máster y se propondrán líneas futuras de investigación que podrían resultar interesantes.

Por lo tanto, a lo largo de esta Tesis, se describirán los diferentes métodos ágiles para gestionar un proyecto, para gestionar a los equipos de desarrollo y gestionar a todas las personas implicadas en el desarrollo de un producto (desde el equipo que lo desarrolla, hasta el usuario final de ese producto, pasando por el cliente que nos encarga ese producto).

El desarrollo del estudio de la sección 2 no se centrará en los proyectos de software únicamente, sino que tratará de dar una visión general de este tipo de métodos, de manera que puedan ser aplicados a cualquier tipo de proyecto en el que sean los trabajadores del mismo quienes posean el conocimiento de como llevar a cabo ese proyecto. En este tipo de proyectos, a menudo se desarrollan productos no tangibles, como es el caso del software.

1.2. DEFINICIÓN DE DESARROLLO ÁGIL DE PROYECTOS (PROYECTOS ÁGILES)

El desarrollo ágil de proyectos consiste en métodos de ingeniería basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos autoorganizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando en lapsos cortos. Las diferentes partes del proyecto se desarrollan en unidades de tiempo llamadas iteraciones, las cuales deben durar de una a cuatro semanas.

Los métodos ágiles enfatizan las comunicaciones cara a cara en lugar de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento".

Los métodos ágiles también enfatizan que el trabajo útil es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

En la figura 1: “Esquema general de una metodología ágil” se muestra el esquema general de una metodología ágil, y en el podemos ver la naturaleza cíclica o “iterativa” de estas.

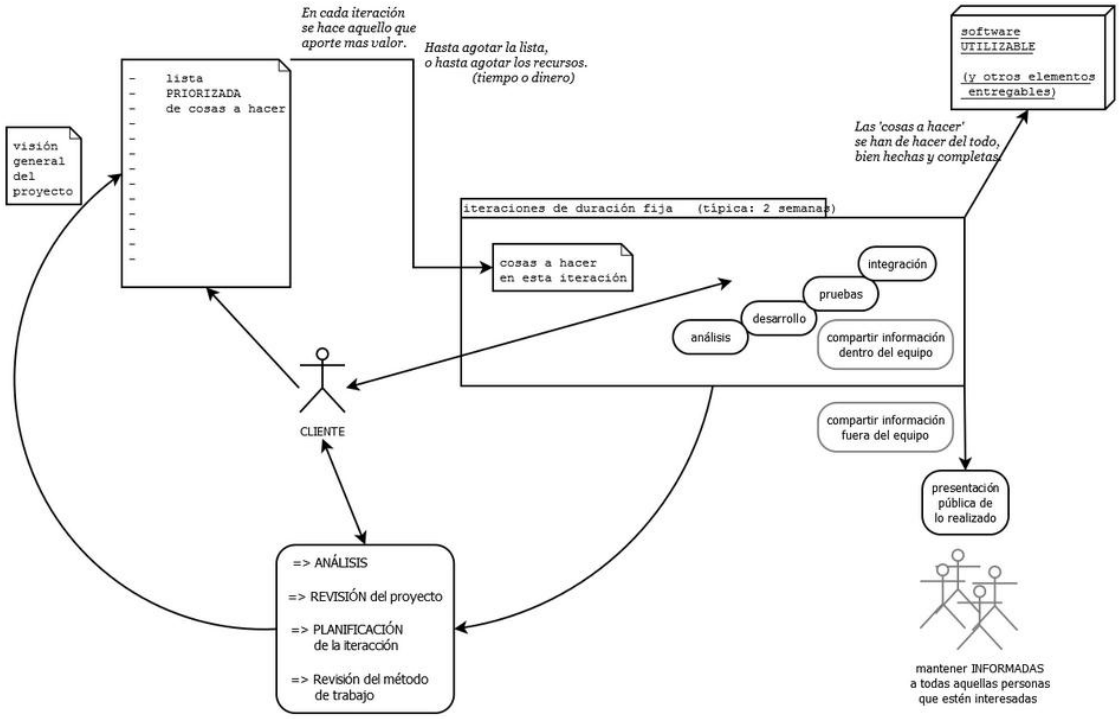


Figura 1: Esquema general de una metodología ágil. (Fuente: Elaboración propia)

1.3. BREVE HISTORIA DE LOS MÉTODOS ÁGILES

La definición moderna de desarrollo ágil de software evolucionó a mediados de la década de 1990 como parte de una reacción contra los métodos de "pico pizado", muy estructurados y estrictos, extraídos del modelo de desarrollo en cascada. El proceso originado del uso del modelo en cascada era visto como burocrático, lento, degradante e inconsistente con las formas de desarrollo de software que realmente realizaban un trabajo eficiente.

Los métodos de desarrollo ágiles e iterativos pueden ser vistos como un retroceso a las prácticas observadas en los primeros años del desarrollo de software (aunque en ese tiempo no había metodologías formales). Inicialmente, los métodos ágiles fueron llamados métodos de "peso liviano".

En el año 2001, miembros prominentes de la comunidad se reunieron en Snowbird, Utah, y adoptaron el nombre de "métodos ágiles". Poco después, algunas de estas personas formaron la "alianza ágil", una organización sin fines de lucro que promueve el desarrollo ágil de aplicaciones. Muchos métodos similares al ágil fueron creados antes del 2000. Entre los más notables se encuentran: Scrum (1986), Crystal Clear (cristal transparente), programación extrema (en inglés eXtreme Programming o XP, 1996), desarrollo de software adaptativo, feature driven development, Método de desarrollo de sistemas dinámicos (en inglés Dynamic Systems Development Method o DSDM, 1995).

Kent Beck creó el método de Programación Extrema (usualmente conocida como XP) en 1996 como una forma de rescatar el proyecto del Sistema exhaustivo de compensaciones de Chrysler (C3). Mientras Chrysler cancelaba ese proyecto, el método fue refinado por Ron Jeffries.

1.4. IMPORTANCIA Y CONTENIDO DEL EXAMEN PMI-ACP

En esta sección, se repasará a grandes rasgos, en que consiste el examen y que sentido o importancia tiene.

Uno de los principales valores de esta certificación, es que procede del PMI, una entidad reconocida por las empresas. El examen PMI-ACP ha sido creado por expertos ágiles para profesionales ágiles y no está ligado a una única metodología ágil.

Podría llegar a ponerse en duda el porque es necesaria una certificación, es decir para que es necesario que una tercera parte verifique que el trabajo es realizado de acuerdo a unas bases determinadas y que se siguen los procesos adecuados.

La respuesta es sencilla: una certificación no se emite principalmente para aquellos que están desarrollando y que pueden perfectamente realizar su trabajo a la perfección, una certificación se emite para dar garantías a un tercero de que, efectivamente, es así.
En la actualidad, sin una certificación que avale la competencia es mucho más complicado ganar proyectos.

1.4.1. Requisitos para poder presentarse al examen

El examen PMI-ACP es únicamente una de las partes de la certificación PMI-ACP. Existen otra serie de requisitos a cumplir relativos a estudios, experiencia y formación. Para poder presentarse al examen se debe cumplir lo siguiente:

	EXPERIENCIA EN PROYECTOS	**EXPERIENCIA EN PROYECTOS ÁGILES**	**FORMACIÓN EN PRACTICAS ÁGILES**
Diploma de escuela secundaria	2.000 horas (en torno a los 12 meses) de experiencia en proyectos de equipo dentro de los últimos 5 años	1.500 horas (en torno a 8 meses) de experiencia en equipos de proyectos ágiles o en metodologías ágiles dentro de los últimos 3 años	21 horas

Tabla 1: Requisitos para el examen PMI-ACP.

Después de 1.500 horas de experiencia en proyectos ágiles, muchos de los conceptos necesarios para pasar el examen serán conocidos. El problema radica en la variedad de métodos ágiles existentes, y en que cada uno de ellos tiene una terminología diferente, por lo que la terminología utilizada en el examen puede diferir de la conocida.

1.4.2. Supuestos de examen

El comprender los siguientes supuestos sobre el examen, puede ser muy útil a la hora de responder ciertas preguntas del examen.

1. A pesar de la experiencia y formación necesarias, el examen se centra en proyectos ágiles básicos. Conceptos complejos quedan fuera del campo de aplicación del examen. Es útil tener esto en cuenta a la hora de comprender el contexto de las preguntas planteadas.
2. Cada método ágil tiene un vocabulario exclusivo. El examen PMI-ACP intenta no basarse en ninguna metodología en concreto, por lo tanto no se utiliza el vocabulario de ninguna de ellas, sino que en el examen se usan términos universales a los métodos ágiles.

1.4.3. Contenido del examen

EL Project management institute proporciona un documento, el "PMI Agile Certified Practitioner (PMI-ACP) Examination Content Outline en su página web (www.pmi.org). En este documento se describen los puntos que cubre el examen, diferenciando entre herramientas y técnicas (En inglés Tools and Techniques (T&Ts)), conocimiento y habilidades (En inglés Knowledge and Skills (K&S)) y secciones.

Herramientas y técnicas (T&Ts)

Son cosas que el examinado debería ser capaz de hacer. El examen trata de comprobar la habilidad de hacerlo con preguntas tipo "haz", "calcula" o "identifica que pasa a continuación". Este tipo de preguntas, constituyen el 50% del examen.

Conocimiento y habilidades (K&Ss)

Conocimientos y habilidades que se examinarán. El examen trata de comprobar la comprensión y memorización de ellas, con preguntas que comprueban si se recuerda el "cómo" y el "porqué" de las materias que se están tratando. Este tipo de preguntas constituyen el 50% del examen.

Secciones del examen

En el examen se definen seis secciones de T&Ts y K&Ss.

En la tabla 2, se resume cada una de las secciones, y las materias que caen dentro de cada una de ellas.

Contenido del examen y reparto según secciones		
Sección	**Herramientas y técnicas (50% del examen)**	**Conocimientos y habilidades (50% del examen)**
Entrega basada en valores	- ROI, VAN, TIR - Gestión ágil del valor ganado - Hoja de ruta del producto - Mapeado de la cadena de valor - Limites del WIP - Priorización relativa - Cartera del riesgo de valores - Diagrama de flujo acumulado - Tablas de Tarea / Kanban - Alquiler - Priorización del valor del cliente - Gráficos de riesgo	- Prototipos, simulaciones, demostraciones - Entrega incremental - Priorización - Normas de proyecto y calidad - Contratación ágil - Contabilidad ágil - Sistemas de pensamiento - Variaciones en métodos ágiles - Análisis basado en el valor
Compromiso de los stakeholders	- Wireframes - Liderazgo de servicio - Trabajo remanente / historia de usuario - Resolución de conflictos - Modelado ágil - Velocidad - Radiadores de información - Equipos distribuidos - Personas - Diagramas burn down / up - Mapas históricos - Negociación	- Incorporando valores de los stakeholders - Gestión de las comunicaciones - Herramientas y técnicas de liderazgo - Gestión de los stakeholders - Escucha activa - Métodos de facilitación - Globalización, cultura y diversidad de equipos - Gestión de ventas - Modelos de decisión participativos

Contenido del examen y reparto según secciones		
Secciones	**Herramientas y técnicas (50% del examen)**	**Conocimientos y habilidades (50% del examen)**
Prácticas para impulsar el rendimiento de equipo	- Reunión de equipo diaria - Equipos localizados - Espacio de equipo - Herramientas ágiles - Liderazgo adaptativo - Inteligencia emocional	- Técnicas de Brainstorming - Construcción de equipos de poder - Orientación y tutoría - Construcción de equipos de alto rendimiento - Motivación de equipos - Co-ubicación y equipos dispersos geográficamente
Planificación adaptativa	- Proceso de adaptación - Iteración y planificación de lanzamiento - Método Wideband-Delphi y planning poker - Elaboración progresiva - Timeboxing - Minimally marketable feature (MMF). - Tiempo ideal - Afinidad estimativa - Tamaño relativo /Puntos de historia	- Estimación del tiempo, presupuesto y coste - Priorización y descomposición basado en el valor - Estatutos ágiles Juegos de innovación
Detección y resolución de problemas	- Tiempo de ciclo - Defectos no detectados - Integración continua - Risk-based space - Verificación y validación frecuentes. - Desarrollo basado en pruebas / Desarrollo de ensayo inicial - Aceptación del desarrollo basado en pruebas	- Resolución de problemas - Control de límites - Modos de fallo y alternativas - Variación y análisis de tendencia

Contenido del examen y reparto según secciones		
Secciones	**Herramientas y técnicas (50% del examen)**	**Conocimientos y habilidades (50% del examen)**
Mejora continua	- Retrospectiva	- Intercambio de conocimiento - Análisis de procesos - Aplicación de nuevas prácticas ágiles - Código PMI de conducta ética y profesional - Mejora continua - Autoevaluación

Tabla 2: Contenido del examen y reparto según secciones

El 50% de las preguntas del examen se basan en herramientas y técnicas y el otro 50% en habilidades y conocimientos. La sección de habilidades y conocimientos se divide en 3 niveles:

- Nivel 1: Existen 18 habilidades y conocimientos en el nivel 1, los cuales son los más importantes a conocer, ya que se corresponden con el 33% del examen.
- Nivel 2: Existen 12 habilidades y conocimientos en este nivel, que se corresponden con el 12% del examen.
- Nivel 3: Existen 13 Habilidades y conocimientos en este nivel, que se corresponden únicamente con el 5% del examen.

2. ESTUDIO TEÓRICO DE LOS MÉTODOS ÁGILES

2.1. ESTRUCTURA ÁGIL

En este capítulo veremos el escenario de la dirección de proyectos ágiles. Se exploraran algunas de las diferencias entre la dirección de proyectos tradicional y la dirección de proyectos ágil viendo las ventajas de la utilización de los métodos ágiles en algunos proyectos.

Muchas de las preguntas del examen tratan sobre herramientas y técnicas específicas, por lo que comprender los conceptos de la estructura ágil es importante para responder las preguntas correctamente.

2.1.1. Justificación de la necesidad de los métodos ágiles

A continuación se responderá a la pregunta de porque son necesarios los métodos ágiles. La respuesta es simple: tipos diferentes de proyectos, precisan de diferentes tipos de métodos. Algunos proyectos, especialmente proyectos que precisan de trabajadores con conocimientos especializados, se desarrollan en ambientes de rápida acción con tiempos acotados, por lo que es más adecuado un método ágil para desarrollarlos.

A continuación, vemos las diferencias entre los proyectos que precisan de trabajadores con conocimientos especializados y los proyectos de producción convencionales. En la tabla 3 se muestra una comparación entre trabajo industrial y trabajo basado en conocimiento.

Características del trabajo industrial	**Características del trabajo basado en conocimiento**
El trabajo es visible (se produce algo tangible)	El trabajo es invisible (no se produce algo tangible)
El trabajo es estable	El trabajo es cambiante
Énfasis en hacer avanzar las cosas	Énfasis en hacer cambiar las cosas
Más estructura con menos decisión	Menos estructura con más decisiones
Se centra en las respuestas correctas	Se centra en las preguntas correctas
Define las tareas	Los trabajadores comprenden la tarea
Control y dirección	Da autonomía
Reglas estrictas	Innovación continua
Se centra en la cantidad	Se centra en la cantidad
Mide el rendimiento con reglas estrictas	Aprendizaje y formación continuos
Minimiza el coste de los trabajadores para una tarea	Trata a los trabajadores como activos, no como gastos.

Tabla 3: Diferencias entre trabajo industrial y basado en el conocimiento.

La comunicación y colaboración que se requiere para los proyectos con trabajadores con conocimientos especializados a menudo es más incierta y menos definida que para trabajos industriales. Los métodos ágiles aparecen como respuesta a la frustración generada de aplicar métodos pensados para proyectos industriales a proyectos basados en el conocimiento de sus trabajadores.

En los métodos ágiles se recogen las técnicas empleadas por los trabajadores con conocimientos especializados y las adaptan para su uso en proyectos, empezando con iniciativas para el desarrollo de software, pero posteriormente aplicándolas a otros proyectos basados en el conocimiento especializado de sus trabajadores.

Este desarrollo de los métodos ágiles tuvo lugar hace muchos años y por mucha gente diferente. Como resultado de esto, se tienen inconsistencias en la terminología y múltiples metodologías. A continuación, se establece un escenario para comprender que es la dirección de proyectos ágiles y explicar algunos de los conceptos clave que es necesario conocer, empezando con el manifiesto ágil.

2.1.2. El manifiesto ágil

El manifiesto ágil surge como resultado de un encuentro en febrero de 2011 que reunió a un gran número de expertos en software y metodologías, los cuales definieron el manifiesto ágil y los principios ágiles.

El manifiesto ágil es el siguiente:

Manifiesto para el desarrollo ágil de software

Estamos descubriendo mejores maneras de desarrollar software haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar:

- *Los individuos y las interacciones* sobre **los procesos y herramientas**
- *Software funcional* sobre **documentación exhaustiva**
- *Colaboración del cliente* sobre **negociación del contracto**
- *Respuesta a los cambios* sobre **seguir un plan**

El significado de esto es que, aunque existe valor en los elementos de la derecha de los enunciados anteriores (en negrita), son más valorados los elementos de la izquierda (en cursiva).

Mientras que el manifiesto ágil es simple en estructura y parco en palabras, hay mucho dentro de los cuatro valores establecidos. La comprensión de las ideas que están siendo transmitidas es importante no solo para el examen PMI-ACP, sino para la aplicación de cualquier metodología ágil en proyectos reales basados en el conocimiento de sus trabajadores, aunque no estén relacionados con el software.

En primer lugar, el formato de "A" (elementos en cursiva anteriores) sobre "B" (elementos en negrita anteriores), indica intención, atención, y esfuerzo. No es blanco o negro, o haz "A" en lugar de "B". En lugar de esto, se sabe que "A" y "B" pueden coexistir en un mismo proyecto, pero que debemos centrar nuestra atención, énfasis en "A" más que en "B". Con esto en mente, vamos a ver en más detalle estas cuatro sentencias.

1. **Los individuos y las interacciones sobre los procesos y herramientas**

El mensaje aquí, es que mientras que los procesos y las herramientas son necesarios en nuestros proyectos, debemos intentar focalizar la atención del equipo en los individuos y las interacciones involucradas.
Esto es así porque los proyectos son llevados por personas, no por herramientas, y los problemas son solucionados por las personas, no por los procesos. De la misma forma, los proyectos son aceptados por las persona, el campo de aplicación es debatido por personas y la definición de proyecto finalizado con éxito es negociada por las personas. Centrándose al principio en desarrollar a los individuos involucrados en el proyecto y enfatizando interacciones productivas y efectivas, ayuda a preparar un proyecto para el éxito.

Esto no quiere decir que los procesos y las herramientas no puedan ayudar a completar de manera exitosa un proyecto. Son ciertamente, activos importantes, y siempre que haya un trasfondo de ingeniería que podemos tender hacia la lógica y predictibilidad de los procesos y las herramientas.

2. **Software funcional sobre documentación exhaustiva**

Este valor, nos habla de la necesidad de entrega. Los proyectos de software, son típicamente iniciados con el objetivo de crear software valioso y de alta calidad, sin embargo muchas veces quedan atrapados en resultados provisionales en forma de extensa documentación, que no ayuda al objetivo final de un software funcional. Un software sin documentación, es realmente problemático y dificulta el soporte y el mantenimiento. Pero la documentación exhaustiva sin software no tiene ningún valor para las organizaciones.

Durante los años 1980 y 1990 se produjo un movimiento que hizo del desarrollo de software más que una disciplina, desembocando en una gran demanda de documentación. Y debido a que en este periodo se dió también la llegada de grandes equipos, sistemas de software complejo y un legado de aplicaciones indocumentadas, está documentación era esencial. Muchos desarrolladores de software, están orientados al detalle y al proyecto guiado; aunque estas características son a menudo muy beneficiosas, muchas veces pueden hacer que la atención del desarrollador se desvíe de la razón principal por la que está realizando el proyecto de software, escribir un software valioso. Por eso el manifiesto ágil marca la atención en la utilidad del software por encima de la documentación exhaustiva como un recordatorio del porqué de estos proyectos – desarrollar algo útil para el cliente y el usuario final.

3. Colaboración del cliente en la negociación del contracto

Este valor nos recuerda que debemos ser flexibles y complacientes, en lugar de intransigentes y poco cooperativos. Es similar a la diferencia entre “ser correcto” y “hacer lo correcto”. Podemos desarrollar el producto exactamente como se ha determinado en la especificación inicial, pero si el cliente cambia su idea inicial o sus prioridades, es mejor ser flexible y trabajar hacia la nueva meta, aunque sea opuesta a la originalmente establecida.
Es extremadamente difícil definir por adelantado una visión sin cambios de lo que se va a desarrollar. Este desafío deriva de la naturaleza dinámica de los proyectos con trabajadores con conocimientos especializados, especialmente en los sistemas de software; el software es intangible y difícil de referenciar, las compañías raramente crean los mismos sistemas dos veces, los negocios necesitan cambiar rápido, y la tecnología cambia rápidamente. En lugar de molestar al cliente con un proceso de cambio de dirección, debemos reconocer al principio que las cosas van a cambiar, y debemos colaborar con el cliente durante el proyecto hacia una definición de “completo”. Esto requiere una relación de confianza y modelos de contrato más flexibles de los que normalmente vemos.

4. Respuesta a los cambios en un plan inicial

Como se acaba de comentar, los planes iniciales son a menudo inadecuados. En lugar de invertir esfuerzo en tratar de reconducir el proyecto a la idea inicial, es preferible gastar más tiempo y esfuerzo en tratar de responder adecuadamente a los cambios que se dan en el proyecto. Con esto no se quiere decir que el manifiesto ágil nos esté diciendo que abandonemos cualquier planificación y únicamente respondamos a los cambios. Seguimos necesitando un plan, pero siempre teniendo en mente que el plan inicial fue realizado cuando no se conocía todo sobre el proyecto (al inicio), y conforme el trabajo progresa, necesitamos actualizar el plan.

La importancia de responder a los cambios sobre seguir rígidamente el plan inicial, es especialmente significativo en los proyectos de software, en los cuales es común un gran número de cambios. De nuevo, en lugar de suprimir los cambios y gastar mucho tiempo intentando dirigir un largo proyecto estático, debemos aprender a que las cosas cambian. Los proyectos ágiles tienen un gran número de colas de trabajo y planes visibles, en forma de retrasos y tablas de tareas. El motivo de la cuarta sentencia, es ampliar el número de personas que pueden ser fácilmente involucradas en la planificación del proceso, ajustar los planes y discutir el impacto de los cambios.

En resumen, el manifiesto ágil no es un conjunto de reglas a seguir, que nos digan que debemos hacer. Es más sutil - nos guía para considerar los proyectos desde una perspectiva basada en el valor.

2.1.3. Principios guía

En adición a los cuatro valores ágiles que hemos visto, los autores del Manifiesto ágil crearon doce principios guía para los métodos ágiles.

Estos valores y principios son la clave para entender los métodos ágiles, y pueden ayudar cuando hay que elegir entre dos opciones que parecen ambas correctas, en el examen PMI-ACP. En esta situación, lo más adecuado es buscar la solución más acorde con estos valores o principios.

Principios detrás del manifiesto ágil.

1. La mayor prioridad es satisfacer al cliente a través de la entrega temprana y continua de software útil.

Existen tres puntos principales en este principio.

El primero es satisfacer al cliente. Si producimos planes perfectos y documentación y solo cumplimos con la oficina de dirección de proyectos o la de calidad, habremos fallado; debemos centrarnos en el cliente.

El segundo punto es la entrega temprana y continua. Debemos estructurar el equipo del proyecto para realizar entregas rápidas y frecuentes. Esto puede ser complicado si la personas del equipo son reacias a compartir un trabajo inacabado, pero debemos ayudar a todos a sentirse a gusto con esta idea. Es mejor encontrar un problema pronto y tener tiempo para solucionarlo, que descubrir ese mismo problema mucho más tarde, cuando ya se ha realizado la mayoría del trabajo, y ver que se ha realizado sobre una base equivocada.

El tercer punto es tener claro que se está realizando, en este caso software válido, pero en otro tipo de proyectos será un producto, un servicio, etc.

2. Estar receptivo a cambios en los requisitos del proyecto sin importar el estado en que se encuentre el mismo. Los procesos ágiles aprovechan los cambios para dar una ventaja competitiva al cliente.

Los cambios pueden ser buenos para un proyecto si nos permiten introducir una nueva característica recién descubierta. Para un punto de vista tradicional de la gestión de proyectos, los cambios se ven de una manera negativa; se designan como un desvío de los objetivos y del plan establecido para el proyecto. Muchos proyectos tradicionales tienen rigurosos procedimientos de control de cambios, bajo los cuales solo las prioridades de más alto nivel pueden ser tratadas. En estos proyectos, gran parte del tiempo y del esfuerzo se gasta registrando y gestionando solicitudes de cambio.

Este tipo de enfoque a una gestión rigurosa de los cambios es problemático para los proyectos de software o para cualquier otro proyecto que experimente altos ratios de cambio. En contraste, la gestión de proyectos ágil acepta los cambios que puedan ocurrir. De hecho, esta metodología aboga por abrazar el cambio. En lugar de crear mecanismos muy pesados para gestionar los cambios, los métodos ágiles utilizan aproximaciones ligeras y muy visibles para priorizar los cambios dentro del trasfondo del trabajo a realizar.

Aceptando que los cambios pueden suceder y desarrollando una vía eficiente para gestionarlos, un equipo puede destinar más tiempo desarrollando el producto final del proyecto. En lugar de suprimir los cambios, los métodos ágiles trabajan para crear una vía bien comprendida y altamente visible, de tratar con esos cambios, lo que dota al proyecto de flexibilidad y adaptabilidad, en la medida de lo posible.

3. Entregar software funcional de manera frecuente, es preferible una frecuencia corta, de un par de semanas frente a una larga, como podría ser un par de meses.

A pesar de nuestras buenas intenciones y aun sabiendo que es importante recibir feedback sobre nuestro trabajo lo antes posible, entra dentro de la naturaleza humana el querer crear algo bueno, antes de compartirlo. No obstante nos estamos perjudicando, ya que es mucho mejor recibir este feedback, y así evitar ir por el camino equivocada durante demasiado tiempo. Este principio, pone énfasis en la importancia de entregar nuestro trabajo para su análisis y recibir feedback sobre éste. El análisis frecuente y el feedback correspondiente son tan importantes que se han creado herramientas para proveer de este feedback a los desarrolladores, sobre cualquier código que rompa el desarrollo. Los equipos ágiles necesitan feedback sobre lo que están creando para saber si pueden continuar progresando o por el contrario es necesario un cambio de rumbo en el desarrollo.

Realizar la entrega en un periodo de tiempo corto, también tiene el beneficio de mantener el negocio en marcha y mantener un diálogo sobre el proyecto de forma continua. Con las entregas frecuente, el equipo del proyecto muestra de manera regular los resultados, obteniendo oportunidades de recibir feedback. A menudo, en estas demostraciones, el equipo aprende nuevos requisitos o cambios en el negocio muy útiles para la planificación del proyecto.

4. Los responsables del negocio y los desarrolladores deben trabajar juntos diariamente durante el proyecto.

Las demostraciones frecuentes son un ejemplo de cómo pueden trabajar juntos los responsables del negocio y los desarrolladores. Los cara a cara diarios con los responsables del negocio es uno de los principios más difíciles de conseguir, pero es importante conseguirlo. Los documentos escritos, e-mails, o las llamadas de teléfono son a menudo ineficientes a la hora de transmitir información, en comparación con la interacción cara a cara.

Trabajando a diario con los responsables del negocio, el equipo de desarrollo alcanza una comprensión del negocio de una manera que va mucho más allá de la alcanzada en las reuniones en las que se dan los requisitos del proyecto. Como resultado de esto, el equipo de desarrollo es más capaz de sugerir soluciones alternativas a las solicitudes de los responsables del negocio. Los responsables del negocio, a su vez, también aprenden que tipo de soluciones son más caras, más lentas de desarrollar, y cuales más baratas, siendo capaces de afinar sus solicitudes en consonancia.

Cuando no es posible realizar una reunión diaria, los métodos ágiles intentan de alguna manera mantener a los dos grupos trabajando juntos de forma regular, tal vez, cada dos días o de cualquier otra manera que funcione. Algunos equipos, sin ser la opción ideal, utilizan un representante del cliente, en el que un analista de negocios experimentado, y familiarizado con los intereses del negocio, sirve como sustituto.

5. Construir los proyectos alrededor de individuos motivados. Proporcionar el ambiente y el soporte que necesiten, y confiar en que conseguirán realizar el trabajo.

En los modelos de estimación de software, la diferencia entre tener la mejor gente comparada con tener los mejores procesos y herramientas, es de un factor de 10:1. Esto significa que tener personal inteligente y motivado en el equipo marca la diferencia para que un proyecto sea entregado exitosa y eficientemente.

Aunque no siempre podamos reunir el mejor equipo, si podemos hacer todo lo posible para motivar a los miembros del equipo que tengamos. Partiendo de que el equipo es un factor muy importante en el proyecto, los métodos ágiles promueven equipos potenciados. El personal trabaja mejor cuando se le da la autonomía para organizar y planificar su propio trabajo. Los métodos ágiles abogan por dar libertad al equipo lo que se traducirá en un incremento de los ratios de productividad.

Los proyectos basados en los conocimientos de los trabajadores, involucran a miembros del equipo expertos en determinadas áreas. Estos equipos de desarrollo se organizan mejor cuando se les permite el tomar las decisiones del día a día, y manejar la planificación diaria del proyecto. Como responsables de proyecto, esto no significa que abandonemos al equipo o que no nos involucremos, en lugar de eso, lo que hacemos es reconocer a los miembros del equipo como expertos en lo que hacen, y les proporcionamos el soporte que necesitan para asegurarnos que alcanzan el éxito.

6. La conversación directa es el método más eficiente y efectivo de transmitir información, tanto al equipo como dentro de éste.

La documentación escrita es buena para crear y dejar plasmados los eventos y las decisiones, pero es lenta y costosa de producir. En contraste, la comunicación cara a cara, nos permite transmitir rápidamente mucha información y emociones y lenguaje corporal.
Con los cara a cara, las preguntas pueden ser rápidamente contestadas en lugar de aparcadas con la esperanza de que se dará una explicación posterior o la respuesta se aclarará después.

En las reuniones de equipo los temas pueden tratarse de manera más directa y abordar problemas o cuestiones más específicos, mientras que en la documentación escrita se debe dar una explicación mucho más extensa, al no conocer que es lo que ya sabe tu interlocutor.
Por supuesto, la recomendación de mantener conversaciones directas no puede aplicarse a todos los negocios, pero se debería seguir siempre que sea posible. Éste es un ejemplo de como los métodos ágiles deben ser adaptados a cada proyecto en concreto. Conforme el equipo crece, las conversaciones cara a cara son más difíciles de realizar, y nos vemos obligados a introducir un nivel apropiado de documentación escrita.

7.El software funcional es el indicador principal del progreso.

Adoptando el software funcional como el primer indicador de progreso, inmediatamente fomentamos el software funcional como el objetivo del proyecto, y los esfuerzos por crear documentación y diseños se transforman en una ayuda, más que en la actividad principal. Si una función no puede ser medida o testeada, no se considera como completa.
El énfasis en el software funcional, ayuda a asegurar que el equipo acepta las características, en lugar de marcar ítems como desarrollo completo, cuando aún no han sido aceptados.

Esta definición de progreso y la naturaleza binaria del software funcional crea una visión orientada a resultados del proyecto. Los entregables internos y el trabajo parcialmente completado, no obtiene reconocimiento externo. En lugar de eso el objetivo final del proyecto - un producto que puede ser utilizado por el negocio - permanece como el objetivo del proyecto.

8. Los procesos ágiles promueven el desarrollo sostenible. Los sponsors, desarrolladores, y usuarios deben ser capaces de mantener un buen ambiente de trabajo.

Los métodos ágiles se esfuerzan por maximizar el valor a largo plazo. Muchas de las técnicas de desarrollo rápido de aplicaciones (RAD en inglés) que preceden a los métodos ágiles promueven periodos largos e intensos de prototipado antes de las demostraciones. El problema con el enfoque RAD de trabajar durante muchas horas a lo largo de un periodo de tiempo muy largo, es que las personas del equipo se queman y comienzan a hacer fallos. Por ello no es una práctica muy aconsejable.

En lugar de periodos de desarrollos largos e intensos, los métodos ágiles reconocen el valor de mantener una paz sostenible que permite a los miembros del equipo tener una vida laboral equilibrada. Un paz sostenible no solo es mejor para el equipo, sino que beneficia también a la organización. Días de trabajo muy largos llevan a las personas a renunciar, con lo que la organización pierde talento y conocimiento. Despedir y contratar nuevos miembros a un equipo es un proceso lento y costoso.

Por lo tanto, trabajar en un ambiente pacificado que puede mantenerse indefinidamente, crea un equipo más feliz y productivo. Los equipos felices interactúan mejor con los responsables del negocio que un equipo estresado. Hay menos tensión, y las relaciones de trabajo mejoran. Algunos de los métodos ágiles, como el método XP recomiendan mantener una jornada laboral de 40 horas como referencia. Hoy en día, muchos métodos no establecen límites específicos, pero recomiendan prestar una atención especial a si el nivel de esfuerzo del equipo se está llevando más allá de lo recomendable para asegurar la paz sostenible.

9. La atención lleva a la excelencia técnica y el buen diseño mejora la agilidad.

A la vez que queremos que el equipo de desarrollo trabaje duro y entregue mucho valor, también nos tenemos que preocupar de mantener el diseño limpio, eficiente, y abierto a los cambios. La excelencia técnica y el buen diseño permiten al producto o al equipo de desarrollo comprender y actualizar el diseño fácilmente.
En el mundo del software, conforme se enreda el código base, la organización pierde la habilidad de responder a las necesidades de cambio. En otras palabras, pierde su agilidad. Por lo tanto, necesitamos dotar al equipo de desarrollo del tiempo suficiente para llevar a cabo de refactorización. La refactorización es el servicio de limpieza y simplificación que necesita el código para asegurar que es estable y puede mantenerse a largo plazo.
Un proyecto necesita balancear sus esfuerzos entre entregar características avanzadas y prestar una atención continua al diseño de soluciones.

10. Simplicidad – el arte de maximizar la cantidad de trabajo no realizado - es esencial.

La característica más segura es la que no implementamos – ya que no hay nada que pueda ir mal en ella. En el mundo del software, hasta el 60% de las características implementadas se usan poco o nada.
A causa de que muchas características nunca se utilizan y a causa de que los sistemas complejos tienen un gran potencial de ser poco fiables, los métodos ágiles se centran en la simplicidad. Es decir, llevar los requisitos a los elementos esenciales únicamente.

Los proyectos complejos llevan mucho tiempo para ser completados, están expuestos a riesgos y tienen más puntos posibles de fallo y más posibilidades de sobrecoste. Por lo tanto, los métodos ágiles buscan "la cosa más simple que pueda funcionar" y recomiendan implementar esta solución la primera. Este enfoque no pretende excluir la extensión y elaboración de un producto – en lugar de eso simplemente dice, "Vamos a coger la opción más corriente primero". Este enfoque no solo mitiga el riesgo, sino que también impulsa la confianza del sponsor.

11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto organizados.

Este principio puede sonar extraño, dependiendo de como se lea. Esencialmente, está diciendo que para sacar lo mejor de tu gente, debemos dejar que se auto organicen. A la gente le gusta auto organizarse; les permite encontrar un enfoque que trabaje mejor con sus métodos, sus relaciones, y su ambiente. Comprenden y apoyan el enfoque, ya que lo han creado ellos y como resultado, producen un trabajo mejor.

Leyendo el principio de una forma más literal, podríamos preguntarnos por qué las mejores arquitecturas, requisitos y diseños, vienen del equipo del proyecto, y no de los arquitectos, analistas de negocios y diseñadores de la organización, quienes pueden o estar en el equipo. La respuesta es que las arquitecturas, diseños y requisitos son mejores si han sido implementados por aquéllos que los originaron. Si las recomendaciones vienen de fuera pueden no ser implementadas exactamente como se pensaron o sin convicción por el equipo de proyecto, por lo que serán menos exitosas.

Los equipos auto organizados con autonomía para tomar decisiones tienen un sentimiento mayor de propiedad y orgullo de las arquitecturas, requisitos y diseños que crean que los que están forzados externamente a realizar las cosas de una manera determinada. Los enfoques creados internamente, pasan por el proceso de aceptación del propio equipo, por lo que no es necesario vender estas ideas al equipo ya que han sido aceptadas por ellos mismos. En cambio, las ideas que proceden de fuera del equipo, deben ser vendidas a éste para ser implementadas de manera correcta, lo que puede ser una tarea complicada.

Otro factor que apoya este principio es que el equipo de proyecto auto organizado está más cercano a los detalles técnicos del proyecto. Como resultado de ello, son más capaces de identificar problemas de implementación, junto con oportunidades de mejora. Así que en lugar de tratar de formar a los recursos externos sobre la estructura de la evolución del proyecto, los métodos ágiles aprovechan la capacidad del equipo para diagnosticar mejor y mejorar las arquitecturas, requisitos y diseños. Después de todo, los miembros del equipo son los más informados sobre el proyecto.

12. A intervalos regulares, el equipo debe reflexionar sobre la eficacia, y ajustar su dinámica en consonancia.

Recopilar las lecciones aprendidas al final de un proyecto suele ser demasiado tarde. En lugar de eso, necesitamos recopilar las lecciones aprendidas mientras éstas son aún aplicables y procesables. Esto significa que debemos recopilarlas durante el proyecto y asegurarnos de que hacemos algo con lo aprendido para ajustar como completamos el trabajo en lo que queda de proyecto.
Los proyectos ágiles, emplean frecuentemente análisis llamados retrospectivas, para reflexionar sobre cómo está funcionando el proyecto e identificar las oportunidades de mejora. Estas retrospectivas típicamente se realizan al final de cada iteración, lo que significa que hay muchas retrospectivas en cada proyecto. Una de las ventajas de hacer retrospectivas de manera frecuente es que los detalles no se olvidan, al contrario de si no se emplea este método y se realiza una revisión única al final del proyecto, en la cual recordar problemas que surgieron un año atrás puede ser muy complicado.

Otra desventaja de no hacer esto, es que lo que se recuerde puede no ser verdaderamente útil para la organización hasta que otro proyecto con un negocio similar llegue. Y en este punto es a menudo fácil, descartar lecciones aprendidas de otro proyecto como no aplicables al proyecto actual. En un proyecto ágil, las lecciones aprendidas son capturadas del equipo conforme el proyecto progresa, por lo que nadie puede decir que no sean aplicables. En lugar de eso el equipo sabe que son relevantes y es empujado a justar su comportamiento de manera acorde.

2.1.4. La declaración de interdependencia

En 2005, los cofundadores del Agile Project Leadership Network (APLN), ahora conocido como el Agile Leadership Network (ALN), crearon la declaración de interdependencia, que veremos a continuación, para la dirección de proyectos ágiles. Esta declaración dice:

Declaración de interdependencia.

Enfoques ágiles y adaptativos para unir gente, proyectos y valor

“Somos una comunidad de líderes de proyecto con gran éxito a la hora de entregar resultados. Para alcanzar estos resultados”:

- Aumentamos el retorno sobre la inversión haciendo que nuestro foco sea la aportación continua de valor.
- Entregamos resultados fiables mediante la participación de clientes en las interacciones frecuentes y promovemos la propiedad compartida.
- Esperamos la incertidumbre y la gestionamos a través de iteraciones, con anticipación y con adaptación.
- Damos rienda suelta a la creatividad y la innovación reconociendo que las personas son la fuente última de valor, y creamos un ambiente donde pueden ser la diferencia.
- Aumentamos el rendimiento a través de la orientación del grupo a los resultados y la responsabilidad compartida para la efectividad del equipo.
- Podemos mejorar la eficacia y fiabilidad a través de estrategias específicas, procesos y prácticas, aplicándolas al contexto especifico del proyecto.

La declaración de interdependencia se centra en la gestión de proyectos como proyectos ágiles. Mientras que el Manifiesto Ágil y los métodos ágiles ofrecen una guía para el proyecto completo, la declaración de interdependencia (DOI) está dirigida a los líderes. Con esta consideración, vamos a analizar con detalle que hay detrás de cada uno de estos principios.

1. Aumentamos el retorno sobre la inversión haciendo que nuestro foco sea la aportación continua de valor.

Esta sentencia quiere decir que los proyectos ágiles concentran sus esfuerzos en desarrollar características que el negocio necesita. A través de este objetivo, el proyecto puede beneficiar al negocio al máximo y ganar un alto nivel de apoyo para el proceso. Cuando un proyecto entrega constantemente resultados, es difícil de ignorar o cancelar y la empresa es más propensa a aprobar peticiones del proyecto.

2. Entregamos resultados fiables mediante la participación de clientes en las interacciones frecuentes y promovemos la propiedad compartida.

Cuando planificamos interacciones con la empresa, debemos intentar ser como el buen vecino que siempre está cuando lo llamas, más que como el pariente que viene a verte y no vuelve a aparecer en un año. Trabajaremos para desarrollar frecuentes interacciones con la empresa, en lugar tener fases muy largas con reuniones, seguidas de no interacciones con la empresa hasta que se finalice el proyecto. Frecuentemente mostraremos a los representantes de la empresa cómo evoluciona el sistema, dejando claro que la empresa está dirigiendo el diseño escuchando y respondiendo a su feedback.

3. Esperamos la incertidumbre y la gestionamos a través de iteraciones, con anticipación y con adaptación.

Dado que la funcionalidad del software es difícil de describir, y dado que la tecnología y las empresas necesitan cambiar rápido, los proyectos de software tienen muchos cambios imprevistos, al igual que otros proyectos con trabajadores especializados. Esta sentencia del DOI indica que en lugar de tratar de crear y seguir un plan rígido que es probable que falle, es mejor planificar y desarrollar en pequeñas partes (iteraciones) y adaptarse a los cambios de requisitos.

4. Damos rienda suelta a la creatividad y la innovación reconociendo que las personas son la fuente última de valor, y creamos un ambiente donde pueden ser la diferencia.

El dicho “Nosotros gestionamos la propiedad y dirigimos personas; si tratas de gestionar personar se sentirán como propiedades” reconoce que necesitamos motivar a las personas, no simplemente decirles que deben hacer. Los proyectos son realizados por personas, no por herramientas o procesos. Para sacar lo mejor de un equipo, debemos tratar a los miembros del equipo como individuos, cubrir sus necesidades y apoyarles en su trabajo. Pagarles un salario puede garantizar que estén ahí, pero el cómo contribuyen mientras están ahí viene gobernado por un amplio abanico de factores. Si quieres obtener los mejores resultados de las personas, provéeles del mejor ambiente.

5. Aumentamos el rendimiento a través de la orientación del grupo a los resultados y la responsabilidad compartida para la efectividad del equipo.

Los equipos potenciados no son solo más felices y más productivos, están más predispuestos a hacerse cargo de los problemas y tratar de resolverlos. Asumiendo un estilo de mando y control para delegar las tareas de trabajo nos podemos encontrar con un lado negativo, y es que si el encontrar una solución a algo no está en la lista de tareas de un miembro del equipo, ¿qué motivación tendrá para hacerlo? En contraste, si los miembros del equipo han hecho un compromiso social con los demás para entregar un trabajo funcional, y surge un problema, trabajarán juntos para proporcionar soluciones. Todos deben compartir la responsabilidad de realizar el proyecto de manera exitosa.

6. Podemos mejorar la eficacia y fiabilidad a través de estrategias específicas, procesos y prácticas, aplicándolas al contexto especifico del proyecto.

Los proyectos reales son complejos y confusos. Raramente se dan todas las condiciones ideales para el desarrollo ágil. En lugar de eso, tenemos que interpretar la situación del proyecto y hacer el mejor uso posible de las técnicas, personas y herramientas que tenemos disponibles. No hay una manera establecida de cómo manejar un proyecto de forma exitosa, necesitamos ajustar nuestro enfoque para adaptarnos lo mejor posible al ambiente del proyecto.

2.1.5. Métodos Ágiles

Existen alrededor de una docena de métodos ágiles que se utilizan actualmente. Los más comunes son Scrum, Extreme Programming (XP), Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), y la familia de métodos Crystal (Crystal Clear, Crystal Yellow…). Otros métodos están basados en los desarrollos Lean o Kanban.

Los métodos más usados de los anteriormente mencionados son el Scrum y el XP, que son los que más aparecen en el examen PMI-ACP. A continuación veremos en detalle los métodos Scrum y XP, y las características principales del resto de métodos.

2.1.5.1. Scrum

Scrum es un popular método ágil ligero y fácil de comprender pero, como el resto de métodos ágiles, difícil realmente de dominar. La metodología documentada en el “Scrum Framework” es un conjunto de prácticas de guía de grupos, roles, eventos, herramientas y reglas, con las cuales ejecutar proyectos. La teoría del método Scrum está basada en tres pilares: Transparencia, Inspección y Adaptación:

- Transparencia: este pilar consiste en dar visibilidad a los responsables del producto final. Un ejemplo de transparencia sería crear una definición común de que se entiende por “acabado”, de manera que todas las partes estén de acuerdo.
- Inspección: este pilar consiste en crear puntos de chequeo en los cuales verificar como está avanzando el proyecto hacia su meta final, vigilando posibles desviaciones problemáticas de lo que sería el resultado final esperado.
- Adaptación: este pilar consiste en ajustar un proceso para minimizar los posibles problemas si una inspección muestra un problema o una tendencia indeseada.

Estos tres pilares guían todos los aspectos de los proyectos Scrum, pero hay además cuatro oportunidades previstas para los pilares de Inspección y Adaptación dentro del esquema Scrum, que veremos a continuación:

- Sprint retrospective
- Daily scrum meeting
- Sprint review meeting
- Sprint planning meeting

Veremos ahora la metodología Scrum, en términos de los diferentes roles dentro de un equipo Scrum, las actividades o eventos que ocurren dentro de la metodología, y sus entregables y herramientas.

EQUIPOS SCRUM

Los equipos Scrum están compuestos de un equipo de desarrollo, el responsable del producto y el ScrumMaster.

Equipo de desarrollo. El equipo de desarrollo es el grupo de profesionales que llevan a cabo los incrementos del proyecto en cada iteración, o “Sprint”, como se define en la metodología Scrum. El equipo de desarrollo, tiene la facultad de dirigir su propio trabajo, y sus miembros están auto organizados y son multifuncionales. En otras palabras, el equipo de desarrollo está formado por personas que pueden desempeñar todos los roles necesarios para completar el trabajo.

Responsable del proyecto. El responsable del proyecto es el encargado de maximizar el valor del producto. Esta persona tiene la única responsabilidad de gestionar lista de cosas pendientes, incluyendo la priorización, precisión, conocimiento compartido, valor y visibilidad.

ScrumMaster. El ScrumMaster es el responsable de asegurar que el método Scrum se entiende y se utiliza. El ScrumMaster es una ayuda para el equipo de desarrollo, eliminando los impedimentos para el avance, facilitando los eventos según se necesiten, y ofreciendo asesoramiento. El ScrumMaster además asiste al responsable del proyecto gestionando la cartera, y comunicando la visión, metas e ítems de la cartera al equipo de desarrollo.

EVENTOS

La metodología Scrum se refiere a varios tipos de actividades o eventos –sprints, sprints planning, daily scrums, Sprint review meetings y sprints retrospectivas.

Sprints. Un Sprint es una iteración delimitada en el tiempo de un mes o menos para construir un producto que potencialmente se pueda poner en venta. Muchos sprints Scrum están entre las dos semanas y un mes. Cada Sprint incluye un "Sprint planning meeting", "daily scrum", trabajo de desarrollo, un "Sprint review meeting", y el "sprint retrospective".
Cada sprint se podría considerar como un miniproyecto. Durante el Sprint, no se realizan cambios que puedan afectar a la meta del Sprint, no obstante el campo de aplicación puede ser renegociado o clarificado tal y como nueva información se haga disponible. Los miembros del equipo de desarrollo se mantienen durante el Sprint.

Sprint planning meeting. Se utiliza para determinar que se va a entregar en el Sprint y como se alcanzará el trabajo propuesto. El responsable del proyecto presenta el listado de cosas pendientes y el equipo completo los discute para crear un conocimiento compartido. El equipo de desarrollo prevé que podrá ser entregado basándose en las estimaciones, la capacidad proyectada, y rendimientos pasados para definir la meta del Sprint. El equipo de desarrollo determina entonces, como se implementará esta funcionalidad y como se organizará el equipo para entregar la meta del Sprint.

Daily Scrum. El "Daily Scrum" es una reunión diaria de 15 minutos. Durante esta reunión, el equipo de desarrollo sincroniza sus actividades, comunicados, y problemas encontrados. Se lleva a cabo en el mismo lugar y horario, y cada miembro del equipo proporciona respuestas para las siguientes tres preguntas sobre el trabajo que ha estado realizando durante el Sprint:

1. ¿Qué se ha conseguido desde la pasada reunión?
2. ¿Qué se va a realizar antes de la próxima reunión?
3. ¿Qué obstáculos hay en el camino?

El "Daily Scrum" se utiliza para asesorar el progreso hacia la meta del Sprint. El ScrumMaster se asegura de que estas reuniones se llevan a cabo y ayudan a eliminar cualquier obstáculo identificado.

Sprint review. Un "Sprint review" es una reunión llevada a cabo al final del Sprint para inspeccionar el avance o evolución del producto, que se ha realizado y cambiar la lista de cosas pendientes si es necesario. El equipo de desarrollo demuestra el trabajo que se ha hecho y responde las preguntas sobre el avance. El responsable del proyecto decide que se puede considerar como terminado y que no. El responsable del proyecto y el equipo discuten la lista de cosas pendientes y determinan que realizar a continuación.

Sprint retrospective. Al final del Sprint, el equipo realiza una retrospectiva del Sprint para reflejar el proceso y encontrar oportunidades de mejora. La retrospectiva se lleva a cabo después del "Sprint review" y antes del siguiente "Sprint planning meeting". Esto permite al equipo incorporar el feeback del responsable de proyecto desde el "Sprint review" y también les permite incorporar las mejoras identificadas durante la retrospectiva en el siguiente plan. El equipo centra su inspección en las personas, relaciones, procesos y herramientas. Exploran que ha ido bien, para identificar oportunidades de mejora que puedan implementar en el siguiente Sprint.

ARTIFACTS

Una vez visto los roles y actividades que estos llevan a cabo, veremos los entregables así como la definición de "acabado" de estos entregables.

Product backlog. Se trata de la lista ordenada de todo lo que será necesario para el producto. Es la única fuente para los requisitos. Este "backlog" es dinámico y evoluciona al mismo tiempo que el producto evoluciona. Contiene características a ser implementadas, funciones, requisitos, atributos de calidad, mejoras y arreglos. Los ítems más importantes están más detallados por lo que sus estimaciones con más precisas. Los ítems menos importantes pueden no ser desarrollados en favor de los ítems más importantes.
La "limpieza" del backlog del producto consiste en añadir más detalle y orden a éste, y refinar las estimaciones de los ítems. Este esfuerzo se lleva a cabo por el equipo de desarrollo y el responsable de proyecto. Por ejemplo, el equipo de desarrollo puede actualizar sus estimaciones, y basándose en esta nueva información, el responsable de proyecto puede cambiar la prioridad de los ítems de la lista de cosas pendientes.

Sprint backlog. El Sprint backlog es el conjunto de ítems extraídos de la lista de cosas pendientes que se seleccionan para un Sprint específico. La lista de cosas pendientes del Sprint se acompaña de un plan de cómo alcanzar esas metas.

Usando estos roles, eventos, herramientas, y definiciones colectivas, el equipo scrum lleva a cabo de manera iterativa pasos del producto final involucrando al cliente de manera frecuente de manera que se aseguran estar creando el producto correcto.

2.1.5.2. Extreme Programming (XP)

Extreme Programming en un método ágil de desarrollo céntrico de software. Mientras que el método Scrum, en el nivel de dirección del proyecto, se centra en priorizar el trabajo y obtener feedback, XP se centra en las buenas prácticas del desarrollo de software.

Los valores principales de esta metodología son la simplicidad, comunicación, feedback, valor y respeto, y éstos se manifiestan en las prácticas llevadas a cabo en el ciclo de vida del proyecto XP.

- **Simplicidad**. Este valor se centra en reducir la complejidad, características extra, y pérdidas. El equipo debe tener la siguiente frase en mente "Buscar la manera más simple de que funcione", y buscar esa solución en primer lugar.
- **Comunicación**. Este valor se centra en asegurarse de que todos los miembros del equipo sepan que se espera de ellos, y que otras personas están trabajando en ello. La reunión diaria es un componente de comunicación clave.
- **Feedback**. El equipo debe obtener impresiones de la idoneidad de su trabajo lo antes posible. Fallar pronto puede ser útil, especialmente si al hacerlo obtenemos nueva información, mientras aún tenemos tiempo de mejorar el producto.
- **Valor**. Se necesita valor para permitir que nuestro trabajo sea enteramente visible para los demás. En programación, los miembros de equipo comparten el código y a menudo necesitan realizar simplificaciones y cambios a éste. Los desarrolladores necesitan tener la confianza para hacer cambios importantes.
- **Respeto**. El respeto es esencial en los proyectos XP, en los cuales las personas trabajan juntas como un equipo y todos cuentan en el éxito o fracaso del proyecto. Los miembros del equipo deben reconocer que cada miembro trabaja diferente y respetar esas diferencias.

En XP, los requisitos de menor peso, también llamados "historias de usuario", se utilizan para planificar lanzamientos e iteraciones. Las iteraciones duran unas dos semanas, y los desarrolladores trabajan en pareja para escribir el código durante esas iteraciones. Todo desarrollo de software está sujeto a rigurosos y frecuentes testeos. Entonces, con la aprobación del cliente, el software es entregado en pequeñas entregas.

"Architectural spikes" son iteraciones utilizadas para probar aproximaciones tecnológicas, y "spikes" son periodos de trabajo utilizados para reducir riesgos. Los "spikes" se introducen en los procesos de planificación de entrega.

Practicas XP.

El método XP ejercita un número de prácticas principales simples pero potentes, que veremos a continuación.

- **Whole Team.** La práctica Whole Team, se basa en la idea de que todos los que contribuyen al proyecto XP se sienten juntos en el mismo lugar, como miembros de un único equipo.

  Este equipo incluye a un cliente – un representante de la empresa que provee los requisitos, prioridades, y dirección de la empresa para el proyecto. El equipo se compone también de desarrolladores, y de personal de calidad (QA) y analistas de negocio (BA), no obstante los roles de QA y BA pueden ser llevados a cabo por desarrolladores si tienen las capacidades adecuadas. Además, en ocasiones, existe un coordinador, el cual ayuda a mantener el equipo en la dirección correcta y guía el proceso, y debería de existir también un gestor que facilitará las comunicaciones externas y coordinará las actividades del equipo.

  XP enfatiza la noción de generalizar especialistas, como oposición a especialistas de funciones específicas. En otras palabras, cualquiera que esté cualificado para una función, la puede llevar a cabo – las funciones no están reservadas para personas que se especializan en una única área en particular. Esta práctica ayuda a optimizar el uso de los recursos, ya que las personas que pueden llevar a cabo múltiples tareas son capaces de cambiar de una función a otra dependiendo de la demanda. Esta práctica también permite compartir la información de manera más eficaz y ayuda a eliminar circunstancias en las cuales personas en determinadas funciones, están inactivas o limitadas en algunos puntos del proyecto.

- **Planning games.** XP tiene dos principales actividades planificadas o “Planning Games” – planificación de entrega y planificación de la iteración.

Una entrega es el impulso de una nueva funcionalidad hasta llegar al usuario de la producción. Un proyecto típicamente tiene una o más entregas, con no más de una o dos entregas en un mismo año. Durante la planificación de entrega, el cliente esboza la funcionalidad requerida, y los desarrolladores estiman la dificultad de incorporar la funcionalidad. Armado con estas estimaciones y prioridades, el cliente presenta el plan para la entrega del proyecto. Los intentos iniciales de estimar pueden ser imprecisos, por lo que este proceso es revisado continuamente y mejorado tal y como evolucionan las prioridades y estimaciones.

Las iteraciones constituyen el ciclo de desarrollo más corto dentro de una entrega, y son llamadas “sprints” por el método Scrum. La planificación de cada iteración, se realiza al inicio de cada una de ellas, o cada dos semanas.

El cliente expone que funcionalidad quiere ver en las próximas dos semanas, y entonces los desarrolladores dividen esta funcionalidad en tareas y estiman el trabajo. Basándose en estas estimaciones (que son más precisas que las estimaciones realizadas durante la planificación de la entrega) y en la cantidad de trabajo completado en la iteración anterior, el equipo decide que cantidad de trabajo piensan que podrán completar en el siguiente periodo de dos semanas.

- **Small releases.** Frecuentemente, en XP se motiva el realizar pequeñas entregas para su testeo, tanto en el nivel de la iteración para demostrar el progreso e incrementar la visibilidad del cliente, como en el nivel de entrega para desplegar rápidamente software funcional a la audiencia objetivo. La calidad se mantiene en estos cortos plazos de entrega por riguroso testeo y a través de prácticas como la integración continua, en las que los tests se realizan tan frecuentemente como sea posible.

- **Customer tests.** Como parte de definir las funcionalidades requeridas, el cliente describe una o más pruebas que muestren que el software funciona. El equipo construye entonces tests automáticos para demostrarse a ellos mismos y al cliente que el software funciona.

- **Collective Code Ownership.** En XP, cada pareja de desarrolladores puede mejorar o completar cualquier código. Esto significa que múltiples personas trabajan en el código completo, lo que resulta en un incremento de la visibilidad y conocimiento del código base. Esta práctica lleva a un alto nivel de calidad; con más gente atendiendo al código, se incrementa la posibilidad de encontrar defectos. Igualmente a menos impacto en el proyecto si uno de los programadores lo abandona, ya que el conocimiento está compartido.

- **Code Standards.** Aunque lo expresado en el punto anterior tiene sus ventajas, permitir que cualquiera pueda completar cualquier código, puede acabar en problemas si cada miembro del equipo realiza la codificación con un enfoque diferente. Para hacer frente a este riesgo, los equipos XP siguen una normativa de codificación consistente, de manera que parezca que todos los códigos han sido escritos por un único programador. Las especificaciones de la normativa que cada equipo utilice no son importantes; lo que importa es que el equipo tenga un enfoque coherente para la escritura del código.

- **Sustainable Pace.** XP reconoce que el mayor nivel de productividad se consigue con un equipo que trabaje a un ritmo sostenible. Mientras que los periodos extra pueden ser necesarios, largas horas de trabajo repetidamente son insostenibles y contraproducentes. La práctica de mantener un ritmo sostenible de desarrollo optimiza la entrega de valor a largo plazo.

- **Metaphor.** Xp utiliza metáforas y símiles para explicar los diseños y crear una visión técnica compartida. Estas descripciones establecen comparaciones que todos pueden entender y ayudan a explicar como debería funcionar el sistema. Incluso aunque el equipo no sea capaz de encontrar una metáfora o símil para describir algo, pueden utilizar nombres comunes para diferentes elementos para asegurar que todo el mundo entiende donde y porque se deben aplicar los cambios.

- **Continuous integration.** La integración implica llevar el código de manera conjunta y asegurándose de que este compila y trabaja correctamente en su totalidad. Esta práctica es crítica, porque saca a relucir los problemas antes de que se haya escrito mucho código de diseño incompatible.

  XP emplea la integración continua, lo cual significa que cada vez que un programador chequea el código (normalmente varias veces al día), se lanzan automáticamente test de integración, por lo que los problemas pueden resolverse de inmediato.

- **Test Driven Development.** Como se ha visto, el testeo es una parte crítica de la metodología XP. Para asegurar una buena cobertura de tests de manera que los problemas puedan ser detectados rápidamente durante el desarrollo, los equipos XP a menudo utilizan la práctica del Test-Driven Development. Con este enfoque, el equipo escribe los tests antes de desarrollar el nuevo código.

  Si los tests funcionan correctamente, el código inicial debe fallar estos tests, ya que la funcionalidad requerida no ha sido aún desarrollada. El código pasará el test una vez que esté escrito correctamente. El Test-Driven Development, se esfuerza en acortar el ciclo de test todo lo posible para recibir los beneficios de un feeback temprano.

- **Refactoring.** Refactoring es el proceso de mejorar el diseño de un código existente sin alterar el comportamiento externo o añadir nuevas funcionalidades. Manteniendo el diseño eficiente, se pueden añadir fácilmente al código cambios y nuevas funcionalidades. Refactoring se centra en eliminar código duplicado, bajos acoplamientos (conexiones dependientes entre dos módulos del código), e incrementando la cohesión.

- **Simple design.** Centrándose en mantener el diseño simple pero adecuado, los equipos XP pueden desarrollar código rápidamente y adaptarlo como sea necesario. El diseño se mantiene adecuado para lo que el proyecto necesita actualmente. Se revisa de manera iterativa e incremental para asegurar que continúa apropiado.

  XP sigue deliberadamente una filosofía de desarrollo que lleva a "¿Cuál es la solución más simple que funciona?" como oposición a complejas estructuras que intentan acomodar una posible flexibilidad futura.

Ya que un código abultado y complejo está unido a muchos proyectos fallidos, el diseño simplificado es una estrategia de mitigación del riesgo.

- **Pair programming.** En XP, el código de producción se escribe por dos desarrolladores trabajando en pareja para escribir y proporcionar análisis en tiempo real del software tal y como éste se va creando. Esta práctica puede parecer insuficiente, pero los defensores de XP afirman que ahorra tiempo ya que las parejas descubren los posibles problemas antes, y existe un beneficio en que el conocimiento esté repartido en dos personas. Trabajar en parejas también ayuda a difundir el conocimiento sobre el sistema a través del equipo.

Adoptando un enfoque disciplinado y riguroso de estas prácticas, los equipos XP salen exitosos a la hora de proporcionar sistemas de software de alta calidad.

2.1.5.3. Desarrollo basado en las características (FDD)

El desarrollo basado en las características (FDD) es un enfoque muy fácil de entender aunque potente para construir productos o soluciones. Un equipo de proyecto siguiendo el método FDD desarrollará en primer lugar un modelo global para el producto, seguidamente una lista de características, y un plan de trabajo. El equipo pasará seguidamente al diseño y desarrollará iteraciones para desarrollar las características (ver figura 2).

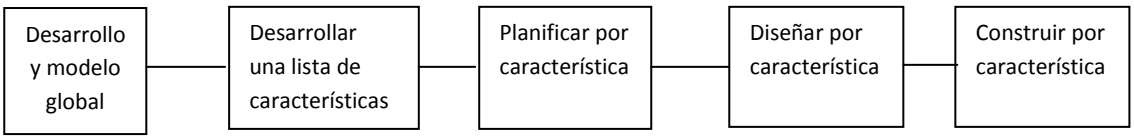


Figura 2: Desarrollo basado en características FDD. (Fuente: Elaboración propia)

FDD recomienda un conjunto de buenas prácticas, derivadas de la ingeniería de software. Estas prácticas incluyen:

>> Ámbito del modelado de objetos: En esta práctica, los equipos exploran y explican el ámbito (o el ambiente de negocio) del problema a ser resuelto.

>> Desarrollo por característica: Esta práctica implica romper funciones trozos de dos semanas o menos y llamarlas características.

>> Clase (código) individual de propiedad: Con esta práctica, las áreas de código tienen un único propietario para garantizar la consistencia, rendimiento e integridad conceptual (a diferencia del planteamiento colectivo del método XP).

>> Equipos por característica: consiste en equipos pequeños, formados dinámicamente que permiten evaluar múltiples opciones de diseño antes de escoger un diseño. Los equipos de característica ayudan a mitigar los riesgos asociados a la propiedad individual.

>> Inspecciones: Son revisiones que ayudan a asegurar un diseño y código de buena calidad.

>> Manejo de la configuración: Esta práctica implica etiquetar códigos, rastrear los cambios, y manejar el código fuente.

>> Construcciones regulares: A través de construcciones regulares, el equipo se asegura de que el nuevo código se integra con el código existente. Esta práctica también permite crear fácilmente una demo.

>> Visibilidad del progreso y de los resultados: Esta práctica rastrea el progreso basado en el trabajo completado.

El desarrollo basado en las características es una metodología ágil que populariza los diagramas de flujo acumulativos y los diagramas de parking. Ambas son herramientas útiles de rastreo y diagnóstico que se utilizan ahora para otros enfoques ágiles.

2.1.5.4. Método de desarrollo dinámico de sistemas (DSDM)

El método DSDM fue uno de los primeros métodos ágiles, y empezó muy prescriptivo y detallado. Su cobertura del ciclo de vida del proyecto es amplia, abarcando aspectos de un proyecto ágil que van desde la viabilidad y la justificación económica hasta la implementación.

DSDM se centra en 8 principios. Aunque estos principios fueron creados antes del Manifiesto Ágil, están alineados con éste. Los 8 principios son:

1. Centrarse en la necesidad del negocio.
2. Entregar a tiempo.
3. Colaborar.
4. Nunca comprometer la calidad.
5. Construir incrementalmente desde bases firmes.
6. Desarrollar de manera iterativa.
7. Comunicar continuamente y de manera clara.
8. Demostrar control.

DSDM ayudó a popularizar anticipadamente las consideraciones arquitecturales, filtros ágiles idóneos, y contratos ágiles.

2.1.5.5. Crystal

Crystal es una familia de metodologías diseñada para proyectos que van desde los dirigidos por pequeños equipos desarrollando sistemas de baja criticidad (Crystal Clear), a aquéllos dirigidos por grandes equipos construyendo sistemas de alta criticidad (Crystal Magenta). El sistema Crystal provee un gran ejemplo de como podemos adaptar un método para que coincida con las características de un proyecto y una organización. En la figura 3, vemos los métodos Crystal básicos:

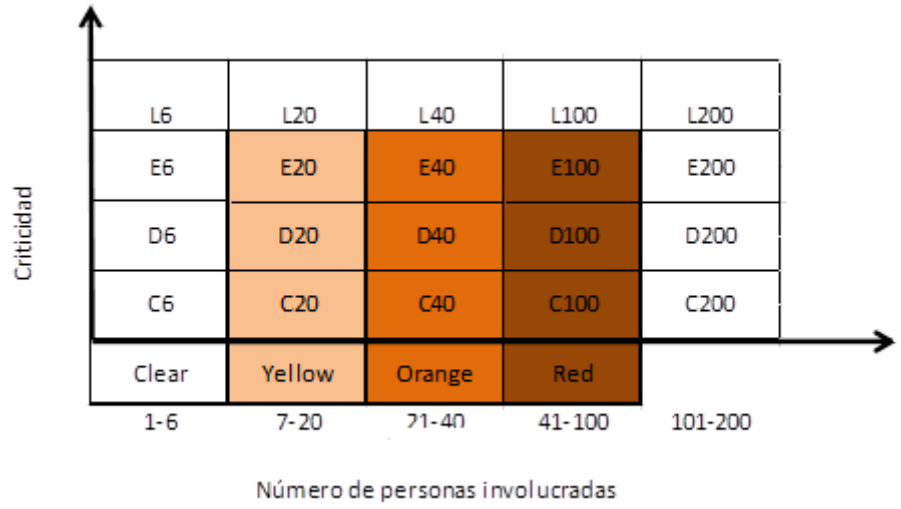


Figura 3: métodos Crystal básicos (Fuente: Elaboración propia)

El tamaño del equipo viene representado en el eje de las X y el nivel de criticidad del proyecto en el eje de las Y. Crystal Clear está dirigido a proyectos pequeños con equipos de una a seis personas. Los proyectos Crystal Clear desarrollan sistemas en los cuales un fallo puede resultar en una pérdida de confort (p.e. un fallo en un videojuego) o una pérdida financiera de bajo nivel (p.e. un fallo del procesador de textos resultante en una pérdida de tiempo en el negocio). Así como el proyecto involucra a más gente y desarrolla aplicaciones más críticas, necesitamos escalar los procedimientos detrás de la comunicación cara a cara e introducir validaciones adicionales y medidas de trazabilidad.

Las metodologías Crystal abrazan y promocionan muchos otros principios ágiles como:

>> Entrega frecuente: Las metodologías Crystal construyen incrementos de una solución y chequean estos incrementos para su aceptación.

>> Mejora reflejada: Esta práctica implica un chequeo regular de las vías para mejorar y luego implementar los nuevos métodos.

>> Comunicación osmótica: Los miembros del equipo son recolocados para permitirles compartir la información.

>> Seguridad personal: Los métodos Crystal enfatizan la necesidad de crear un ambiente en el cual las personas puedan plantear con seguridad problemas o preguntas.

>> Foco: Los miembros del equipo saben en que trabajar y tienen el tiempo y la tranquilidad para trabajar en ello.

>> Fácil acceso para usuarios expertos: A través de este acceso, el equipo puede obtener actualizaciones de los requisitos y feedback.

>> Ambiente técnico: Los métodos Crystal confían en tests automatizados, manejo de configuraciones e integración frecuente.

2.1.5.6. Desarrollo de software Lean

Estrictamente hablando el desarrollo de software Lean no es un método ágil, pero los valores de Lean y ágil están cercanamente alineados. Lean es un conjunto de principios que han sido obtenidos de los enfoques Lean manufacturing y aplicados a desarrollo de software. Estos principios se centran en siete conceptos fundamentales, que vemos en la figura 4.

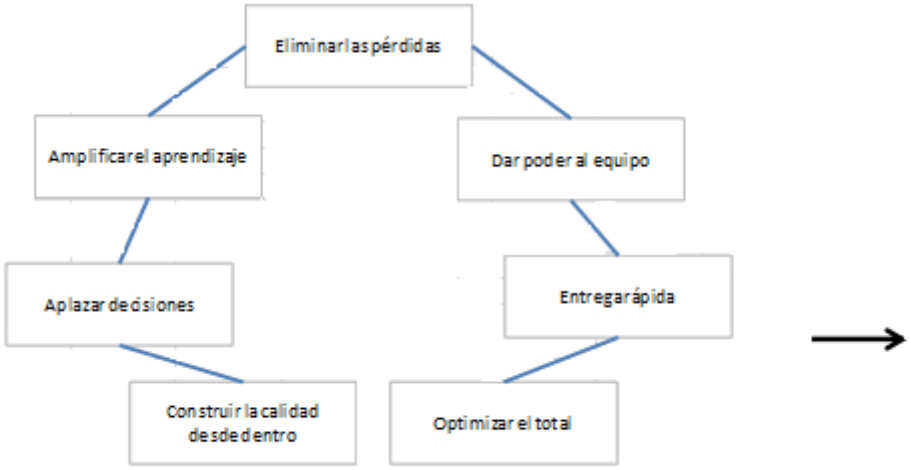


Figura 4: Conceptos fundamentales desarrollo Lean. (Fuente: Elaboración propia)

Ahora los veremos con más detalle:

>> Eliminar las pérdidas: Para maximizar el valor, debemos minimizar las pérdidas. Para los sistemas de software, las pérdidas pueden tomar la forma de trabajo parcialmente hecho, retrasos, transferencias, características innecesarias… Por lo tanto, para incrementar el valor que obtenemos de los proyectos, debemos desarrollar vías para identificar y eliminar estas pérdidas.

>> Dar poder al equipo: En lugar de tomar un enfoque de microgestión, debemos respetar el conocimiento superior que tienen los miembros del equipo de los pasos técnicos necesarios en el proyecto y permitirles tomar decisiones locales para ser productivos y obtener éxito.

>> Entrega rápida: Podemos maximizar la rentabilidad en inversión del proyecto entregando un software valioso de manera rápida e iterando a través del diseño. Encontramos la mejor solución a través de la rápida evolución de las opciones.

>> Optimizar el total: Nuestro objetivo es ver el sistema como la suma de sus partes. Vamos más allá de las partes del proyecto y buscamos como se alinean con la organización. Como parte de optimización del total, también nos centramos en la formación de mejores relaciones intergrupales.

>> Construir la calidad desde dentro: El desarrollo Lean no intenta testear la calidad al final; construye la calidad en el producto y continuamente asegura la calidad a través del proceso de desarrollo, usando técnicas como la refactorización, integración continua, y testeo de las unidades.

>> Aplaza decisiones: Equilibramos la planificación temprana tomando decisiones y comprometiéndonos con las cosas lo más tarde posible. Por ejemplo, esto puede significar re-priorizar la cartera de pedidos hasta que sea hora de planear una iteración, o evitar estar atado a una solución tecnológica determinada.

>> Amplificar el aprendizaje: Este concepto implica facilitar la comunicación desde el primer momento y a menudo, obtener feedback lo antes posible, y construir sobre lo que aprendemos. Los proyectos de software son experiencias de aprendizaje de negocios y tecnología, por lo que debemos empezar pronto y mantenernos en aprendizaje continuo.

Lean nos da técnicas y conceptos como el mapeo del flujo de valor, las siete formas de pérdidas, y trabajo en progreso (WIP).

2.1.5.6. Desarrollo Kanban

El desarrollo Kanban deriva de los sistemas de producción Lean utilizados en Toyota. “Kanban” es una palabra japonesa que significa tablón de anuncios. El tablón de anuncios o tablón de tareas Kanban, como también se llama, juega un importante papel en la metodología de desarrollo Kanban.

La metodología de desarrollo Kanban limita el trabajo en progreso (WIP) para ayudar a identificar los problemas y minimizar las pérdidas y costes asociados con los cambios durante el desarrollo. Utiliza un sistema de salida para tomar el trabajo, dentro de los límites del WIP, a través de las etapas de desarrollo.

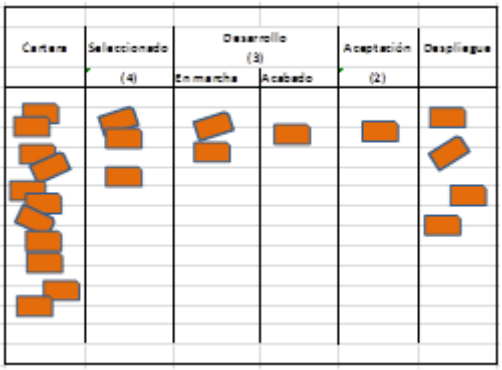


Figura 5: tablón de tareas Kanban. (Fuente: Elaboración propia)

En la figura 5 anterior podemos ver la carga de trabajo en la primera columna. En la segunda columna se indica que hay seleccionadas tres tareas para su desarrollo actualmente. El proyecto tiene un límite de WIP de cuatro tareas. El límite de WIP es el número máximo de ítems que puede haber en esta columna o número máximo de tareas en curso. Si se alcanza el límite no se podrán mover nuevos ítems a la columna hasta que otros ítems se muevan fuera. Las columnas de "Desarrollo" y "Aceptación" también tienen límites de WIP; en este caso hay tres y dos, respectivamente.

El desarrollo Kanban opera según cinco principios fundamentales:

>> Visualiza el flujo de trabajo: Los proyectos con trabajadores con conocimientos, por definición, manipulan el conocimiento, el cual es intangible e invisible. Por lo tanto, es muy importante tener una manera de visualizar el flujo de trabajo para organizar, optimizar y seguirlo.

>> Limitar el WIP: Manteniendo la cantidad de trabajo en proceso baja, aumentamos la visibilidad de los problemas y cuellos de botella, y a la vez facilitamos la mejora continua. Esto crea un sistema de salida del trabajo a través del esfuerzo de desarrollo, reduce los costes asociados con los cambios, y minimiza los costes perdidos.

>> Dirige el flujo de trabajo: Mediante el seguimiento del flujo de trabajo a través de un sistema, los problemas pueden ser identificados y se puede medir la efectividad de los cambios.

>> Hacer políticas de proceso explicitas: Es importante explicar claramente como funcionan las cosas de manera que el equipo pueda tener discusiones abiertas sobre las mejoras objetivamente en lugar de emocional o subjetivamente.

>> Mejorar la colaboración: A través de la medida y experimentación científica, el equipo debe mejorar de manera colectiva los procesos que utiliza.

La metodología de desarrollo Kanban ha ayudado a popularizar las tablas de tareas con límites en el WIP, y los sistemas de salida creando limitaciones en el WIP. También da permiso y alienta a los miembros del equipo para intentar nuevos enfoques y cambios en los procesos. Con un modelo de salidas continuas, no son necesarias iteraciones, y actividad como crear estimaciones pueden ser consideradas como pérdidas y eliminadas por completo.

2.1.6. Proceso de adaptación

2.1.6.1. Guías del proceso de aprendizaje

Igual que cuando haces de electricista en tu casa, puede ser peligroso aprender los procesos ágiles si no entiendes completamente por qué las cosas son como son. Como recomendación general, las organizaciones y equipos nuevos en los métodos ágiles, deberían utilizar los métodos tal y como están concebidos, antes de intentar cambiarlos. La razón de tomar este enfoque es que los problemas que encuentra un nuevo equipo con una técnica o práctica estándar se deberán a la falta de experiencia o habilidad del equipo utilizándola, más que a problemas con la propia técnica. Descartando o cambiando una práctica antes de reconocer su valor, el equipo se arriesga a perder sus beneficios para el proyecto para los cuales la práctica fue originalmente diseñada.

Además, las diferentes técnicas y prácticas se han creado en equilibrio las unas con las otras. Eliminando o aumentando alguno de estos elementos sin entender la relación entre ellos, puede llevar a ocasionar problemas. Por ejemplo, en XP, el testeo permite una refactorización valiente, y tener conversaciones frecuentes con el usuario permite al proyecto tener requisitos más suaves. Si un equipo elimina una de las prácticas sin comprender su contrapeso, el proyecto puede ir de cabeza a un problema.

Para simplificar, las personas deben conocer exactamente cómo funciona el proceso, antes de eliminar nada o inventar nuevas prácticas.

2.1.7. Ética y responsabilidad profesional

Antes de pasar al resto de capítulos veremos como el código de conducta ética y profesional se aplica al concepto ágil. Hay cuatro áreas principales de este código – responsabilidad, respeto, justicia y honestidad – que veremos por turnos, buscando ejemplos de como se relacionan con los conceptos vistos anteriormente.

Responsabilidad.

Tomar decisiones basadas en los intereses de la compañía. Las prácticas ágiles apoyan fuertemente esta área de responsabilidad profesional y social. Estableciendo una definición de grupo de que es "hecho" y teniendo unos altos niveles de participación de los sponsors y los usuarios se ayuda a asegurar que los intereses de la compañía permanezcan al frente de los proyectos ágiles.

Reportar comportamiento no ético y violaciones. Los valores principales del XP de respeto y coraje se aplican aquí. Debemos tener respeto por hacer lo correcto y coraje para hablar si observamos comportamientos no éticos.

Respeto.

Mantener una actitud de mutua cooperación. El concepto Kanban de "mejorar la colaboración" encaja bien con esta meta de cooperación mutua. Conceptos como la definición de hecho, y el rol integral del negocio en la entrega y la planificación de la iteración también aplica aquí.

Respetar las diferencias culturales. De nuevo, nos podemos fijar en el valor principal de respeto del XP, que se encuentra directamente alineado con esta meta. Debemos respetar a los demás y sus diferencias para lograr un ambiente de trabajo sano.

Negociar de buena fe. El valor del manifiesto ágil, "colaboración con el cliente sobre negociación del contrato", nos centra más en el valor del elemento negociador, que en el del contrato. Las negociaciones y contratos, deben resultar beneficiosos para ambas partes, el contratado recibe el pago necesario para crear un producto que sea realmente útil para el cliente.

Hacer frente al conflicto directamente. Crear un ambiente sano para la discusión y el debate es uno de los aspectos principales de potenciar equipos. El conflicto puede ser positivo si nos ayuda a crear mejores soluciones, pero puede ser perjudicial si ocurre sin respeto y una mediación razonable. Crear un ambiente en el cual el conflicto es productivo en lugar de destructivo es parte del rol facilitador del Scrum Master y cualquier líder.

No uses tu posición para influenciar a otros. Los proyectos ágiles enfatizan el rol de liderazgo compartido que ayuda a reducir la posibilidad de que una persona con autoridad utilice su poder para influenciar en las decisiones del equipo. De todas formas, es algo que debemos vigilar. Gran parte del valor de los métodos ágiles se encuentra en potenciar equipos que tomen sus propias decisiones y creen sus propias estimaciones. Este valor se pierde si las personas con autoridad anulan las decisiones del equipo o intimidan a los miembros del equipo, de manera que éstos no expresen sus ideas.

Justicia.

No discriminar. El valor de respeto del método XP, y el enfoque del método Scrum de posicionar al ScrumMaster en un rol de facilitador, ayuda a proteger de la discriminación. La guía para la retrospectiva, que veremos más adelante, habla sobre crear un ambiente seguro para todos para contribuir a la discusión y advertir contra los peligros de la discriminación.

Honestidad.

Comprender la verdad. Muchos de los eventos ágiles programados, incluyendo sprints demos y retrospectivas, están centrados en la transparencia y la comprensión de la verdad. La meta de estos eventos es sacar a la luz rápidamente estos malentendidos para que la acción correctora apropiada pueda ser tomada.

Ser veraz en todas las comunicaciones. El método Scrum incluye la transparencia como uno de sus valores principales. Transparencia significa que honesta y abiertamente reportamos velocidad, demostramos el trabajo realizado y el estatus del proyecto. También reportamos el trabajo basándonos en las características conseguidas, en lugar de basándonos en las que quedan por desarrollar, para eliminar cualquier confusión o falta de claridad en el estatus de las comunicaciones.

EJERCICIOS

En el Anexo F.1, se encuentra un examen tipo sobre lo tratado en este punto.

2.2. ENTREGAS BASADAS EN EL VALOR

Proporcionar valor, especialmente en negocios, es una de los componentes principales de los métodos ágiles. Este concepto se teje dentro del ADN ágil con su inclusión en los valores ("Software funcional por encima de documentación comprensible") y principios ("Entregar software funcional de manera frecuente" y "El software funcional es la primera medida de progreso") del "Manifiesto ágil". Centrarse en entregar valor es la base de muchas de las actividades y decisiones de un proyecto ágil, y se manifiesta en muchas de las herramientas, técnicas, conocimiento y habilidades utilizadas. Este enfoque es un componente esencial de los métodos ágiles.

Definición de entregas basadas en el valor

La razón por la cual los proyectos se llevan a cabo, es para generar valor de negocio, ya sea para producir un beneficio o para mejorar un servicio. Incluso los proyectos de cumplimiento de seguridad y regulatorios, se pueden expresar en términos de valor de negocio, considerando el riesgo del negocio y el impacto de no considerarlo. Por lo tanto, si el valor es la razón de llevar a cabo proyectos, las entregas de valor deben ser el centro del proyecto a través de planificar, ejecutar y controlar los esfuerzos.

Con el objetivo de maximizar el valor, la entrega es el tema central, de los equipos ágiles. Cuando nos enfrentamos a una decisión entre dos o más opciones, a menudo nos hacemos una pregunta para ayudarnos a elegir el camino a seguir, ¿Qué opción dará mayor valor al negocio o al cliente?". También tenemos la obligación de considerar las dependencias técnicas y los riesgos. Un elección que pueda llevar a futuros problemas o rehacer el trabajo ya hecho, no sería compatible con la meta futura del desarrollo basada en el valor, aunque diera la ilusión de un progreso rápido. Por lo tanto, necesitamos considerar los requisitos no funcionales y los riesgos, y advertir al cliente sobre como, estos elementos, impactarán en el proyecto.

Riesgo como anti-valor.

El concepto de riesgo es muy cercano al de valor, tanto es así que podemos pensar en los riesgos negativos del proyecto como anti-valor, o factores que tienen el potencial de erosionar, eliminar, o reducir el valor si llegan a ocurrir. Si el valor es la cara de una moneda, el riesgo es la cruz. Para maximizar el valor, debemos minimizar el riesgo, ya que el riesgo puede reducir el valor. Ésta es la razón por la cual la entrega basada en el valor enfatiza en las técnicas y conceptos de reducción del riesgo.

Otra manera de comprender la relación entre valor y riesgo, sería pensar en la entrega basada en el valor como en la creación de créditos o depósitos que se pagan en la cuenta de un banco. Los riesgos – o como mínimo las amenazas que ocurren y se transforman en problemas en el proyecto – serían como pagos o cargos extraídos de esa cuenta. Para crear el mayor valor, necesitamos maximizar las entradas y minimizar las salidas de la cuenta.

Entrega anticipada de valor.

Los métodos ágiles promueven la entrega anticipada de valor. Esto quiere decir que el equipo tiene como objetivo entregar las porciones de mayor valor del proyecto lo antes posible. Hay algunas razones clave para este enfoque. Primero, la vida es corta, suceden cosas extrañas, y cuanto más se alargue un proyecto más posibilidades hay de encontrar riesgos de fallo, reducción de beneficios, erosión de las oportunidades, etc. Para maximizar el éxito, debemos tener el objetivo de entregar cuantos más componentes de alto valor, lo antes posible, antes de que las cosas cambien o vayan en otra dirección.

La segunda razón principal es que la satisfacción de los stakeholders juega un papel principal en el éxito del proyecto. Mantener comprometidos a los patrocinadores y a los representantes del negocio que mantienen el proyecto es vital para eliminar los obstáculos y definir el éxito. Todos los equipos de proyecto están en periodo de prueba al inicio, mientras que los patrocinadores no se convencen de que pueden realizar las entregas. Entregando elementos de alto valor anticipadamente, el equipo demuestra una comprensión de las necesidades de los stakeholders, reconocimiento de los aspectos más importantes del proyecto, y demuestran que pueden realizar las entregas. Los resultados tangibles, aumentan la confianza de los patrocinadores, establecen una buena comunicación con ellos y los sube a bordo del proyecto, creando círculos virtuales de apoyo.

En resumen, la entrega basada en el valor trata de tomar decisiones que prioricen las actividades que añadan valor y las acciones que reduzcan el riesgo para el proyecto, y llevan a cabo el proyecto basándose en estas prioridades.

Una vez vistos los conceptos básicos de la entrega basada en valor, el resto de este capítulo se centrará en las prácticas de evaluación, planificación, entrega, confirmación, seguimiento, e informar del valor añadido, incluyendo las herramientas, técnicas, conocimientos y habilidades involucrados en estas prácticas, como se muestra en la siguiente tabla.

Práctica	Herramienta / Técnica	Conocimiento / Habilidad (nivel)
Evaluación del valor	- Rentabilidad de la inversión - Valor actual neto - Ratio interno de rentabilidad	
Planificación del valor	- Chartering - Valor “Stream mapping” - Priorización del valor por el cliente. - Priorización relativa / clasificación. - Hoja de ruta del producto - Reservas ajustadas al riesgo	- Priorización (nivel 1) - Métodos ágiles de contratación (nivel 3)
Entrega de valor	- Tablas de Tareas / Kanban - Límites del WIP	- Entrega incremental (nivel 1)
Confirmación del valor	- Priorización del valor por el cliente.	- Retroalimentación a través de prototipos, simulaciones y demostraciones (nivel 1)
Seguimiento e informe del valor	- Gestión del valor ganado por los proyectos ágiles. - Diagramas de flujo acumulativo (CFDs). - Gráficos de riesgo - Tablas de Tareas / Kanban	

Tabla 4: Herramientas, técnicas, conocimientos y habilidades involucrados en las prácticas ágiles.

2.2.1. Evaluación del valor

El valor de los negocios generalmente se evalúa en términos financieros. Algunos proyectos se realizan con propósitos de seguridad o para cumplir con la normativa y no se pueden determinar fácilmente por un valor monetario. En estos casos, la organización puede querer ver las ramificaciones financieras de no llevar a cabo el proyecto, tales como el riesgo de cerrar de la empresa, multas o demandas. O puede simplemente etiquetar el proyecto como obligatorio y no perder tiempo adicional en intentar cuantificar su valor.

Para los proyectos de negocios, el valor se estima comúnmente, utilizando métodos tales como la rentabilidad de la inversión (ROI), tasa interna de rentabilidad (TIR), y el valor actual neto (VAN). ROI, TIR y VAN se encuentran entre las herramientas y técnicas cuya comprensión será evaluada en el examen.

El examen PMI-ACP se centra en el cuándo y el porqué del ROI, VAN y TIR por encima del cómo y el qué. En otras palabras, se harán preguntas sobre cuando esas técnicas se deberían utilizar y porqué se usan esas técnicas, pero no se pedirá que se realicen los cálculos. Utilizar modelos económicos para evaluar el valor, elimina vicios y emociones individuales del proceso de seleccionar y justificar los proyectos. En lugar de eso, con los modelos económicos, te puedes centrar en comparar una variable común (rentabilidad financiero) a lo largo del proyecto. Los resultados están en los números, y son (en teoría) más objetivos que otros modelos de selección de proyectos (en la práctica, no obstante, las entradas de los cálculos pueden ser manipuladas para balancear los resultados).

2.2.1.1. Rentabilidad de la inversión (ROI)

En el ámbito empresarial el cálculo o determinación de las rentabilidades (en especial de la rentabilidad económica o rentabilidad de la inversión y de la rentabilidad financiera) de una empresa es uno de los puntos esenciales del análisis de sus cuentas anuales y una información financiera muy apreciada, tanto a los agentes económicos externos (entidades financieras, proveedores, acreedores, Administración Pública, …) como a sus accionistas y gestores.

La rentabilidad económica (RE) o rentabilidad de la inversión se conoce también con las siguiente terminologías ROI (Return on Investments = Rentabilidad de las inversiones), ROA (Return on Assets = Rentabilidad de los activos) o ROCE (Return on Capital Employed = Rentabilidad de los activos empleados).

La rentabilidad económica mide la tasa de devolución producida por un beneficio económico (BAII – Beneficio Antes de Intereses e Impuestos) respecto al Pasivo Total (o Activo Total). Por lo tanto, la fórmula para el cálculo del ROI es:

ROI = BAII / Activo Total

La rentabilidad económica expresa cuántos euros gana la empresa por cada euro que tiene.

Si multiplicamos y dividimos la fórmula por la cifra de ventas el ratio quedaría, la fórmula de la rentabilidad económica quedaría expresada en dos ratios, como se muestra a continuación.

ROI = (BAII / Ventas) * (Ventas / Activo Total)

El primer ratio (BAII / Ventas) mide el margen económico de la empresa.
Suponiendo que este ratio es toma un valor de 0,3, por cada euro que ingresa la empresa en concepto de ventas, 30 céntimos corresponden a su beneficio, mientras que 70 céntimos corresponden a los costes. A la empresa le interesa que este ratio sea lo más alto posible.

El segundo ratio (Ventas / Activo Total) se conoce como el ratio de rotación. Este ratio de rotación de activos se interpreta normalmente como una medida de eficiencia productiva. Cuanto más elevado sea el ratio, mejor, puesto que significa que la empresa está aprovechando mejor sus activos.

Suponiendo, un ejemplo muy sencillo de dos empresas con la siguiente información financiera:

Empresa A

BAII = $1.000
Ventas = $2.000
Activo Total = $3.000

Empresa B

BAII = $1.500
Ventas = $2.500
Activo Total = $3.000

ROI = (BAII / Ventas) * (Ventas / Activo Total)

ROI (A) = 1.000/2.000 * 2.000/3.000 = 0,5 * 0,667 = 0,33

ROI (B) = 1.500/2.500 * 2.500/3.000 = 0,6 * 0,833 = 0,5

Como podemos ver en este sencillo ejemplo, la empresa B tiene una Rentabilidad económica más alta que la empresa A. La empresa B tiene un margen económico más alto que la empresa A, 0,6 frente a 0,5, lo que indica que por cada euro que ingresa la empresa Ben concepto de ventas, 40 céntimos corresponden a su beneficio, mientras que 60 céntimos corresponden a los costes. En relación al ratio de rotación, la empresa B tiene un 0,833 frente al 0,667, la empresa B tiene una mejor eficiencia productiva. Si analizamos con detalle, comprobamos que teniendo el mismo valor de activos totales, $3.000, la empresa B consigue obtener unas mayores ventas.

Veamos un ejemplo de rentabilidad de la inversión. Tenemos un proyecto que durará desde enero hasta junio, cuando entregaremos una solución que generará cierta rentabilidad financiera. De enero a junio, gastaremos dinero en el proyecto, pagando tarifas por hora por los recursos (se muestra en la siguiente figura con la línea con marcadores de triángulos indicando el flujo de caja negativo acumulado). En junio, desplegaremos nuestra solución y empezaremos a obtener rentas positivas (mostrado en la figura 6 por la línea con marcadores circulares).

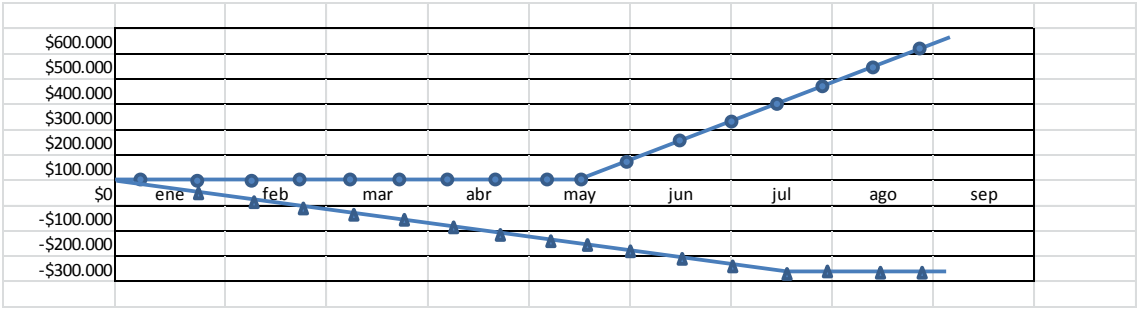


Figura 6: Rentabilidad de la inversión. (Fuente: Elaboración propia)

Cuando añadimos una nueva renta a las figuras del flujo de salida, podemos ver el flujo neto de efectivo (indicado en la figura 7 por la línea con marcadores cuadrados).

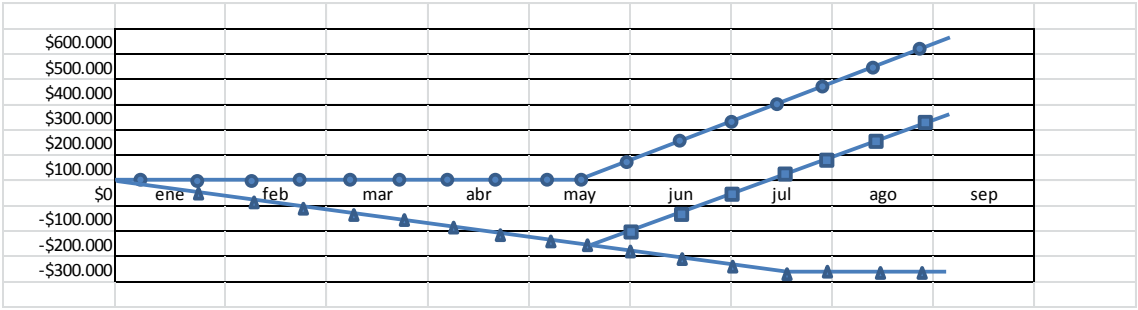


Figura 7: Rentabilidad de la inversión con flujo neto de efectivo. (Fuente: Elaboración propia)

Se está tratando de asumir que cuando este flujo neto de efectivo llega a cero (julio en el ejemplo), habremos conseguido la rentabilidad de la inversión y que de enero a julio es nuestro periodo de amortización. En cierto modo, esto es cierto; habremos recuperado el dinero que invertimos. No obstante, necesitamos considerar el efecto de la inflación. El dinero que se recibe en el futuro, tiene menos valor que el dinero que tenemos, o gastamos, ahora. Esto es especialmente cierto si, para financiar el proyecto, tenemos que pedir prestado el dinero que tendremos que devolver con intereses. Así que, a la hora de determinar el ROI, necesitamos considerar el efecto de la inflación y el coste de pedir dinero prestado.

2.2.1.2. Valor actual neto

Aquí es donde el concepto de valor actual entra en juego. El valor actual es una manera de calcular el valor en términos de a día de hoy de una cantidad futura, dado un ratio de interés asumido o un ratio de inflación. Veamos cómo se calcula el valor presente.

Valor presente.

Valor presente= el valor a día de hoy de una cantidad futura

$PV = FV / (1+r)^n$

Donde:

FV = valor futuro

r = ratio de interés

n = número de periodos de tiempo

Cuál es el valor presente de $200,000 recibido en 4 años a partir de ahora si el ratio de interés es del 7%?

$200,000 / (1+0.07)^4 = \$ 152,579$

Ahora vamos a llevar el concepto de valor actual más allá, al valor actual neto (VAN). El VAN es el beneficio total (ingresos menos gastos) para un flujo de ingresos ajustados al valor de hoy. Las empresas han encontrado los cálculos del valor actual y la capacidad resultante de comparar el dinero que sale y el dinero que entra como algo tan valioso que el VAN es ampliamente utilizado para evaluar la rentabilidad de los proyectos.

Mediante el cálculo del VAN para evaluar el coste del proyecto y las fuentes de ingresos, obtendremos, después del ajuste de la inflación, el punto de rentabilidad de la inversión real y el período de recuperación de la inversión. En los grandes proyectos plurianuales, el plazo de amortización puede ser bastante mayor, sobretodo si el dinero para invertir en proyectos se pide prestado a tasas de interés altas.

2.2.1.3. Tasa Interna de Rentabilidad (TIR)

Ahora vamos a ver el concepto de tasa interna de rentabilidad. El cálculo de la TIR ayuda a simplificar la evaluación de proyectos. En lugar de utilizar un tipo de interés para calcular el valor de una inversión en términos de hoy, utiliza la información y la duración de amortización para calcular el tipo de interés efectivo del proyecto.

La definición de la TIR es la tasa de descuento "en el que las entradas del proyecto (ingresos) y salidas del proyecto (costes) son iguales." En otras palabras, "¿Qué tasa de descuento tengo que usar para desplazar el período de rentabilidad real al final de mi proyecto?" Desde una perspectiva económica, cuanto mayor sea la tasa de descuento, mejor será el proyecto.

Tasa Interna de Rentabilidad

Desde una perspectiva económica, cuanto mayor es la tasa de descuento, mejor es el proyecto. Si el proyecto A tiene una TIR del 25% y un proyecto B tiene una TIR del 18%, se debe elegir el proyecto A para maximizar la tasa de rentabilidad.

2.2.1.4. Ejemplos de cálculo del VAN y de la TIR

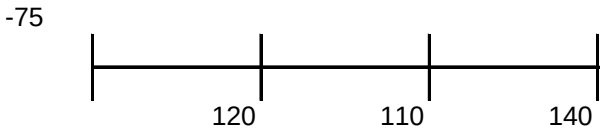
Veamos un ejemplo de cálculo del VAN.

Inversión inicial: $75

Flujos netos de caja:

- Para el primer año: $120
- Para el segundo año: $110
- Para el tercer año: $140

Coste de capital o tasa mínima de rentabilidad exigida para este proyecto: 7%



El Valor Actual de los Flujos netos de caja es $322,15.

VA = 120/(1+0,07) + 110/(1+0,07) 2 + 140/(1+0,07) 3

VA = $ 322,15

VAN = 322,15 – 75 = $ 247

Según el resultado del VAN podremos ver si el proyecto o no rentable:
Si VAN > 0 → el proyecto es rentable.
Si VAN = 0 → el proyecto es rentable también, porque ya está incorporado ganancia de la TD.
Si VAN < 0 → el proyecto no es rentable.

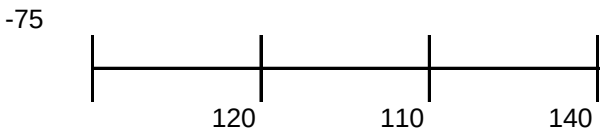
Veamos un ejemplo para el cálculo de la TIR

Partiendo del supuesto anterior, vamos a calcular la TIR del ejercicio y veremos qué relación mantienen ambas variables.

Recordemos la información de dicho supuesto:

Inversión inicial: $75
Flujos netos de caja:

- Para el primer año: $120
- Para el segundo año: $110
- Para el tercer año: $140



La TIR es el tipo de interés que iguala la inversión inicial, $75, con los flujos netos de caja, $120, $110 y $140.

La TIR es de un 149%

$75 = 120/(1+i) + 110/(1+i)^2 + 140/(1+i)^3$

TIR = 149%

Si tenemos en cuenta que la tasa de rentabilidad mínima exigible para este proyecto es de un 7%, es interesante llevar a cabo este proyecto puesto que su TIR es mucho más alta, 149%.

Entre dos proyectos siempre se seleccionará el proyecto con un VAN y una TIR más alta.

La TIR tiene como fin ubicar a la inversión en un rango aceptable, mientras que el VAN es un indicador de cuánto vale el proyecto hoy.
A través de la figura 8 se puede observar la relación existente entre la TIR Y el VAN

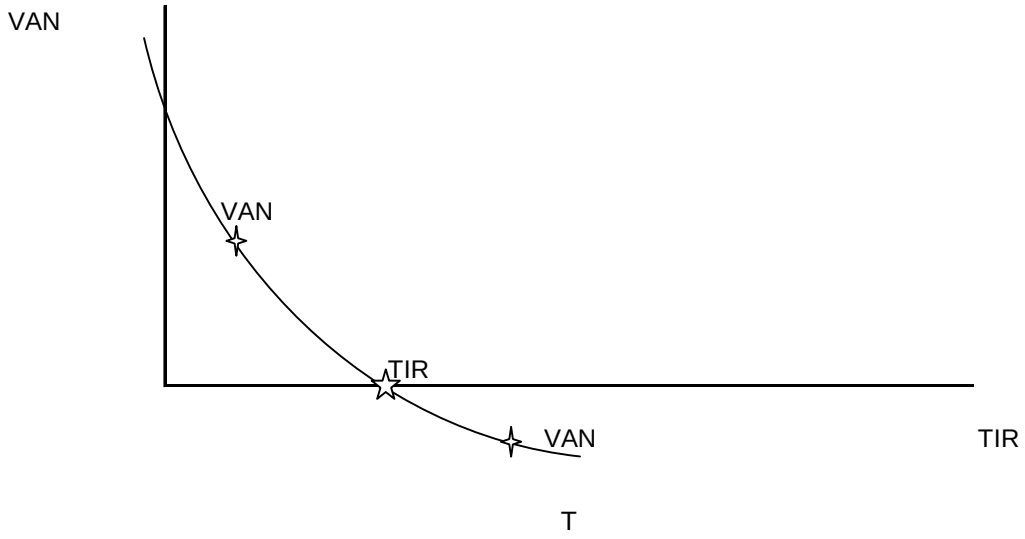


Figura 8: Relación existente entre la TIR y el VAN

En esta figura se observa:
- Existen múltiples VAN: tantos como las tasas de descuento puedan ser utilizadas para actualizar los flujos netos de caja del proyecto.
- Un proyecto de inversión puede ser aceptable para una empresa e inaceptable para otra: aunque los flujos netos de caja esperados de una determinada inversión sean exactamente iguales para distintas empresas, no todas tienen la misma rentabilidad mínima exigida; y por lo tanto no seleccionarán el mismo proyecto.

- También se muestra, que a diferencia del VAN, solo existe una TIR.

La TIR es el tipo de interés para el cual el VAN es igual a “0”.

Las empresas actualmente utilizan más la TIR, pero es más prudente utilizar el VAN, porque da el valor neto.

Cuando los proyectos son mutuamente excluyentes, la regla de decisión de la TIR no sirve para elegir el mejor proyecto. En otras palabras, en este caso, no se puede decir que el proyecto con la mayor TIR es el que debería llevarse a cabo.

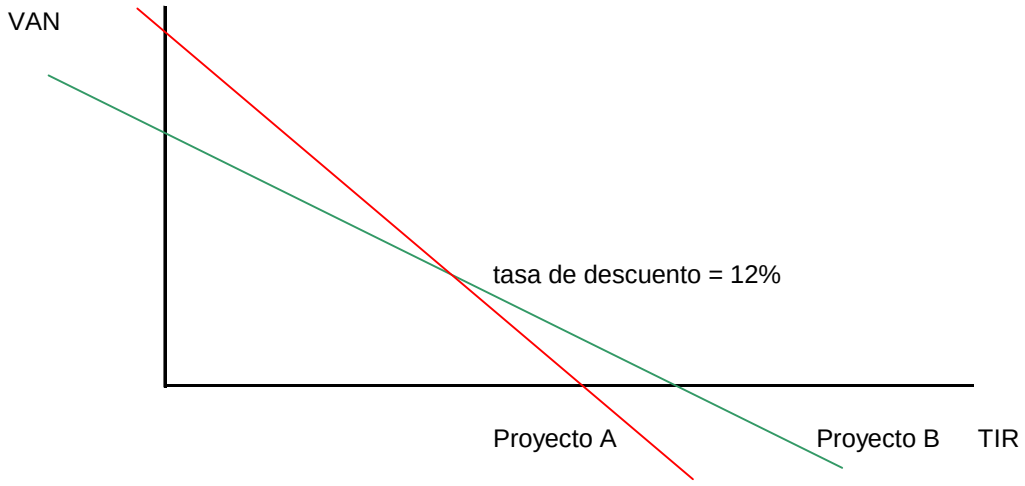


Figura 9: Gráfica proyectos mutuamente excluyentes

Por ejemplo, si tenemos dos proyectos como los de la figura 9 se puede observar que el proyecto B es el de mayor TIR.

Sin embargo, como se puede observar en el gráfico el proyecto B tiene un VAN superior al de A sólo cuando i>12%.

Para tasas de descuento inferiores al 12% (i<12%) se observa que el VAN del proyecto A supera al de B.

En este caso particular, cuando i<12% el criterio de elegir el proyecto con mayor TIR sería contrario al del VAN.

Si los proyectos son mutuamente excluyentes, el criterio de elegir aquel proyecto de mayor TIR podría ser contrapuesto al criterio del VAN. En estos casos, debería utilizarse el VAN como criterio de selección entre proyectos, o sea, elegir aquel proyecto de mayor VAN.

Por último, se puede concluir que por lo general el mejor indicador para decidir que proyectos deben llevarse a cabo es el del VAN: "invertir en aquellos proyectos que tengan VAN positivo" o "seleccionar aquel proyecto de mayor VAN".

2.2.2. Planificación del valor

Ahora que hemos visto las formas de evaluar el valor al seleccionar los proyectos, vamos a ver qué papel juega el valor impulsado por la entrega en la planificación de un proyecto una vez se ha seleccionado. El enfoque en el valor impulsado por la entrega impregna el proceso de planificación ágil. Priorizamos el trabajo a realizar en un proyecto de acuerdo con el valor del negocio, colocando los entregables de alto valor al inicio de la lista, o cartera. Cuando empezamos a trabajar en el proyecto, empezamos con los elementos al inicio de la cartera. Las solicitudes de cambio y las correcciones de defectos, de hecho cualquier trabajo que no estaba previsto inicialmente y que se agrega más adelante, se priorizan en la cartera en función de su valor para el negocio y de si se consideran más importantes que los elementos existentes en la lista.

2.2.2.1. Chartering

La técnica de “Chartering” en los proyectos ágiles tiene el mismo objetivo general que el “Desarrollo del Acta de Proyecto” definido en la guía PMbok, pero el nivel de detalle y el conjunto de supuestos son diferentes. Al igual que un Acta no-ágil, el objetivo de un Acta ágil es describir el proyecto a alto nivel, obteniendo un acuerdo sobre los llamados atributos W5H del proyecto (qué, por qué, quién, cuándo, dónde y cómo) y obtener la autorización para proceder. Sin embargo, ya que los métodos ágiles son de uso frecuente en proyectos donde los requerimientos o tecnología son inciertos o se prevén cambios, el ámbito típicamente está menos definido.

Como resultado, las Actas ágiles suelen tener menos detalles que las Actas no ágiles, son documentos más breves, y se centran más en la forma en que el proyecto se llevará a cabo en lugar de lo que se va construir. Cuando estamos apuntando a un objetivo estático (con requisitos o tecnología invariables), es apropiado para planificar, planificar un poco más, y luego ejecutar. Pero en un proyecto dinámico con un objetivo en movimiento, un alto grado de planificación puede ser inadecuado porque los elementos claves del proyecto son susceptibles de cambiar y hacerlo rápidamente. Para este tipo de proyectos, es necesario estar preparado para realizar ajustes en pleno vuelo y asegurarnos de que tenemos los procesos (por ejemplo, establecimiento de prioridades, demos, retrospectivas, etc) en el lugar adecuado para permitirnos hacer ajustes de manera eficaz y eficiente.

Cualquier elemento de la metodología ágil que pueda ser diferente de un proceso normal de la organización, como, la manera en la que los cambios son aprobados y priorizados en la cartera después de ser aprobados, deben estar claramente delimitados en el Acta. Esto es especialmente importante si los métodos ágiles son de nueva aplicación en la organización y si cualquier parte del enfoque, como el control de cambios e inclusión, se aparta de como se llevaron en el pasado.

Debemos entender que la técnica de "Chartering" en un entorno ágil resulta en un documento flexible que permite al equipo responder a las necesidades y tecnología cambiantes, para finalmente, entregar componentes de alto valor que la organización puede comenzar a utilizar rápidamente.

2.2.2.2. Mapa de flujo de valor

El mapa de flujo de valor es una técnica de fabricación Lean que ha sido adoptada por los métodos ágiles. Esta técnica ilustra el flujo de información (o materiales) necesarios para completar un proceso. Puede ayudar a determinar los elementos de pérdida que podrían ser eliminados para mejorar la eficiencia de un proceso. El mapa de flujo de valor por lo general implica la creación de mapas visuales del proceso (conocidos como mapas de flujo de valor), siguiendo estos pasos:

1. Identificar el producto o servicio que se está analizando.

2. Crear un mapa de flujo de valor del proceso actual, identificando los pasos, colas, retrasos y flujos de información.

3. Revisar el mapa para encontrar retrasos, pérdidas y limitaciones.

4. Crear un mapa de flujo de valor nuevo optimizado, con el fin de eliminar o reducir los retrasos, pérdidas y limitaciones del estado futuro del proceso.

5. Desarrollar un plan de trabajo para crear el estado óptimo.

6. Planificar la revisión del proceso en el futuro para afinarlo y optimizarlo.

Para ilustrar esta técnica, crearemos un mapa de flujo de valor para la compra de un pastel con un amigo. Consideraremos que el proceso consiste en elegir un pastel, esperar en el mostrador de la panadería hasta obtenerlo, el pago del mismo en caja, y caminar hacia casa con el pastel. A continuación, deshacer y cortar el pastel antes de disfrutar de los beneficios del proceso (comer el pastel).

En primer lugar, identificaremos el punto de partida del proceso (quien lo inicia) y el punto final (quien recibe el resultado final).

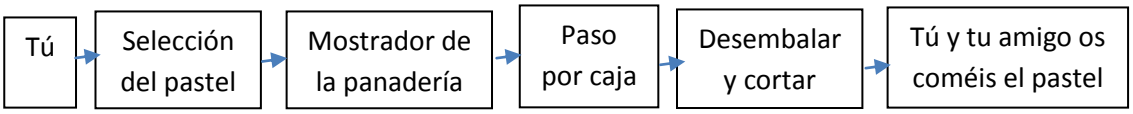
Punto de partida

Tú

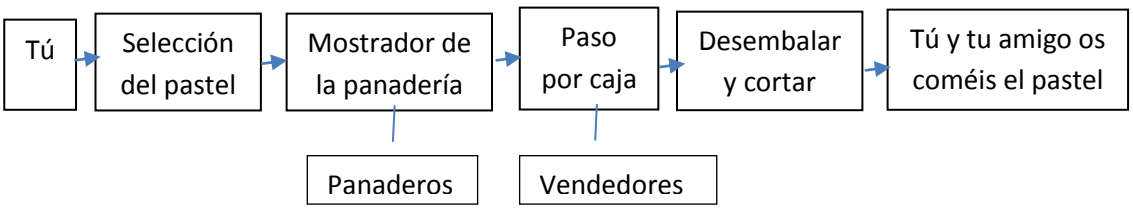
Punto final

Tú y tu amigos os comeis el pastel

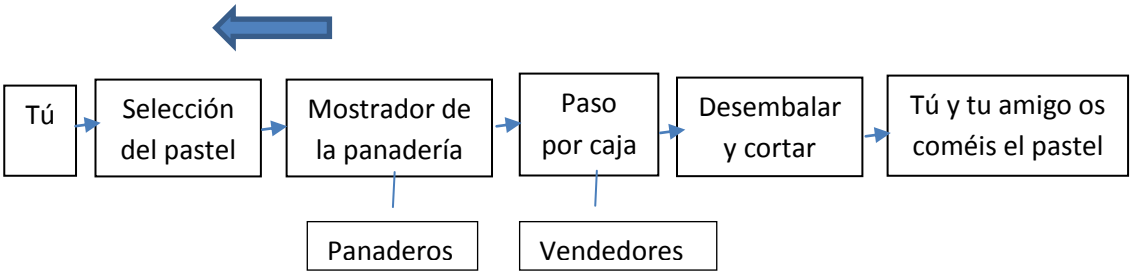
A continuación, se identifican los pasos de alto nivel, los inventarios y las colas a través del proceso, centrándose en el flujo principal.

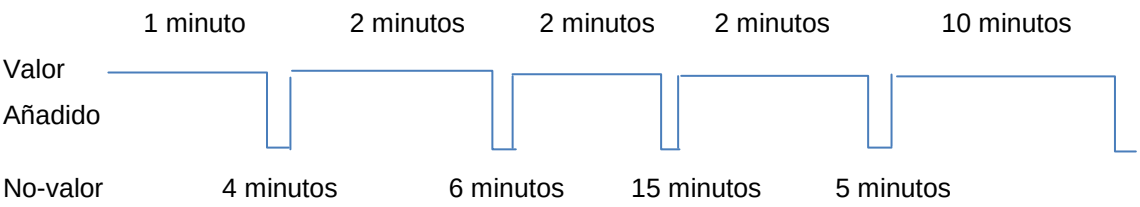


Identificar los grupos de apoyo (tales como los panaderos y los empleados de ventas) y los flujos alternativos (como la selección de otro pastel si el contador de la panadería no tiene el que usted desea).



Medir las actividades de valor añadido y las de no-valor añadido; calcular la eficiencia e identificar las pérdidas, cuellos de botella, y acciones de mejora.





Tiempo total de ciclo = Valor añadido + no-valor añadido Ciclo total de tiempo = 47 minutos

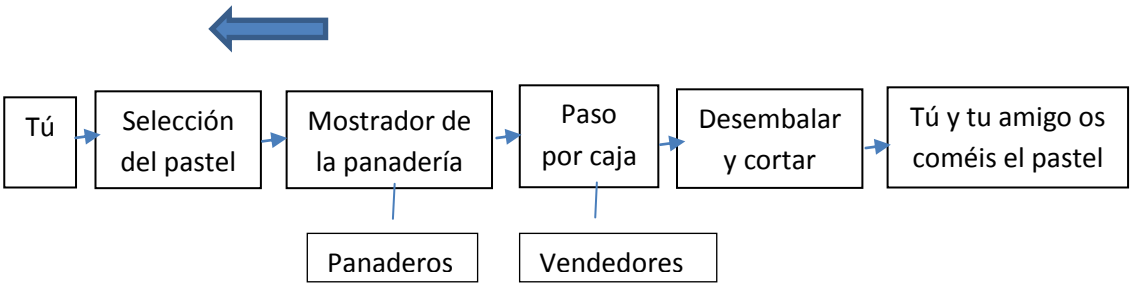
$$\text{Eficiencia del proceso} = \frac{\text{Tiempo total de valor añadido}}{\text{Tiempo total de ciclo}} \quad \text{Eficiencia del proceso} = \frac{17 \text{ minutos}}{47 \text{ minutos}} = 36\%$$

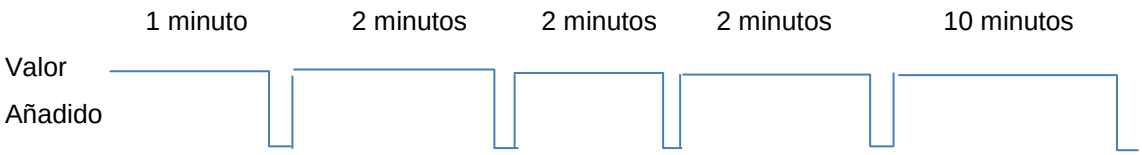
Esto nos lleva al paso 3 del proceso de mapa de flujo de valor, en el cual se revisa el mapa para identificar las pérdidas o las ineficiencias del proceso. Los siguientes pasos tienen como fin eliminar las pérdidas y crear un mapa de flujo de valor de nuevo del futuro estado optimizado, junto con una hoja de ruta de como llegar a ese estado. Para completar la técnica, planificamos en que puntos vamos a revisar el proceso con el fin de ir mejorando el mismo.

En este ejemplo, podemos ver que se esperan retrasos en el mostrador de la panadería y en la caja. También podemos ver que hay ineficiencias en el desplazamiento, con una gran cantidad de tiempo transcurrido entre el pago del pastel y ser capaz de comérselo, ya que hay que viajar de vuelta a casa desde la tienda.

Para diseñar un proceso más eficiente, se podría pensar en telefonear a un servicio de catering especializado en pasteles y obtener el pastel directamente en nuestra casa.

En este nuevo proceso, más eficiente, digamos que nuestro tiempo de espera para realizar el pedido se reduce de 4 minutos a 2 minutos. El tiempo de 6 minutos de espera para pagar el pastel se reduce a 1 minuto, y el desplazamiento de 15 minutos a pie para llegar a casa se sustituye por un plazo de entrega de 10 minutos.





Tiempo total de ciclo = Valor añadido + no-valor añadido Ciclo total de tiempo = 35 minutos

$$\text{Eficiencia del proceso} = \frac{\text{Tiempo total de valor añadido}}{\text{Tiempo total de ciclo}} \quad \text{Eficiencia del proceso} = \frac{17 \text{ minutos}}{35 \text{ minutos}} = 49\%$$

Con estos cambios, el tiempo de valor añadido se mantiene en 17 minutos, pero el tiempo de no-valor añadido se reduce de 47 minutos a 35 minutos, lo que significa que la eficiencia del proceso se eleva del 36 por ciento al 49 por ciento Por supuesto, la eficiencia del proceso no lo es todo - si el correo urgente triplica el coste del pastel, podríamos pensárnoslo dos veces antes de utilizarlo. Sin embargo, este diseño hace maximizar la eficiencia de la operación.

Antes de pasar al tema siguiente, la priorización basada en el valor para el cliente, vamos a profundizar en el concepto de pérdida en los proyectos de software. María y Tom Poppendieck han escrito extensamente sobre el uso de los principios de Lean para proyectos de software.

Han convertido las siete pérdidas tradicionales de un proceso de fabricación en siete pérdidas equivalentes relacionadas con el software, como se indica en la siguiente tabla:

Pérdidas	Descripción	Ejemplo
Trabajo realizado parcialmente	El trabajo se inició pero no se acabó; el trabajo parcialmente hecho puede causar entropía	» Código de espera para la prueba »Especificaciones esperando para el desarrollo
Procesos extra	Trabajo extra que no da valor	» Documentación sin utilizar »Aprobaciones innecesarias
Características extra	Características que no son necesarias, o pensadas como “sería bueno tenerlas”	» Gold-plating, características de la tecnología
Conmutación entre tareas	Multitarea entre varios proyectos diferentes, cuando se sanciona el cambio de contexto	» Gente en varios proyectos
Esperas	Las demoras de espera para las revisiones y aprobaciones	» Esperas para la revisión de prototipos » Esperas para la aprobación de documentos
Movimiento	El esfuerzo necesario para comunicar o mover información o entregables de un grupo a otro, si los equipos no están co-ubicados, este esfuerzo será mayor	» Equipos distribuidos » Handoffs
Defectos	Documentos o software defectuosos que necesitan corrección	» Defectos en los requerimientos » Errores de software

Tabla 5: Pérdidas relacionadas con el software

2.2.2.3. Priorización basada en el valor para cliente

La priorización basada en el valor para el cliente consiste en trabajar en los elementos que generan el mayor valor al cliente lo antes posible. Esta técnica tiene como objetivo involucrar al cliente en el proceso de priorización, en el que el equipo identifica las características de alto valor y las desplaza a la parte superior de la cartera de artículos en los que trabajar. La priorización es esencial para que el equipo pueda ajustar el alcance para cumplir los objetivos presupuestarios o de tiempo, al mismo tiempo que conserva el valioso conjunto de funcionalidades que constituyen la mínima característica comercializable.

El uso de esquemas de priorización basada en el valor para el cliente es un hilo conductor a través de los diferentes métodos ágiles. Aunque la terminología varía a menudo - Scrum, por ejemplo, tiene una "cartera de productos", FDD tiene una "Lista de funciones", y DSDM tiene una "lista de necesidades prioritarias", la idea es la misma.

El proyecto funciona a través de una lista de prioridades de los elementos que tienen valor discernible para el cliente.

Esquemas de priorización

El equipo debe elegir el esquema de prioridades a utilizar en función de las necesidades del proyecto y lo que funciona mejor para la organización. Vamos a ver algunos esquemas de asignación de prioridades comunes.

Esquemas sencillos. Uno de los esquemas más simples es etiquetar elementos como "Prioridad 1", "Prioridad 2", "Prioridad 3, etc… Si bien este enfoque es sencillo, puede ser problemático, ya que la gente tiene tendencia a designar todo como " Prioridad 1" Si demasiados artículos están etiquetados como" Prioridad 1", el sistema se convierte en ineficaz. Los representantes empresariales rara vez pedirán una característica y dirán que debería ser prioridad 2 o 3, ya que saben que las de menor prioridad corren el riesgo de quedar fuera del proyecto. Por las mismas razones, el orden de prioridades "alto," medio "y" bajo " también puede ser problemático. Sin una razón compartida y defendible para definir la prioridad "alta", nos encontraremos con demasiados elementos en esta categoría y una falta de verdadera prioridad.

Esquema de prioridades MoScOw. El esquema de prioridades MoScOw, popularizado por DSDM, es otro ejemplo. Este esquema deriva su nombre de las primeras letras de las siguientes etiquetas:

» “Must have” (debe tener)

» “Should have” (debería tener)

» “Could have” (podría tener)

» “Would like to have, but not this time” (lo querré tener, pero no ahora)

Las categorías utilizadas en MoScOw son más fáciles de identificar y defender que las etiquetas "Prioridad 1" o "alta prioridad" de los esquemas más simples. Los requisitos o características "Must-have" son aquellas fundamentales para el sistema, sin ellos, el sistema no funcionará o no tendrá ningún valor. Las características "Should have" son importantes, por definición, las debemos tener para que el sistema funcione correctamente, y si no están, la solución será probablemente más costosa o complicada. Las características "Could have" son adiciones útiles que añaden valor tangible, y los requisitos “Would like" son peticiones de lo que nos gustaría tener, pero que tienen pocas probabilidades de pasar el corte.

El dinero del Monopoly. Otro enfoque que funciona bien es dar a los patrocinadores dinero del Monopoly igual al importe del presupuesto del proyecto y pedirles que lo distribuyan entre las características del sistema. Este enfoque es útil para identificar la prioridad general de los componentes del sistema, pero se puede ir demasiado lejos si las personas que distribuyen el dinero comienzan a cuestionar las actividades, tales como la documentación, que según ellos añaden poco valor al proyecto. La técnica de dinero del Monopoly es más eficaz cuando se limita a priorizar las funciones de negocio.

El Método de los 100 puntos. El Método de los 100 puntos, originalmente desarrollado por Dean Leffingwell y Don Widrig para casos de uso, es otra manera de priorizar características. En este método, a cada participante se le dan 100 puntos que él o ella puede utilizar para votar a favor de los requisitos más importantes. Las partes interesadas pueden distribuir los 100 puntos de cualquier forma: 30 puntos aquí, 15 puntos allí, o incluso los 100 puntos en un solo requisito, si es la única prioridad de la parte interesada.

Análisis Kano. El análisis Kano, desarrollado en la década de 1980 por Noriaki Kano, también puede servir como esquema de priorización. Esta técnica puede ser utilizada para clasificar las preferencias del cliente en cuatro categorías, a las cuales nos referiremos como encantados / excitados, satisfechos, insatisfechos e indiferentes. Estas categorías pueden ayudar al equipo a entender mejor cómo se relacionan con las necesidades la satisfacción del cliente. El análisis de Kano puede establecer el contexto para consultas respecto a características y ayudar a construir planes de entrega que promuevan la satisfacción al cliente.

Echemos un vistazo a cada categoría con más detalle:

»**Excitados**: Estas características proporcionan algo inesperado, novedoso, o nuevos beneficios de alto valor para el cliente. Por ejemplo, una función de excitación podría ser el mapeo visual de los datos que antes sólo estaba disponible en forma de tabla. Los excitadores producen altos niveles de atención del cliente.

»**Los satisfechos**: Para las funciones que se clasifican como satisfechos, cuanto más, mejor. Estas características aportan valor al cliente. Por ejemplo, una función de información útil podría ser un satisfactor.

»**Insatisfechos**: Éstas son las cosas que harán que al usuario no le guste el producto si no están ahí, pero no necesariamente aumentarán la satisfacción si están presentes. Por ejemplo, la capacidad de cambiar una contraseña en el sistema podría ser un Insatisfechos.

»**Indiferente**: Estas características no tienen un impacto en los clientes de ninguna manera. Puesto que los clientes son indiferentes a ellos, debemos tratar de eliminar, minimizar o diferirlos.

Modelo de priorización de los requisitos. El modelo de priorización de los requisitos establecido por Karl Wiegers es un método matemático más riguroso para calcular la prioridad que los otros esquemas que hemos discutido. Con este enfoque, el beneficio, la penalización, el coste y el riesgo de cada característica propuesta se califica en una escala relativa de 1 (bajo) a 9 (alto). Los clientes valoran tanto el beneficio de tener la característica como la penalización por no tenerla.

Los desarrolladores valoran el coste de producción de la característica y el riesgo asociado con la producción de la misma. Todos los números de cada función se introducen en una fórmula ponderada que se usa para calcular su prioridad relativa.

2.2.2.4. Priorización relativa / Clasificación

Independientemente del esquema de prioridades que se utiliza, la atención se debe poner en el objetivo final - comprender la prioridad de las características. A veces, el esfuerzo que supone arbitrar los esquemas puede restar valor a discusiones importantes sobre el establecimiento de prioridades. Por esta razón, es preferible simplemente pedir a la empresa una lista de características en orden de prioridad – nada de categorías 1,2 o 3; alta, media o baja; must haves, etc. Una simple lista (ya sea en una hoja de cálculo o en una herramienta de gestión de requisitos ágil) elimina las categorías, en las que se tiende a fijar la vista, del debate y permite que el centro de la discusión sean las prioridades.

Veamos un ejemplo de una lista de prioridades simple en la figura 10

Lista priorizada

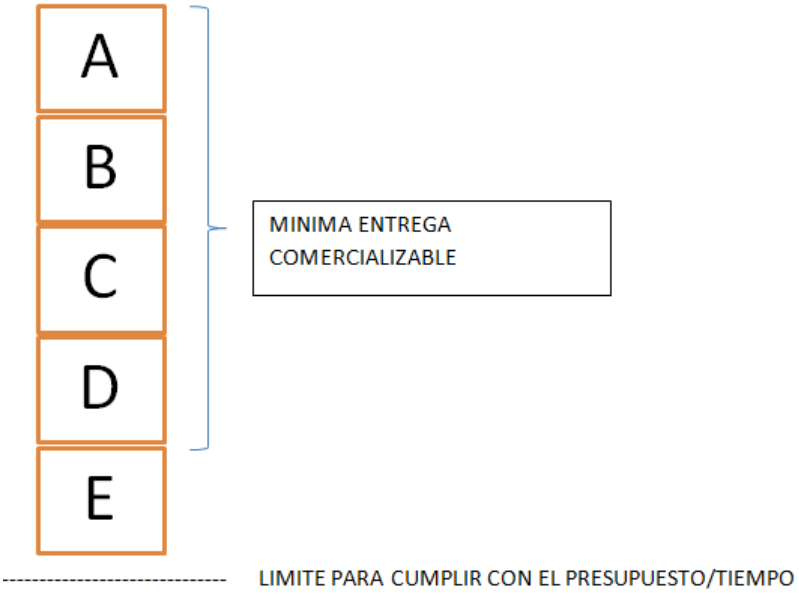


Figura 10: Lista priorizada. (Fuente: Elaboración propia)

En la figura 10 anterior, las características se enumeran en orden de prioridad. Los elementos de la parte superior de la lista - características de la A a la D - forman parte de la entrega comercial definida mínima. Si los objetivos deben ser recortados para cumplir con los objetivos de presupuesto y el calendario, está claro mirando la lista que se deben hacer ajustes en el punto E.

La priorización relativa también proporciona un marco para decidir si y cuándo incorporar cambios. Cuando se solicita un cambio, el equipo puede pedir a los representantes de las empresas, "¿Qué elementos son más importantes que este cambio?" El nuevo cambio se podrá insertar en la lista de prioridades de trabajo en el lugar correspondiente, como se muestra en la figura 11.

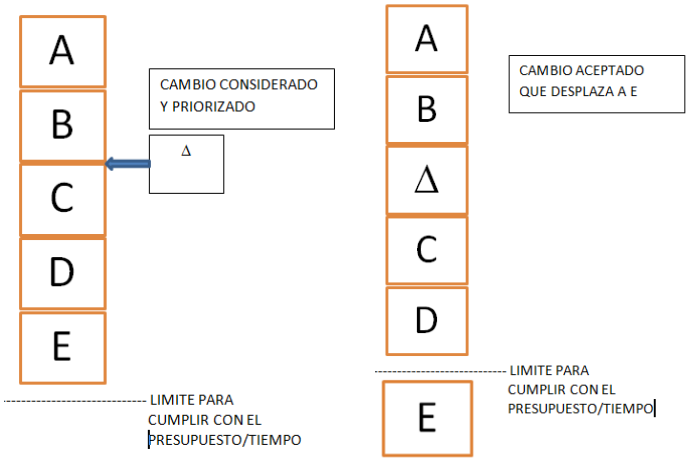


Figura 11: Cambios en lista priorizada. (Fuente: Elaboración propia)

Mientras que los métodos ágiles proporcionan una gran flexibilidad en la aceptación de cambios de última hora, no pueden violar las leyes del tiempo y del espacio. Así que si se estimó que el tiempo o presupuesto del proyecto sería completamente consumido por el actual campo de aplicación, añadir un nuevo cambio inevitablemente hará que una de las funciones de más baja prioridad quede por debajo del" punto de corte "de lo que esperamos entregar. Por lo tanto, sí, podemos aceptar cambios de última hora en los proyectos, pero siempre a expensas de los elementos de baja prioridad.

Una lista única priorizada de trabajo es útil para simplificar la visión de todo el trabajo restante, en lugar de tener diferentes "cubos" de trabajo que representan las solicitudes de cambio, corrección de defectos y nuevas características, una lista única priorizada combinando estos elementos da a todos una visión clara y completa de todo lo que queda por hacer en el proyecto. La primera vez que se adopta un enfoque ágil, muchos equipos se pierden en este punto y conservan cubos separados para correcciones de errores y cambios. Sin embargo, esta separación enturbia la velocidad.

Una lista única priorizada de trabajo a ser realizado, independientemente de su origen, ofrece una mayor transparencia y control sobre las ventajas y desventajas, porque la prioridad relativa del trabajo está claramente mostrada por la posición que cada elemento ocupa en la lista.

Aunque estoy de acuerdo con el uso de una lista simple, no hay un único método mejor a utilizar para priorizar las características de cada proyecto. Cualquiera que sea la técnica que utilice, debe tratar de diagnosticar cualquier problema que surja en el proceso de priorización, ya sea la "falta de participación" o "demasiadas prioridades". A continuación, se pueden añadir diferentes enfoques, tales como el uso de dinero del Monopoly, aplicando el esquema de prioridades M0SC0W, o simplemente enumerando las características en orden de prioridad, para comprobar si se pueden resolver los problemas al enmarcar la discusión de una nueva manera. El objetivo de estos esfuerzos es entender como las características se relacionan unas con otras, en lugar de asignar simplemente una etiqueta de categoría. Con esto se obtendrá una lista flexible de requisitos de prioridad que puede ser revisada y re priorizada según sea necesario. Todo esto permite mantener la agilidad necesaria para ofrecer el conjunto de características de alto valor en el tiempo y presupuesto disponibles.

2.2.2.5. Hoja de Ruta de producto

Una hoja de ruta de producto es una descripción visual de las entregas de un producto y sus componentes principales. Se trata de una herramienta de comunicación que ofrece a los stakeholders una vista rápida de los puntos principales de entrega y la funcionalidad deseada. No hay una única manera de describir una hoja de ruta de producto, pero los mapas de historial, popularizados por Jeff Patton, son una manera de hacerlo comúnmente utilizada.

Mapas de historia

Los mapas de historia ayudan a seleccionar y agrupar las características para una entrega. Estos diagramas muestran la secuencia de las características e indican su importancia para el proyecto al clasificarlas como "backbone", "walking skeleton", o características adicionales. Todos los productos y sistemas tienen algunas funciones esenciales que deben realizar. Esta funcionalidad esencial constituye la columna vertebral (backbone) del sistema. No tiene mucho sentido priorizar demasiado estas funciones, sólo necesitan estar en el sistema. Colgando de la columna vertebral está la funcionalidad de “walking skeleton”. “Walking skeleton” describe el sistema más pequeño que podría funcionar. Finalmente, cualquier función cuelga por debajo del “walking skeleton” y se prioriza según su importancia en el sistema.

Una vez que las funciones se colocan en el mapa de acuerdo a su importancia y su secuencia, las prioridades del cliente se equilibran con la capacidad del equipo para obtener resultados, y las entregas que componen la hoja de ruta del producto son identificados.

Con este enfoque, la hoja de ruta del producto final consiste en grupos de mapas de historia y muestra que entrará en cada entrega. Representa lo que es absolutamente esencial para el sistema (backbone) y necesario para la solución (walking skeleton), junto con otras características que conformarán entregas posteriores.

2.2.2.6. Cartera Ajustada al riesgo

Como se ha comentado al principio, los riesgos negativos (amenazas para los proyectos) son como “anti-valor”, si se produce un riesgo, se necesita utilizar tiempo y recursos de los elementos que si aportan valor y pone en peligro los beneficios del proyecto. Por lo tanto, no debemos únicamente realizar una planificación para entregar características de alto valor de manera rápida, sino que también debemos planificar la ejecución de actividades para evitar y mitigar riesgos de forma temprana.

La gestión de riesgos puede parecer un proceso tradicional, que no va a funcionar bien en un entorno ágil. Sin embargo, los métodos ágiles son grandes vehículos de reducción de riesgo y en realidad son muy adecuados para la rápida identificación y reducción de los riesgos. Vamos a explorar la gestión del riesgo en más detalle, comenzando con la definición de que es un riesgo.

Un riesgo es un evento o circunstancia que puede impactar en el proyecto. La mayoría de lo escrito sobre el riesgo se centra en los acontecimientos que tienen el potencial de impactar negativamente o amenazar el proyecto, pero la Guía PMBOK también habla de riesgos positivos u oportunidades, para el proyecto.

La figura 12 representa el proceso de gestión del riesgo que se describe en la Guía PMBOK.

Gestión de la planificación de riesgos.

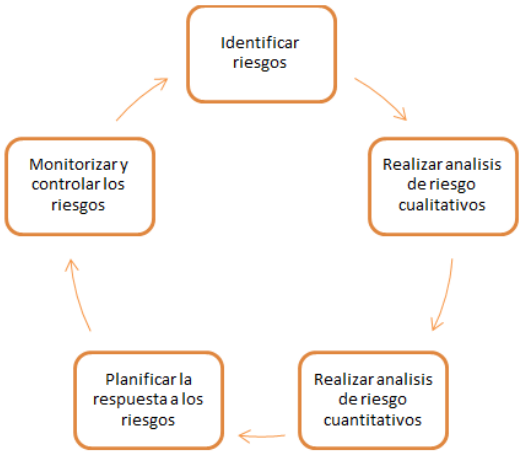


Figura 12: Proceso de gestión del riesgo. (Fuente: Elaboración propia)

Un paso que parece estar ausente de este proceso es el de "respuesta al riesgo realizada", un paso en el que las acciones identificadas en el plan de respuesta al riesgo se han completado. Este paso realmente no falta, estas actividades se han programado con las actividades de trabajo regulares del proyecto. Sin embargo, la aparente falta de un paso "realizado" refleja un problema visto en muchos proyectos - a saber, que la gestión del riesgo se lleva a cabo como una actividad pasiva independiente (a veces sólo una vez) que no lleva suficiente acción al proyecto para prevenir que estos riesgos sucedan. Como resultado, vemos los riesgos que ocurren y vemos donde fueron identificados en la lista de riesgos, y sin embargo, no se hizo lo suficiente para evitarlos.

La naturaleza iterativa de proyectos ágiles ayuda a superar este problema. A través de iteraciones, podemos abordar áreas de alto riesgo del proyecto más temprano que tarde.

Este enfoque pone los problemas al descubierto, mientras aún hay sitio en el programa y presupuesto para trabajar en ellos. También reduce la cantidad de esfuerzo invertido en trabajos que pueden terminar siendo descartados.

Se dice que los proyectos ágiles son proyectos tanto de negocio enfocado al valor como basados en el riesgo. Esto significa que elegimos paquetes de trabajo basándonos en la prioridad comercial asignada y qué los artículos de alto riesgo permanecen en la lista de características prioritarias (o historia de cartera del usuario). Así que si tenemos una historia con riesgos residuales, querremos mover los elementos de riesgo a la parte alta de la cola y tratarlos según corresponda.

La mayoría de los equipos se sienten cómodos con los requisitos de clasificación (historiales, características, casos de uso) por valor de negocio y niveles de riesgo. Este proceso es a menudo subjetivo, basándose en los presentimientos del cliente o usuario. Si desea obtener algo más objetivo, nos podemos centrar en la rentabilidad de la inversión por cada función. Este cálculo empieza por comprender el rendimiento de la inversión para el proyecto en su conjunto. Por ejemplo, antes de aprobar un proyecto de $ 2 millones, una empresa (presumiblemente) hace un análisis de costo-beneficio para determinar si los ingresos o ahorros que se pueden obtener del proyecto justifican el costo de llevarlo a cabo, y calcula el rentabilidad de la inversión del proyecto (por ejemplo, $ 4 millones en tres años). Para proyectos de cumplimiento regulatorio o de mantenimiento, todavía podemos llegar a la figura del dólar, la cantidad en dólares se relacionará con las sanciones económicas en caso de incumplimiento (multas, pérdida de negocios, etc.).

Una vez que ya tenemos una figura de rentabilidad de la inversión para un proyecto, los representantes de las empresas (no el equipo de desarrollo) deben distribuir o prorratear esta cantidad a través de las características identificadas. Los representantes empresariales a menudo no aceptan esta idea con comentarios como: "No se puede poner un valor monetario a un informe de ventas, eso sería demasiado subjetivo o engañoso." Puede ser útil recordarles que alguien pensó en una cifra para ingresos por la inversión en ese proyecto. Si esa cifra no puede ser dividida a través de las características de los productos, entonces ¿de dónde vino en primer lugar?

Después de que los representantes del negocio hayan atribuido (tal vez de forma arbitraria) un valor en dólares a las características del producto, tendremos una lista con las características priorizadas con los valores de rentabilidad de la inversión, como se muestra en la figura 13.

Figura 13: lista priorizada de requisitos. (Fuente: Elaboración propia)

Ahora necesitamos una manera de contabilizar todos los requisitos para anular y mitigar el riesgo. Para obtener esto, podemos calcular el valor monetario esperado del riesgo. Para el cálculo de una muestra, supongamos que el utilizar una herramienta de reporte interna no tiene costes para el proyecto., pero que adquirir una herramienta externa de reporte de alto rendimiento, tiene un coste de 10.000 dólares. Hay una posibilidad del 50% de que acabemos necesitando la herramienta de alto rendimiento. Con estos datos, podemos utilizar la siguiente fórmula para calcular el valor monetario esperado.

Valor monetario esperado (EMV) = Impacto del riesgo (dólares) x Probabilidad del riesgo (en porcentaje).

En nuestro ejemplo, EMV = 10.000 x 50% = 5.000 dólares. Este dato, nos ayuda a explicarle a los sponsors del proyecto que creemos que el valor económico para la compañía de mitigar este riesgo es de 5000$; por lo tanto, la prioridad de desarrollo de su acción de respuesta debe estar a la par con las características funcionales que se valoran a 5.000 $.

El cálculo del EMV puede realizarse para muchos riesgos. Los riesgos técnicos tienen un coste de compra asociado (p.e $10.000 para una herramienta de reporte de alto rendimiento) o penalización por tiempo (p.e. empleará a dos desarrolladores y tres semanas adicionales) que pueden ser trasladados a una cantidad en dólares. Los recursos humanos y los riesgos del negocio pueden de la misma forma estimarse en términos monetarios. Tendemos a pensar que no podemos asignar un valor monetario a ciertas cosas, pero recordemos que las compañías de seguros son capaces de determinar el valor de un dedo perdido o la angustia emocional para sus clientes. El mismo concepto se aplica para asignar un valor de riesgo a un proyecto. Es importante tener en mente que estamos buscando números relativos, más que precisos. En muchos casos, el valor inicial de la rentabilidad de la inversión que el negocio establece para justificar el proyecto, es sospechoso de todos modos, por lo que no debemos quedarnos atascados en el nivel de precisión de los valores del riesgo o intentar crear una hoja de balance perfecto del riesgo. En lugar de eso, nos debemos centrar en llegar a números generales y justificables que tengan el consenso de todos los stakeholders del proyecto para usarse como bases de priorización.

Utilizando este enfoque, podemos clasificar los riesgos del proyecto en una lista de prioridades, ordenados por severidad del riesgo, como se muestra en la siguiente tabla:

Lista de riesgo priorizada
Riesgo 1 $9.000 x 50% = $4.500
Riesgo 2 $8.000 x 50% =$ 4.000
Riesgo 3 $3.000 x 50% = $1.500
Riesgo 4 $6.000 x 25% = $1.500
Riesgo 5 $2.500 x 25% = $625
Riesgo 6 $500 x 25% = $125
Riesgo 7 $500 x 20% = $100

Tabla 6: Lista de prioridades

Naturalmente no todos los riesgos tienen pasos que podamos programar dentro del proyecto de forma que podamos evitarlos o mitigarlos. Muchos riesgos deben ser aceptados (p.e. “estamos esperando al Service Pack 2”) o transferidos (p.e “Hemos contratado un seguro). Para los riesgos que pueden ser abordados de manera proactiva, no obstante, las acciones de respuesta pueden ser priorizadas a lo largo de las características funcionales. Cuando lo combinamos, tendremos una cartera de riesgos ajustados.

Atribuir la figura del dólar tanto a los requisitos funcionales como a los no funcionales permite al equipo tener una discusión más significativa con los sponsors y usuarios. Por ejemplo, en la anterior figura, podemos ver como el riesgo 2 tiene un valor monetario esperado de $4.000; como resultado, cuando llega la hora de seleccionar los requisitos o características de una nueva iteración, la acción de respuesta al riesgo que parecía no tener un valor comercial de peso cobra sentido ya que podemos mostrar que es de igual valor para la empresa.

De nuevo, no nos centramos realmente en la precisión de los números. Una conjetura multiplicada por una conjetura es poco probable que sea muy precisa. En lugar de eso, el propósito de todo esto, es facilitar mejores discusiones con los representantes del negocio sobre como secuenciar mejor los elementos del trabajo. Si las personas empiezan a manipular las cifras para satisfacer sus agendas personales, el propósito y el verdadero valor del proceso puede perderse, y las dependencias técnicas y las órdenes de los sponsors tendrán prioridad sobre las secuencias derivadas de este enfoque. Por lo tanto, se debe pensar en la cartera de riesgo ajustado como una herramienta que utiliza números únicamente para obtener lo que es realmente importante – la prioridad necesaria de los elementos de trabajo.

Los verdaderos beneficios de esta herramienta es que ayuda a crear un puente entre los representantes del negocio y el equipo de desarrollo para tener discusiones significativas sobre la programación y alcance del negocio. Por lo que aunque asignemos valores numéricos para ayudar a nivelar el campo de juego, crear una cartera de riesgo ajustado es realmente más una práctica cualitativa que cuantitativa.

2.2.2.7. Contratación ágil

Ahora que hemos visto las herramientas y técnicas utilizadas para planificar el valor de los proyectos ágiles, ¿cómo funciona esta agilidad cuando un proyecto se realiza en base a un contrato? Mientras que los métodos ágiles proveen gran flexibilidad y nos permiten manejar cambios en los requisitos y prioridades, esta adaptabilidad y flexibilidad en el alcance puede crear problemas cuando se definen los criterios de aceptabilidad para los contratos y cuando se externaliza el trabajo.

Estos desafíos han existido desde la creación de los métodos ágiles. Afortunadamente como resultado, se ha puesto un gran esfuerzo en su resolución. En 1994, la primera edición del manual DSDM presentaba el modelo del triángulo invertido, que se muestra en la figura 14.

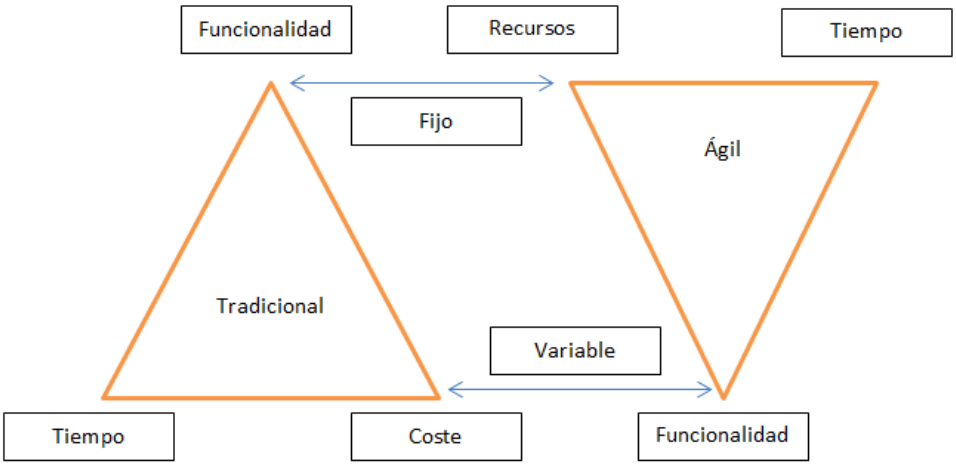


Figura 14: Módelo del triángulo invertido. (Fuente: Elaboración propia)

Los proyectos ágiles intentan fijar los recursos y tiempo (que son componentes claves del coste) y variar la funcionalidad para conseguir la más alta de las prioridades, el producto con la mayor calidad posible dentro de las limitaciones existentes. En contraste, cuando la funcionalidad está fijada, hay un riesgo de que el proyecto se quede sin dinero o tiempo – o aún peor, se cree un producto de baja calidad.

Esta idea de potencialmente no entregar toda la funcionalidad puede no sentar bien a todo el mundo. Quieren una estimación inicial de la entrega del producto completo, no un subconjunto de las partes fáciles; también quieren un contrato escrito para mostrar ante el vendedor o gerente del proyecto. No obstante, para que cualquier proyecto, basado en el conocimiento de los trabajadores, logre esto (p.e. completa funcionalidad dentro del coste y calendario programados), se deben realizar muchos análisis de coste y de consumos de tiempos, se deben producir especificaciones cuidadosamente elaboradas, y las contingencias importantes deben ser incorporadas al contrato para acomodar los cambios razonables e imprevistos, incidencias técnicas y contratiempos. Los clientes pueden tener un alcance fijo y unas estimaciones firmes, pero con un coste – las estimaciones pueden inflarse como un amortiguador para la incertidumbre, y el cliente puede llegar a pagar por actividades que no añadirán mucho valor al negocio después de su salida.

El objetivo de los métodos y los contratos ágiles es la estrecha cooperación entre el equipo de proyecto y el cliente o negocio. Esta cooperación ayuda a redirigir los esfuerzos del equipo hacia la entrega de características con valor añadido. Este objetivo está representado en el tercer valor del Manifiesto Ágil, que indica “Colaboración con el cliente por encima de la negociación del contrato”. Un enfoque ágil requiere una mayor confianza entre las partes que los enfoques tradicionales, pero centra los recursos en lo que se está tratando de construir, en lugar de atascarse en debates sobre como los cambios deben ser negociados o como es realmente el criterio de finalización.

Un enfoque ágil también requiere que la empresa esté más involucrada en proveer feedback en cada una de las iteraciones del proyecto, re priorizando la cartera y evaluando el valor de las peticiones de cambios de los ítems de trabajo restantes. Para clientes de confianza, los contratos ágiles son grandes herramientas para extraer más valor y dan a los clientes una ventaja competitiva. Para clientes no de confianza los contratos ágiles pueden ser difíciles de vender y no muy apropiados.

Hay diferentes formas en las que los contratos ágiles se pueden estructurar. Veremos rápidamente algunas de estas maneras.

Contrato DSDM

El contrato DSDM fue encargado por el consorcio DSDM y continúa evolucionando. Este contrato se centra en el trabajo "ajustado al propósito del negocio" y en pasar las pruebas, en lugar de en cumplir con una especificación. El contrato DSDM se usa principalmente en el Reino Unido y otras áreas de Europa.

Dinero por nada y Cambios gratuitos.

Jeff Sutherland también promueve una estructura para contratos ágiles en su popular presentación "Dinero para nada y Tus Cambios gratuitamente". Sutherland sugiere incluir opciones de finalización tempranas y propone un modelo que permite flexibilidad a la hora de realizar cambios. La estructura de Sutherland comienza con un precio estándar fijo para el contrato que incluye tiempo y materiales para trabajos adicionales, pero inserta una opción de cláusula de "cambios gratuitos".

El cliente únicamente puede utilizar esta cláusula si trabaja con el equipo en cada una de las iteraciones. En caso de no comprometerse a esto, la cláusula queda anulada y el contrato vuelve al tiempo y materiales. Asumiendo que el cliente se compromete, el propietario del producto puede re priorizar la cartera al final de una iteración, y los cambios son gratuitos si el contrato de trabajo total no se cambia. Esto permite que las nuevas características se añadan de forma gratuita si otros ítems de menor prioridad que requieran un tiempo igual o mayor son eliminados como se puede ver en la figura 15.

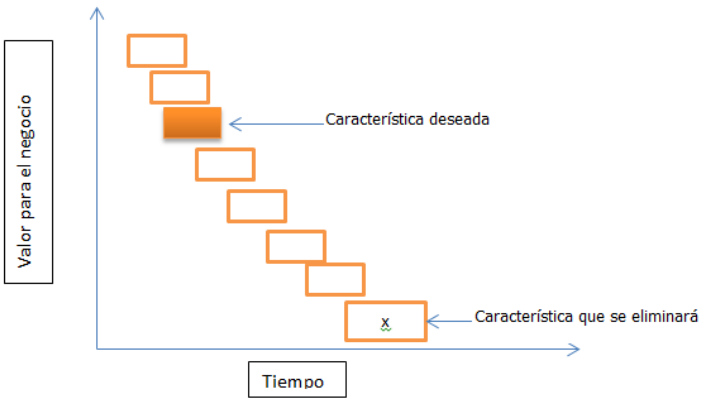


Figura 15: Cambios sin coste. (Fuente: Elaboración propia)

Al igual que la cláusula de "cambio gratuito", el concepto de "dinero por nada" únicamente es válido si el cliente realiza su parte en el proyecto ágil. "Dinero por nada" permite al cliente finalizar el proyecto anticipadamente cuando crea que la rentabilidad de la inversión que resta en la cartera no es la suficiente para garantizar futuras iteraciones.

Por ejemplo, el vendedor puede permitir la finalización del contrato en cualquier momento por un 20 por ciento del valor del contrato restante.

Este 20 por ciento del valor restante del proyecto, es la parte que corresponde al "dinero por nada" y ayuda a evitar el riesgo de tener un equipo parado mientras que el vendedor paga sus salarios. Utilizando el enfoque de "dinero para nada" el cliente obtiene los valores de más alta prioridad para su proyecto, los proyectos siempre finalizan anticipadamente, y se mantienen unas buenas relaciones con el vendedor. Adicionalmente, el sistema permanece eficiente sin necesidad de inflar el código (para proyectos de software) o de características innecesarias. El acuerdo permanece abierto a entregas adicionales si se requiere en los ratios contratados de tiempo y materiales.

Contrato de precio fijo graduado.

Los contratos de precio fijo graduado son promovidos por Thorup y Jensen como otro tipo de contrato ágil. Con este tipo de contrato, ambas partes comparten parte del riesgo y la recompensa asociados con las variaciones en el calendario. Thorup y Jensen sugieren utilizar diferentes ratios temporales basados en entregas anticipadas, a tiempo o tarde. Por ejemplo:

Finalización del proyecto	Ratio graduado	Coste total
Finalización anticipada	$110/hora	$92.000
Finalización a tiempo	$100/hora	$100.000
Finalización tarde	$90/hora	$112.000

Tabla 7: Contrato de precio fijo graduado

Bajo este acuerdo, si el vendedor entrega a tiempo, obtiene el pago por las horas trabajadas al precio estándar. Si entrega anticipadamente, obtiene el pago de menos horas pero a un precio superior. El cliente está feliz porque el trabajo se realiza antes y paga menos en global por él. El vendedor está feliz porque obtiene un margen mayor. No obstante si el vendedor entrega tarde, se le pagará por más horas, pero a un precio menor. Cuando esto ocurre, ambas partes están, de alguna manera, descontentas ya que están ganando menos dinero, pero a un ratio gradual y sostenible, que con suerte no acabará en un abandono del proyecto.

Paquetes de trabajo a precio fijo.

Otro enfoque para establecer un contrato ágil, es establecer paquetes de trabajo a precio fijo. Los paquetes de trabajo a precio fijo mitigan el riesgo de sobrestimar o subestimar una parte del trabajo reduciendo la extensión y costes involucrados en el trabajo a ser estimado. Por ejemplo, Marriott Internacional rompió sus declaraciones de trabajo (SOW) en paquetes individuales de trabajo, cada uno con su precio fijo.

Conforme avanzaba el trabajo, el vendedor podía re-estimar los paquetes de trabajo restantes en el SOW basándose en nueva información y nuevos riesgos.

Este enfoque permite al cliente re-priorizar el trabajo restante basándose en la evolución de los costes, y da al contratista la habilidad de actualizar sus costes tal y como aparecen, eliminando la necesidad del contratista de crear un exceso de fondo para contingencias en el coste del proyecto. Los cambios se localizan entonces en pequeños componentes (p.e. paquetes de trabajo), y si se requieren fondos extra, es fácil identificar la necesidad y justificarla. La siguiente figura ilustra las diferencias entre un SOW tradicional y uno basado en paquetes de trabajo de precio fijo.

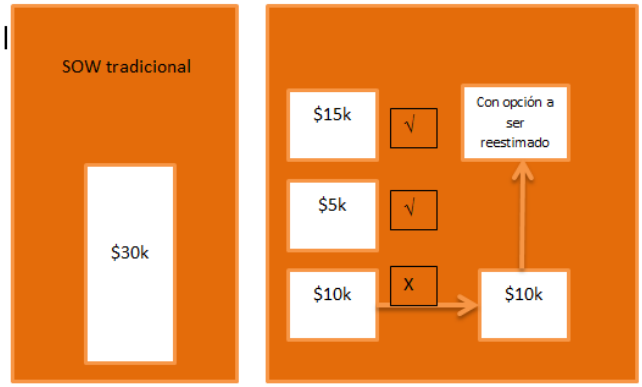


Figura 16: diferencias entre un SOW tradicional y uno basado en paquetes de trabajo de precio fijo. (Fuente: Elaboración propia)

Contratos personalizados.

Todos estos diferentes enfoques ágiles a los contratos pueden unirse para crear un contrato personalizado que beneficie tanto al cliente como al vendedor. Con estos contratos, el cliente mantiene la flexibilidad de poder re-priorizar el trabajo y el vendedor no es penalizado por compartir la información sobre el incremento de los costes. Además eliminan el incentivo para el vendedor de añadir grandes cantidades de costes para contingencias al precio del proyecto. Combinando elementos de un contrato de precio fijo graduado y paquetes de trabajo a precio fijo e incorporando los conceptos de finalización anticipada (dinero para nada) y la re-priorización (cambios gratuitos), podemos crear un contrato que proteja a ambas partes y fomente un comportamiento positivo.

Aunque los contratos ágiles pueden ser altamente beneficiosos para ambas partes, debemos recordar que crear un contrato entre una parte que quiere minimizar el coste de un producto o servicio y otra parte que está en el negocio para maximizar sus ingresos, es siempre complicado.

En los proyectos ágiles, las adquisiciones han sido siempre particularmente desafiantes, ya que el ámbito no está siempre totalmente definido de primeras en el proyecto. Además, la naturaleza intangible de los proyectos, como el software, puede hacer difícil evaluar y recibir la aceptación del trabajo. Cualquier tipo de adquisición – ya sea para contratos ágiles o tradicionales – funciona mejor cuando ambas partes quieren resultados exitosos que lleven a futuros trabajos. El éxito de un proyecto se determina en última instancia por el nivel de colaboración sobre la marcha entre el cliente y el vendedor. Los contratos ágiles no pueden generar o reforzar la colaboración, pero con trabajo y creatividad, pueden estar finalmente mejor estructurados para permitirla.

2.2.3. Entregando valor

Ahora que hemos visto que hay dentro de la planificación y la evaluación del valor en los proyectos ágiles, veremos que significa entregar valor. El objetivo de entregar valor se enfatiza a través de la ejecución de los proyectos ágiles. Para cumplir el objetivo, el equipo debe aplicar los conceptos “Lean” de maximizar las actividades de entrega de valor y minimizar las actividades que no aportan valor. Por ejemplo, las actividades que pueden ser necesarias para una organización pero no están directamente centradas en la entrega de valor, como el tiempo utilizado en reportar o seguir, debería ser reducido o transferido mientras un proyecto se está ejecutando.

Cuando el objetivo es maximizar el valor, es útil revisar las siete pérdidas relacionadas con el software de Poppendiecks, que veremos más adelante.

Pérdida	Descripción	Ejemplo
Trabajo parcialmente hecho	Trabajo empezado pero no completado; el trabajo parcialmente hecho posibilita la entropía	- Código esperando para ser probado - Especificaciones esperando a ser desarrolladas.
Procesos extra	Trabajo extra que no añade valor	- Documentación no utilizada - Aprobaciones innecesarias
Características extra	Características que no son necesarias, o que son pensadas como que “sería bueno tenerlas”	- Chapado en oro - Características tecnológicas

Pérdida	Descripción	Ejemplo
Cambio entre tareas	Multitarea entre diferentes proyectos cuando hay penalizaciones por el cambio	- Personas en múltiples proyectos
Esperas	Retrasos esperando por revisiones o aprobaciones	- Esperando la revisión de un prototipo - Esperando la aprobación de un documento
Movilidad	El esfuerzo requerido para comunicar o mover información o entregables de un grupo a otro; si los equipos no están cerca unos de otros, este esfuerzo puede necesitar ser grande	- Equipos distribuidos - Handsoffs
Defectos	Documentos o software defectuosos que necesitan correcciones.	- Defectos en la requisitos - Errores de software

Tabla 8: Siete pérdidas relacionadas con el software de Poppendiecks

Donde veamos actividades en nuestros proyectos que sean inútiles, podemos aspirar a eliminarlas. Vamos a ver algunas herramientas y técnicas que pueden clarificar el proyecto y ayudar a identificar y eliminar actividades residuales.

2.2.3.1. Tablas de tareas y Kanban

Las tablas de tareas y Kanban pueden ayudar a entregar valor en los proyectos ágiles de diversas maneras. Mientras que puede parecer un paso atrás pasar de sofisticadas herramientas de software para la planificación a unas simples tarjetas en un muro, esta transición ofrece muchos beneficios.

Las características técnicas que los diagramas de Gantt ofrecen, pueden presentar algunas desventajas en los proyectos ágiles. El software de planificación puede ilustrar profundamente las jerarquías entre las tareas, ayudar a las comprobaciones sobre la integridad de dependencia de las tareas, y calcular indicadores interesantes, como el slack, los costes de subensamblaje y utilización de los recursos.

Sin embargo, la sofisticación técnica de estas herramientas es la principal causa por la que no son ideales para los métodos ágiles. Las matemáticas, estadísticas e informes cuantitativos que pueden obtenerse con estas herramientas disfrazan la naturaleza volátil de lo que está siendo analizado – estimaciones y tareas del proyecto. Además las herramientas de planificación sofisticadas pueden alejar al equipo y desalentar la colaboración del equipo completo.

Cuando utilizamos herramientas que realizan cálculos de planificación y precisión, nos encontramos dos problemas: la percepción de la precisión de los datos aumenta, y se crean barreras para la interacción con los stakeholders del proyecto. Veamos estos proyectos en detalle.

1. **La percepción de la precisión de los datos aumenta:** Solo porque podamos introducir las estimaciones de un desarrollador en una herramienta de planificación cara, no alteramos el hecho de que puede ser una mala estimación o, más probablemente, una estimación buena hoy pero que cambiará tal y como el proyecto progrese y más información se obtenga. Las herramientas de planificación pueden crear sofisticados modelos de la característica que implica más credibilidad de la que su base de datos ofrece.

2. **Se crean barreras para la interacción con los stakeholders del proyecto:** Una vez que las tareas y estimaciones se codifican dentro del diagrama de Gantt, el número de stakeholders del proyecto que pueden fácilmente mejorar o actualizar el plan se reduce drásticamente. Con una planificación sofisticada, normalmente es el responsable del proyecto el responsable de actualizar el plan. El responsable de proyecto pregunta regularmente al equipo como está progresando el proyecto y actualiza la duración de las tareas basándose en lo que realmente ha ocurrido en el proyecto y el tiempo estimado para finalizar, pero ¿con qué frecuencia los miembros del equipo tienen que reprogramar las tareas e insertar nuevos grupos de tareas?

Esta observación se refleja en el libro de Donald Reinertsen, Managing the Design Factory, en el cual advierte de los modelos sofisticados:

Hay una sólida razón práctica para no utilizar un análisis más sofisticado: no todo el mundo lo entiende. Cuanto más complicado hagamos un modelo, persiguiendo un poco más de precisión aquí o allí, más creamos un formidable laberinto de cálculos que puede no ser creído por el equipo. Es mucho mejor elegir una técnica de modelado más sencilla y que el 100% del equipo entienda el modelo que correr el riesgo de confundir a la mitad del equipo con un modelo elaborado pero que es un 5% más preciso.

En contraste, las herramientas de planificación y seguimiento ágil emplean un enfoque poco tecnológico y muy accesible. Como implica el nombre, estas herramientas son simples, como por ejemplo tarjetas o diagramas, y fácilmente manipulables por todo el equipo de stakeholders, haciendo cosas como mover las tarjetas, reordenando listas, etc. Adoptando estas técnicas deliberadamente primitivas, evitamos una percepción de la precisión derivada de la herramienta y permitimos a más gente actualizar los planes según la realidad del proyecto.

Entonces, ¿Qué debe hacer un Responsable de Proyectos inteligente?

Si todavía no lo está haciendo, adoptar estos enfoques poco tecnológicos y muy accesibles. Animar al equipo a utilizar un sistema visual para planificar las iteraciones y controlar el estatus global del proyecto. Por ejemplo, puede dedicar un muro en la habitación del equipo para colocar un diagrama bien visible (como el nombre implica, son grandes diagramas de información que la gente puede ver y entender fácilmente). Llevar al cliente frente al diagrama y mostrarle el trabajo que está realizado y el que está planificado realizar en las próximas iteraciones. Cuando se solicitan los cambios, debe trabajar con el equipo y el cliente para priorizar visualmente el nuevo trabajo contra el telón de fondo de las características existentes.

2.2.3.2. Límites de WIP

En trabajo en progreso (WIP), también conocido como trabajo en proceso, es el término que se le da al trabajo que ha sido empezado pero que aún no ha sido completado. Tener niveles excesivos de WIP se asocia con un número de problemas, incluyendo:

- El WIP consume el capital de la inversión y no entrega la rentabilidad de la inversión hasta que se convierte en el producto aceptado. Representa un gasto de dinero sin rentabilidad, lo cual es algo que queremos limitar.
- El WIP esconde los cuellos de botella en los procesos que retardan el flujo de trabajo global y enmascaran los problemas de eficiencia.
- El WIP representa riesgo en forma de tener que volver a realizar el trabajo, ya que todavía puede haber cambios en los elementos hasta que esos elementos se hayan aceptado. Si hay un largo inventario de WIP, a la vez puede haber mucho trabajo rehecho si se requiere un cambio.

A causa de estos problemas, los enfoques ágiles, generalmente animan a limitar el WIP. Una manera común de aplicar límites al WIP en los proyectos ágiles, es utilizar tablas Kanban que restringen la cantidad de trabajo en el sistema y ayudan a asegurar que los límites de WIP no se exceden. Las tablas pueden indicar los límites de WIP mostrando un número preseleccionado de tareas que pueden ser trabajadas en un tiempo dado.

Otra opción es restringir la cantidad de espacio designado en el cual las tarjetas de tareas pueden ser colocadas en la tabla, de manera que solo un número seleccionado de tarjetas puedan caber en el espacio. Los equipos ágiles utilizan herramientas como el las tablas Kanban con límites de WIP para ayudar a identificar y eliminar los cuellos de botella para poder mantener el proceso funcionando de manera eficiente con los niveles óptimos de WIP. Como resultado, estas herramientas ayudan a reducir el riesgo de capital inmovilizado, rehacer el trabajo, y pérdidas en el proyecto.

Sin límites en el WIP, un equipo de trabajo puede estar tentado a llevar a cabo demasiadas partes diferentes del trabajo a la vez. Esto puede hacerse con las mejores intenciones, como por ejemplo para utilizar a todo el que esté disponible y mantener a todo el mundo ocupado en el proyecto. Pero el problema es, si tenemos un cuello de botella en el proceso de los requisitos de la base de datos o designando las interfaces de usuario, por ejemplo, entonces las tareas pueden quedar paradas y el trabajo se acumulará en el sistema. Cuando esto ocurre, es difícil identificar donde se encuentra el cuello de botella, ya que todo el mundo parece ocupado.

La figura 17 ilustra una tabla Kanban sin límites para el WIP. Podemos ver que hay una gran cantidad de WIP, lo que significa que el equipo está ocupado. Pero no podemos ver que tareas están paradas y en qué lugar están los cuellos de botella que están contribuyendo a reducir el flujo de trabajo.

Figura 17: tabla Kanban. (Fuente: Elaboración propia)

Limitar el WIP es como drenar el agua de un rio; nos muestra donde están todas las rocas y obstáculos. No queremos ejecutar el sistema sin WIP demasiado tiempo – esto reduciría la salida ya que todo el mundo menos los implicados en el cuello de botella estarían esperando por trabajo. No obstante, tener un límite de WIP muy bajo por un corto periodo de tiempo, es un método efectivo para identificar los cuellos de botella.

La figura 18 muestra una tabla Kanban con un límite de WIP seleccionado bajo. Únicamente hay un elemento en progreso, lo que significa que algunas personas están paradas y que hay un flujo de trabajo lento. Este límite bajo hace muy fácil identificar que elemento está parando el flujo de trabajo, no obstante.

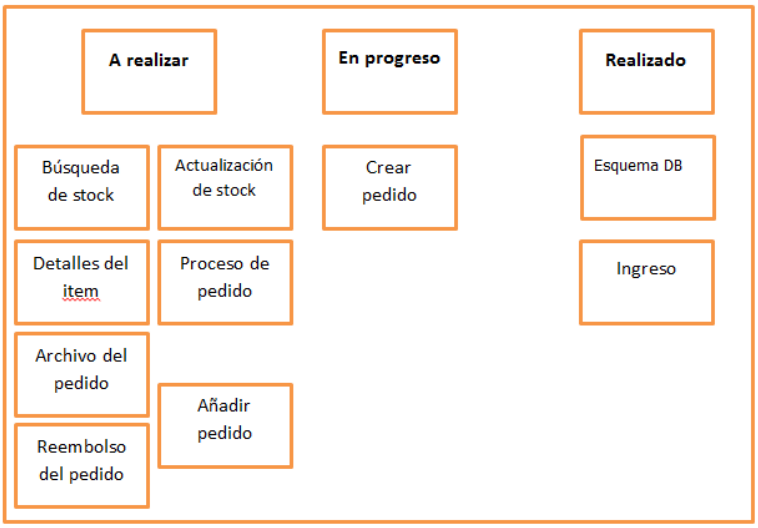


Figura 18: tabla Kanban con un límite de WIP seleccionado bajo. (Fuente: Elaboración propia)

Una vez que el cuello de botella ha sido identificado, podemos elevar la restricción y eliminar el cuello de botella haciéndolo más eficiente. Por ejemplo, podemos tener a otros miembros del equipo haciendo un pre-procesado para esta actividad, preguntar a los consumidores si están dispuestos a recibir un producto menos pulido, o incrementar los recursos para la actividad restringida.

Con la primera restricción eliminada, podemos ejecutar el proceso de nuevo y comprobar si hay otro elemento que esté causando una reducción del flujo de trabajo. Repetiremos el proceso hasta que haya personas temporalmente paradas pero las tareas están siendo llevadas a cabo a tiempo y la salida sea rápida.

La figura 19 muestra una tabla Kanban con un límite apropiado de WIP para el proyecto. Vemos que hay suficiente trabajo en marcha. Hay personas temporalmente paradas, o no tienen ocupación total, pero los cuellos de botella están claros y hay un flujo de trabajo rápido.

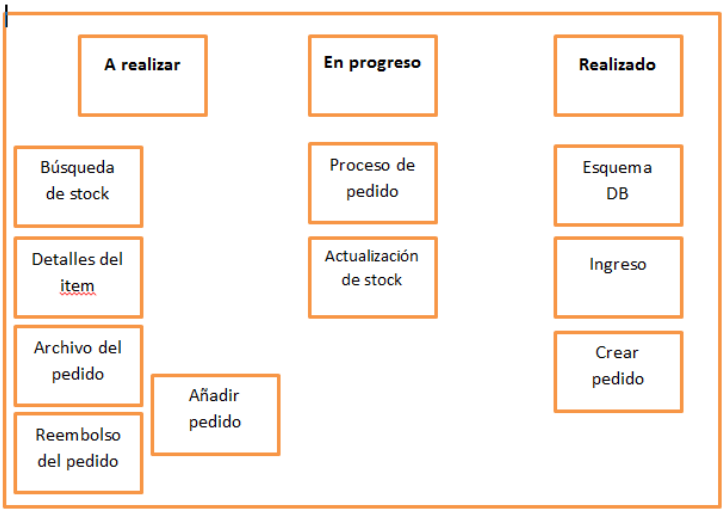


Figura 19: tabla Kanban con un límite apropiado de WIP. (Fuente: Elaboración propia)

El objetivo de limitar el WIP es optimizar la salida de trabajo, no optimizar la utilización de recursos. Esto a menudo va contra lo que nos marca la intuición en primer lugar. Tendemos a pensar que los miembros de un equipo deben estar continuamente trabajando todo el tiempo y todo lo que no sea eso es pereza o ineficiencia. Consideremos una autopista. ¿Cuándo va más fluida – cuando está completamente utilizada en hora punta, o durante el resto de horas cuando tiene cierta holgura?

Limitar el WIP ayuda a identificar los cuellos de botella y maximizar la salida de un proyecto, justamente igual que limitar el número de coches en una carretera ayuda a que el tráfico fluya más rápido. En los proyectos de software, el límite de WIP equivale al número de características que están siendo trabajadas pero que aún no han sido aceptadas por el negocio.

La ley de Little tiene que ver con el WIP. Esta ley establece que el ciclo de tiempo (cuanto tiempo tenemos que esperar para obtener beneficios) es proporcional al tamaño de la cola (cuanto WIP tenemos). En otras palabras, podemos predecir el tiempo que tardaremos en acabar basándonos en el tamaño de la cola de WIP.

2.2.3.4. Entrega incremental

La entrega incremental es otra manera de optimizar la entrega de valor en un proyecto. Con la entrega incremental, el equipo despliega regularmente incrementos de trabajo del producto en el transcurso del proyecto. En el caso de proyectos de desarrollo de software, el software operativo se despliega habitualmente a un ambiente de test para su evaluación, pero si tiene sentido para el negocio, el equipo puede entregar funcionalidad a la producción en incrementos. Si podemos entregar la versión más simple del producto o servicio mientras trabajamos en los elementos más complejos, tenemos una oportunidad de darnos cuenta de los beneficios del producto y obtener una rentabilidad de la inversión anticipada.

Incluso si estamos entregando incrementalmente software en ambientes de test, en lugar de en producción, este enfoque nos puede aún ayudar con la entrega global de valor. Los problemas que se encuentran en el software entregado para su evaluación en el ambiente de test (punto 1 en la figura 20) son mucho más baratos de solucionar que los encontrados durante la producción (punto 2). La entrega incremental reduce la cantidad de trabajo re-hecho encontrando los problemas antes y contribuyendo a la entrega de valor en el proyecto.

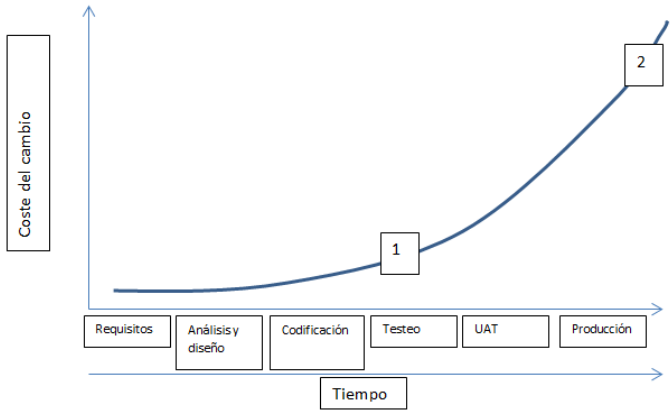


Figura 20: Entrega incremental. (Fuente: Elaboración propia)

2.2.4. Confirmación de valor

Una cosa es pensar que estamos desarrollando grandes productos y servicios y otra es que tus sponsors y clientes lo confirmen. Los métodos ágiles se utilizan a menudo en proyectos intangibles (p.e. diseños, software, etc.). La naturaleza intangible de estos proyectos significa que es mucho más importante validar que lo que estamos haciendo está, de hecho, en el camino correcto y es visto por el negocio como algo altamente valioso.

Ya que una diferencia entre las características que nos piden y las que entregamos nos puede llevar a tener que rehacer el trabajo, es importante descubrir estas diferencias pronto. Por lo tanto, la confirmación del valor – cuando significa que el equipo verifica que está construyendo lo correcto que lo que construye funciona como se desea en lugar que como se había dicho en un principio que tenía que funcionar, lo cual puede diferir – es una práctica clave cuando se utilizan proyectos ágiles.

2.2.4.1. Priorización basada en el valor para el cliente

Vimos la priorización basada en el valor para el cliente como un parte de la planificación anticipada del valor. Esta herramienta y técnica puede también utilizarse para confirmar el valor. Trabajando con el cliente para priorizar los elementos restantes a ser realizados ayuda a asegurar que el trabajo está progresando hacia el objetivo del proyecto, el cual puede ser cambiante. Juntos, el equipo y el cliente, responden a las preguntas de, “¿Han cambiado las cosas?” y “¿Todavía nos queremos mover hacia el siguiente elemento X?”. Estas sesiones de re-priorización, las cuales se realizan normalmente al final de cada iteración, sirven como un importante punto de control para los proyectos ágiles. Las prioridades nuevas o en evolución se capturan entonces en la cartera de la historia del usuario, la cual es revisada de nuevo en las siguientes sesiones.

A menudo sorprende lo que para el cliente ocupa un lugar destacado cuando empezamos a dar prioridad a las características e historias en las que trabajar. Al discutir estas primeras opciones aparentemente improbables, la discusión suele sacar las preocupaciones políticas y los riesgos residuales a la superficie. Por ejemplo, "Estamos haciendo las ratios Premium primero, porque son un objetivo para nuestros hitos del primer Trimestre" o "Tenemos que entregar el salpicadero CIO para asegurar la financiación del proyecto para el próximo año."

Aunque podríamos sentirnos tentados a discutir que esto no es un sistema real de beneficios para el negocio, es a menudo inteligente escuchar estas motivaciones, ya que los representantes del negocio o clientes son nuestros aliados en el proyecto y quienes declaran donde está el éxito del proyecto.

Al pedir a los representantes de las empresas o los clientes cuáles son sus características de máxima prioridad, aprendemos acerca de sus motivaciones, riesgos y criterios de aceptación. Los proyectos que no participan de la priorización basada en el valor para el cliente probablemente se perderán en la identificación de factores críticos de éxito.

2.2.4.2. Prototipos, simulaciones, demostraciones

Las demostraciones de la funcionalidad son fundamentales para confirmar el éxito en los proyectos de software. El software es intangible y difícil de referenciar. Además, las compañías raramente construyen el mismo sistema dos veces, lo cual significa que los proyectos desarrollan productos que, en su mayor parte, son nuevos para los usuarios. Por lo tanto, los usuarios necesitan la oportunidad de mirar a algo, y probarlo para ser capaces de confirmar que la funcionalidad es la adecuada. El término IKIWISI (I'll Know It When I See It "Lo conoceré cuando lo vea") se utiliza a menudo en el desarrollo de software, ya que los requisitos reales aparecerán una vez el producto sea mostrado y utilizado.

Además, para ayudar a clarificar los requisitos, las demostraciones pueden descubrir la necesidad de nuevas características. A menudo, solo cuando los representantes del negocio ven la funcionalidad pueden pensar en nuevos elementos que sean necesarios. Por ejemplo, "La pantalla de entrada de comandos está muy bien, pero después de probarla pensamos que necesita un duplicado de la función de comandos".

Cuando los equipos demuestran la funcionalidad, ocurren dos cosas. Primero, aprenden sobre las diferencias entre lo que se pidió y lo que se interpretó y se construyó. Segundo, aprenden sobre nueva funcionalidad o ajustes en la existente (IKIWISI).

Por lo tanto, los requisitos evolucionan con los prototipos, simulaciones y demostraciones. Dando una oportunidad a la gente de evaluar y utilizar algo se ayuda a descubrir los requisitos reales del negocio. Nuestra solución converge entonces, hacia los requisitos emergentes y la necesidad de evaluación se hace más pequeña. Este concepto se ilustra en la figura 21, en la cual X1, X2 y X3 son los requisitos emergentes y O1, O2 y O3 representan que fue interpretado e implementado. Como podemos ver, la solución coge forma con los requisitos y lo que se ha creado converge sobre la tercera iteración.

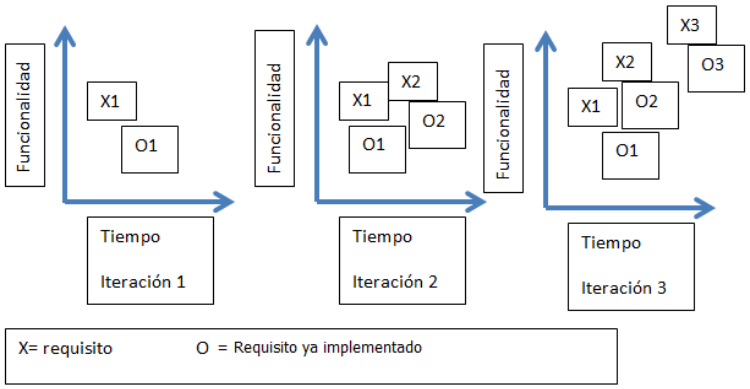


Figura 21: Convergencia de la solución. (Fuente: Elaboración propia)

2.2.5. Siguiendo y reportando el valor

Hemos hablado sobre las prácticas de evaluación, planificación, entrega y confirmación del valor. Vamos a ver ahora el seguimiento y reporte del valor. En un ambiente ágil, es importante monitorizar el ratio al cual las características y el valor se están entregando, para asegurarse de que estamos en el camino de completar el proyecto como se había acordado. Una vez que determinamos el estatus del proyecto, necesitamos comunicar ese estatus a los stakeholders. Hay diversas herramientas y técnicas que podemos utilizar para seguir y reportar el valor, incluyendo el valor ganado, diagramas acumulativos de flujo, gráficos Risk Burndown, y tablas de tareas y Kanban.

2.2.5.1. Valor ganado ágil

Cuando se trata de seguir y reportar el progreso de nuestros proyectos, las curvas S son grandes herramientas para dar seguimiento al gasto del proyecto. Son simples de interpretar, y rápidamente transmiten si el proyecto está sobre o bajo presupuesto.
Esta curva S, no obstante, no nos revela ninguna información sobre la planificación. Podemos estar afinando el gasto, pero estar detrás de la perspectiva de programación. Para reportar el estatus de la planificación, normalmente se utilizan diagramas de seguimiento de Gantt.

Pero tenemos una limitación similar con los diagramas de Gantt. Les falta el componente de gasto, al igual que a las curvas S les falta el componente de programación. Muy pronto muchos proyectos se adelantan o retrasan al presupuesto o a la planificación, por lo que intentar medir la salud global del proyecto puede ser difícil.

Además, la naturaleza iterativa de los proyectos ágiles significa que los diagramas de Gantt pueden llegar a ser profundamente confusos.

Estas son las razones por las cuales se crearon el análisis del valor ganado (EVA) y la administración del valor ganado (EVM). El valor ganado combina los datos de gasto y planificación para producir un conjunto completo de medidas e indicadores.

¿Se puede utilizar el valor ganado en los proyectos ágiles? La respuesta es sí. Pero mientras que las matemáticas funcionan de la misma manera que los proyectos no ágiles, necesitamos ser cuidadosos con lo que estamos midiendo. El valor ganado, compara el rendimiento actual del proyecto con el rendimiento planificado en un punto particular del tiempo. Esto significa que si nuestro plan inicial está equivocado, podríamos estar siguiendo nuestro proyecto en el lugar equivocado. La calidad del plan de línea base es un factor crítico de éxito en el uso del valor ganado.

En los proyectos ágiles, sabemos que los planes iniciales muy probablemente necesitaran cambiar, por lo que la base de para EVM efectivo se erosiona rápidamente con la evolución de los planes.

Otro punto de atención en cuanto al valor ganado es que este verdaderamente no indica si el proyecto es exitoso o no. Podemos estar dentro del tiempo y del presupuesto, pero construyendo un producto malo y de baja calidad que el negocio no quiere ni necesita.

¿Qué tiene de bueno el valor ganado?

Acabamos de discutir algunos aspectos del valor ganado sobre los que tenemos que tener cuidado en los proyectos ágiles. ¿Entonces, porque lo utilizamos? Uno de los beneficios clave del valor ganado es que es un indicador principal. Mirar hacia atrás no nos sirve de mucho. EVM mira hacia adelante, intentando predecir las fechas de finalización y los costes finales. Después de todo, los indicadores principales imperfectos son generalmente más valiosos que indicadores secundarios perfectos, ya que los indicadores principales nos dan la oportunidad de replantear y cambiar nuestro enfoque.

Otro beneficio del valor ganado es que es visual. A menudo se olvidan los diagramas EVM y uno se centra en los números únicamente, pero en el corazón de esta técnica hay muchos gráficos útiles. La representación visual de la información activa el lado correcto de nuestros cerebros y nos ayuda a absorber los datos mientras trabajamos en entender, interpretar y planificar las respuestas adecuadas. Las representaciones visuales son también mejores cuando se discute la información y se trabaja en colaboración, ya que la gente puede señalar, anotar y extrapolar más fácilmente con imágenes que con palabras y números.

Cálculo del valor ágil ganado.

Entonces, ¿cómo podemos mantener los componentes predictivos y visuales del EVM y reducir algunas de sus desventajas? La curva S, es un buen comienzo para esto.

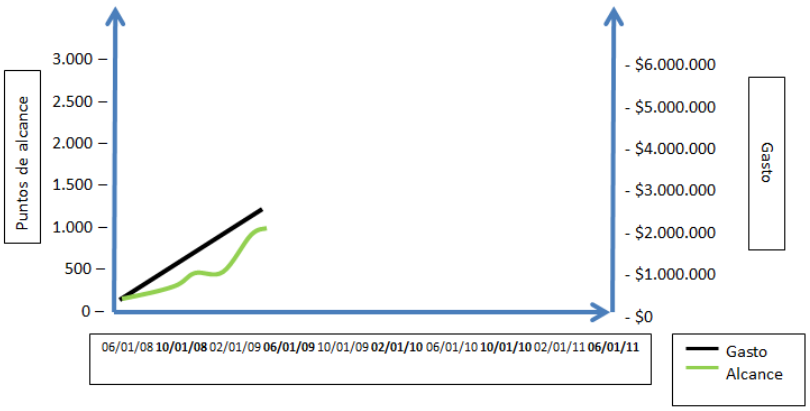


Figura 22: curva S. (Fuente: Elaboración propia)

En la figura 22, vemos la línea de gastos familiar, en negro, la cual se traza contra la escala de dólares en el eje de la derecha. También tenemos una línea basada en las características (el alcance realizado hasta la fecha en el proyecto) mostrado en verde, la cual se traza contra la escale de puntos a la izquierda.

El gradiente de la línea verde indica la velocidad del proyecto. Cuando se eleva bruscamente, desarrollamos muchos puntos en poco tiempo. Cuando es plana, el progreso es lento.

Añadir un trasfondo al gráfico puede ser una manera útil de mostrar las áreas funcionales del proyecto. En la figura 23, podemos ver que los componentes de “configuración” y “stock” del sistema han sido construidos y que estamos actualmente en la parte de “Ventas” del ámbito del proyecto, justo por encima de los 1000 puntos del valor de la funcionalidad completa. También, en Octubre de 2008, el paso en las líneas del trasfondo, muestra donde se incrementó el ámbito de la parte de “Ventas”, desplazando las restantes áreas funcionales hacia arriba para acomodar el nuevo trabajo.

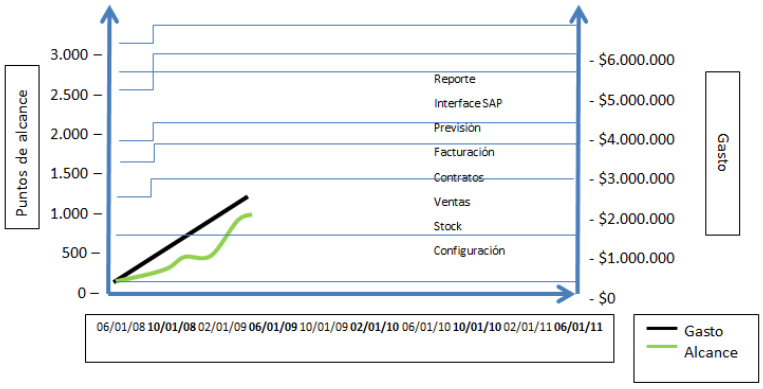


Figura 23: Curva S. (Fuente: Elaboración propia)

Por lo tanto, tenemos una gráfica que muestra el alcance, programación y el rendimiento de los costes actualizados, pero aun nos faltan las previsiones que nos den una idea de si estamos por encima o por debajo de las perspectivas de presupuesto y calendario. Aquí es donde entran la predicción de valores, como vemos en la figura 24:

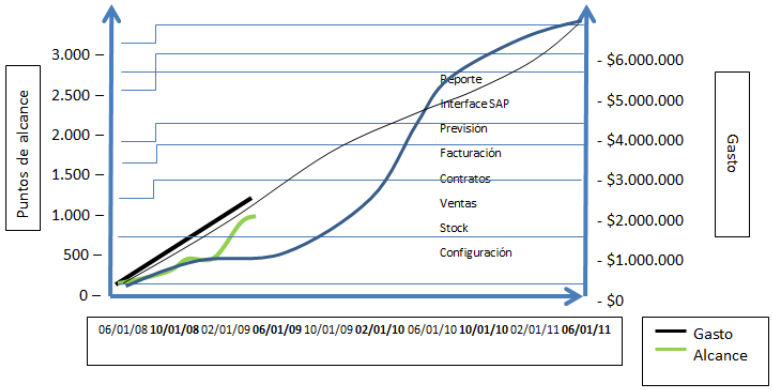


Figura 24: curva S con previsiones. (Fuente: Elaboración propia)

Ahora podemos ver como la progresión real del proyecto se compara con nuestras previsiones. La figura anterior muestra que, en este punto del proyecto, hemos gastado de más y que vamos un poco por encima en el ámbito de aplicación.

Estos gráficos pueden servir como un sustituto para la gestión del valor ganado, aportando los mismos indicadores e índices pero de forma visual.

Los indicadores tradiciones EVM como el Índice de Rendimiento de la Programación (SPI) y el Índice de Rendimiento de los Gastos (CPI) pueden trasladarse fácilmente a términos ágiles. Por ejemplo, habíamos planificado completar 30 puntos de historial en esta iteración, pero solo hemos completado 25. Para obtener el SPI, podemos dividir 25 por 30 obteniendo un valor de 0.83 (significando que estamos trabajando al 83% de lo planificado). Igualmente, CPI es el valor ganado (EV, o el valor de la características completadas hasta la fecha) dividido por los costes actuales (AC) hasta la fecha. Por lo tanto, en el ejemplo del gráfico anterior, CPI = $2,200,000 / $2,800,000 = 0.79. Esto significa que estamos obteniendo solo 79 centavos de dólar comparado con lo que habíamos predicho.

2.2.5.2. Diagramas de Flujo Acumulado (CFDs)

Los diagramas de flujo acumulado, son herramientas valiosas para realizar seguimientos y previsiones en proyectos ágiles. Los CFDs nos pueden ayudar a obtener una visión de los problemas ocasionados dentro del proyecto, de los ciclos de tiempo, y fechas de finalización probables.

La figura 25 es un ejemplo de CDF.

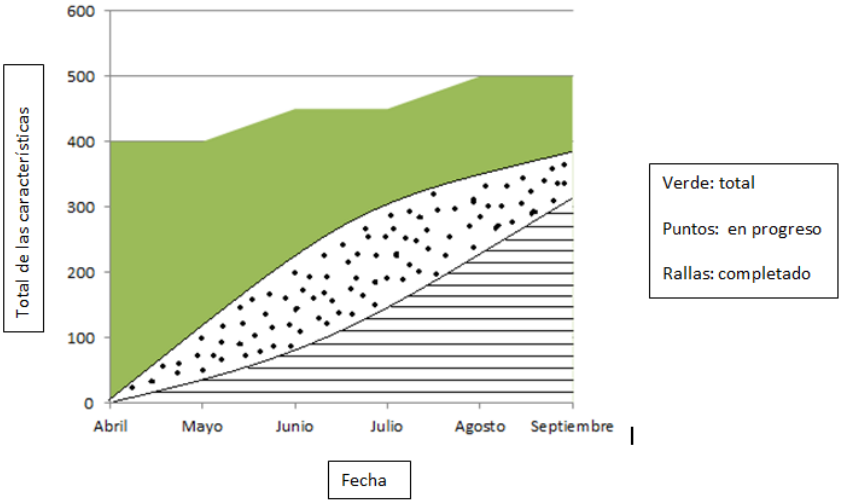


Figura 25: diagrama de flujo acumulado. (Fuente: Elaboración propia)

La anterior figura nos muestra las características completadas contra las características aún pendientes para un proyecto ficticio que continúa en proceso. El área verde representa todas las características planeadas para ser desarrolladas. Este número aumentó de 400 a 450 en Junio y a 500 en Agosto ya que se añadieron características adicionales al proyecto. El área punteada muestra el trabajo en progreso, y la sección rallada muestra el número total de características completadas.

La Ley de Little.

La ley de Little es un concepto introducido por Donals Reinertsen que puede ser utilizado para analizar las colas de los diagramas CFDs. El concepto se ilustra en la figura 26.

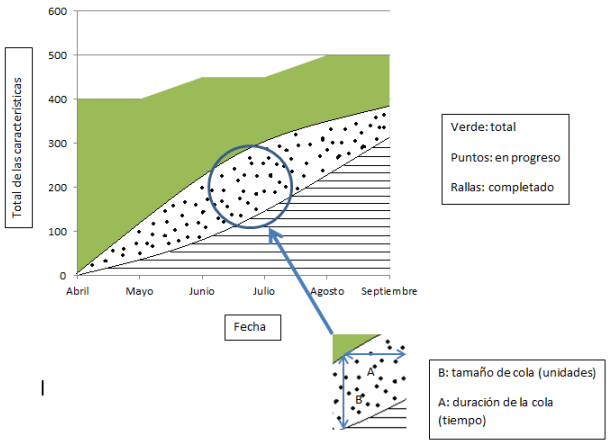


Figura 26: Ley de Little. (Fuente: Elaboración propia)

El área punteada representando el trabajo en progreso (WIP) es nuestra cola; es el trabajo que hemos empezado pero aún no hemos completado. Podemos ver cuantos elementos están en la cola mirando a la distancia vertical B, y podemos determinar el ciclo de tiempo, o cuanto probablemente tardaremos en completar los elementos en la cola, examinando la distancia horizontal A.

Los ciclos de tiempo son importantes indicadores para los sistemas de producción lean. Estos datos nos ayudan a predecir cuándo estará hecho todo el trabajo que actualmente tenemos en progreso. Debemos aspirar a mantener el WIP y los tiempos de ciclo lo más bajos posibles, ya que representan los costos de inversión a fondo perdido que aún no han resultado en beneficios para el negocio. Cuanto más alto sea el WIP y los ciclos de trabajo en el proyecto, más alta será la cantidad de pérdidas potenciales que tendremos en el sistema, si encontramos un problema.

Buscando cuellos de botella y la Teoría de las limitaciones.

Hemos visto anteriormente los conceptos de cuello de botella, WIP, teoría de las limitaciones en la entrega de valor, pero ahora los veremos de nuevo desde la perspectiva del seguimiento y reporte de valor en los proyectos ágiles.

Eli Goldratt introdujo la Teoría de las Limitaciones (TOC) como una herramienta para optimizar los sistemas de producción. Observó que “los cambios en muchas de la variables en una organización normalmente solo tienen un pequeño impacto en el rendimiento global. Hay unas cuantas variables (tal vez solo una) para las cuales un cambio significativo en el rendimiento local puede producir un cambio significativo en el rendimiento global”. Por lo tanto, para conseguir los mejores beneficios debemos encontrar esas variables (o cuellos de botella en el sistema) y centrarnos en mejorar sus problemas.

La pregunta 3 de la reunión diaria, que pregunta por cualquier impedimento para progresar, es una adopción del método TOC. Con esta pregunta, estamos buscando barreras en el proyecto y eliminándolas.

Las limitaciones se pueden manifestar también como restricciones en la capacidad de salida. Por ejemplo, el grupo que se encarga de la base de datos, puede no ser capaz de tratar con los cambios que les llegan del equipo de desarrolladores, o tal vez, el proxy del cliente no puede actualizarse con preguntas sobre las reglas de validación para una nueva pantalla que el equipo está desarrollando. En ambos casos. El grupo que se encarga de la base de datos y el proxy del cliente son limitaciones, pero para el responsable del proyecto, estos cuellos de botella no siempre son fáciles de detectar.

Usualmente se presiente dónde pueden estar estos cuellos de botella, pero a no ser que se esté encima de la gente, controlando cuantas características pueden procesar al día, ¿cómo podemos medir estos cuellos de botella así como la actividad de salida? Aquí es donde entra en juego el CFDs.

Podemos usar el CFD para encontrar los cuellos de botella si, en lugar de agrupar todo el trabajo en proceso en una simple medida, lo separamos por actividades y graficamos el flujo de este trabajo. La figura 27 muestra un ejemplo de una CFD detallado.

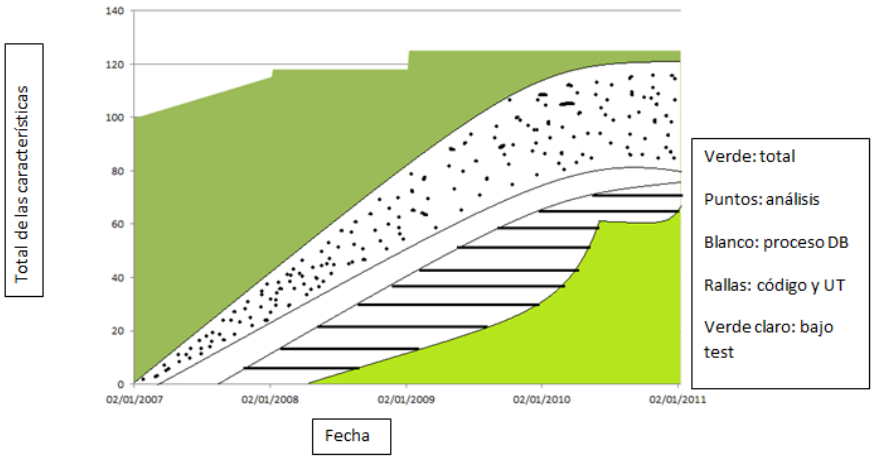


Figura 27: separado por actividades. (Fuente: Elaboración propia)

En la figura 27, el trabajo en progreso se ha separado por actividades, y las actividades se apilan secuencialmente. Análisis, procedimientos de la base de datos, codificación de la unidad de testeo y el trabajo de testeo del usuario han sido realizados en el proyecto por diferentes grupos. Cuando examinamos el CFDs buscando cuellos de botella, buscamos áreas que se ensanchan antes de que la actividad final esté hecha. En este ejemplo, el testeo por el usuario es la actividad representada por la franja fina. El ensanchamiento de un área indica la finalización creciente del trabajo. Para explicar esta idea utilizando conceptos matemáticos, un área ensanchada se crea cuando una línea es seguida por otra línea de gradiente menos profundo. Ya que la línea de gradiente indica el ratio de progreso para una actividad (características sobre tiempo), un área ensanchada se crea sobre una actividad que está progresando a un ratio más lento.

En la figura 28, podemos ver que “Análisis” es una actividad de área ensanchada, mientras que “Procedimientos de la base de datos” es una actividad con cuello de botella.

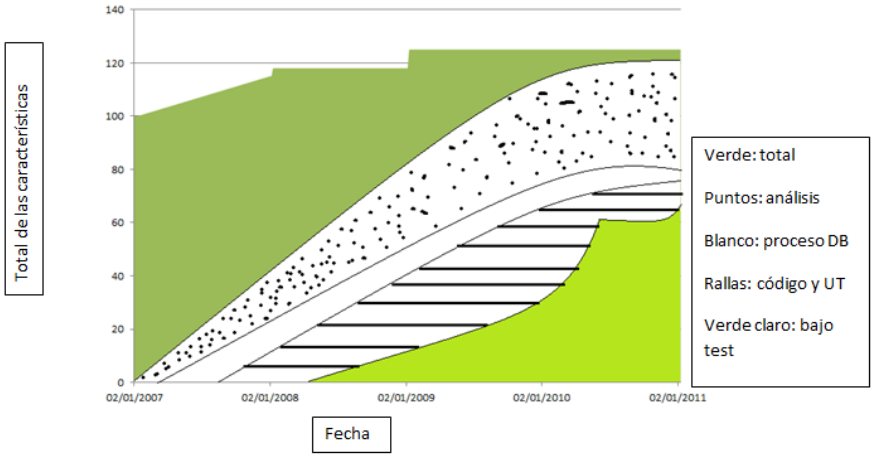


Figura 28: separado por actividades. (Fuente: Elaboración propia)

Independientemente de si la explicación matemática tiene sentido, todo lo que hay que recordar para examen es que la actividad con cuello de botella es la actividad que se encuentra por debajo del área ensanchada. El área ensanchada es la alimentación que alimenta, no la actividad problemática. Por lo tanto, en el ejemplo mostrado en la figura anterior, “Análisis” va bien. No obstante, la actividad que se encuentra por debajo del área ensanchada – la creación de los procedimientos de la base de datos almacenados (DB Procs) – muestra un ratio de progreso más lento. El CFD nos da una advertencia de que la creación de los procedimientos de la base de datos almacenados es un cuello de botella en el sistema.

Por lo tanto, no necesitamos micro gestionar la salida del equipo para determinar donde se encuentra el cuello de botella. En lugar de eso, utilizando el CFDs para seguir el progreso, podemos identificar los cuellos de botella discretamente. Entonces, una vez que sabemos dónde está el problema, podemos comenzar a abordar el problema aplicando los “Five Focussing Steps” de la teoría de las limitaciones de Goldratt, empezando por el paso 2:

1. Identificar la limitación
2. Aprovechar la limitación
3. Subordinar todos los otros procesos a explotar la limitación.
4. Si después de realizar los pasos 2 y 3, se necesita más capacidad para satisfacer la demanda, elevar la limitación.
5. Si la limitación no se ha eliminado, volver al paso 1, pero no dejes que la complacencia se convierta en la limitación del sistema.

El uso de los métodos ágiles inherentemente reduce la posibilidad de cuellos de botella basados en la actividad, ya que estos métodos promueven los equipos multidisciplinares o con capacidad múltiples. Evitando la especialización en una tarea determinada, las personas son capaces de desplazarse entre los diferentes roles de manera más afectiva y compartir la carga de trabajo. No obstante, mientras que el objetivo es tener equipos completamente multidisciplinares, en la práctica cierta especialización y gestión del flujo de trabajo son la norma en los proyectos.

2.2.5.3. Gráficos Risk Burn Down

Como hemos visto anteriormente, el riesgo y el valor están íntimamente relacionados. Por lo tanto, la gestión del riesgo es una actividad importante en los proyectos ágiles. La gestión del riesgo debe servir como motor para la planificación del trabajo, colocando las actividades de alto riesgo en las primeras iteraciones del proyecto e incorporando acciones para mitigar el riesgo en la cartera. Tal y como el proyecto progresa, necesitamos continuar gestionando este riesgo y monitorizando la efectividad de nuestros esfuerzos.

Necesitamos atacar de manera efectiva las amenazas antes de que se transformen en problemas en el proyecto. Desafortunadamente, demasiado a menudo, simplemente nos limitamos a realizar los pasos de análisis y gestión de riesgos junto al resto de tareas normales del proyecto, en lugar de tomar la gestión del riesgo como motor de la planificación del trabajo. Muchos proyectos crean planes para la gestión del riesgo y listas de riesgo, pero luego no dejan que esas conclusiones influyan en la selección de tareas y la planificación. Como resultado, los pasos de respuesta al riesgo tampoco influyen en el plan del proyecto.

Con un enfoque ágil, hay muchas oportunidades de atacar de manera activa a los riesgos en el proyecto antes de que se conviertan en problemas. El desarrollo iterativo permite abordar anticipadamente el trabajo de alto riesgo en el ciclo de vida del proyecto. Las características que llevan altos niveles de riesgo asociado pueden llevarse a cabo en iteraciones tempranas para probar el enfoque tecnológico y eliminar las dudas.

Debemos sopesar cuidadosamente los objetivos de entregar valor para el negocio y reducir el riesgo, cuando seleccionamos las características.

Los riesgos se tratan generalmente vía dos medidas: - probabilidad de riesgo (una medida de como de probable es que un riesgo ocurra) y el impacto del riesgo (una medida de las consecuencias para el proyecto si el riesgo ocurriera). Cuando la probabilidad se indica como porcentaje y el impacto se define en términos monetarios, estas medidas pueden utilizarse para calcular el valor monetario esperado del riesgo (EMV), como hemos visto antes. Por ejemplo, si se estima que un riesgo tiene un 25% de probabilidades de ocurrir y se estima su impacto como $8,000, el EMV = 0.25 x $8,000 = $2,000.

Cuando clasificamos la probabilidad y el impacto en una escala numérica, podemos utilizar estas medidas para calcular la severidad del riesgo, como se indica en la siguiente formula:

Severidad del riesgo = Probabilidad del riesgo x Impacto del Riesgo

Calcular la severidad del riesgo nos permite clasificar los riesgos y determinar la prioridad de respuesta al riesgo. Por ejemplo, si puntuamos bajo como (1), medio como (2) y alto como (3) tanto para la probabilidad como para el impacto del riesgo, podemos ver que los riesgos de alta probabilidad y alto impacto obtienen una puntuación de severidad del riesgo de 3 x 3 =9. Un riesgo de alta probabilidad pero bajo riesgo obtendrá una puntuación de severidad del riesgo de solo 3 x 1 = 3.

Es posible llevar el análisis de los riesgos más allá utilizando técnicas como el valor monetario esperado, pero para el propósito de ilustrar los perfiles y tendencias del riesgo, los valores abstractos de la puntuación de severidad del riesgo del 1 al 9, es todo lo que necesitamos.

Para cualquier proyecto, debemos involucrar al equipo de desarrollo, a los sponsors, clientes y otros stakeholders, en el proceso de identificación de riesgos. Sus ideas, junto con revisiones de las notas de proyectos anteriores, registros de riesgos y perfiles de riesgo de la industria, deben ser utilizadas para identificar los riesgos conocidos probables para el proyecto. Una vez que tenemos la lista de riesgos, podemos llevar a cabo el análisis de riesgos y asignar las puntuaciones de probabilidad y riesgo para cada riesgo. Entonces, podemos utilizar estas puntuaciones para calcular la severidad de los riesgos. De forma similar que con la estimación, involucramos a los miembros del equipo en el análisis de riesgo, porque están en contacto con los detalles técnicos y porque incluyéndolos generamos una mejor aceptación del plan de gestión de riesgos y el plan de acciones de respuesta. Si no están involucrados, no habrá compromiso.

En la siguiente tabla se muestran algunos riesgos identificados para un proyecto ficticio y sus puntuaciones de probabilidad, impacto y severidad asociadas.

ID	Nombre corto para el riesgo	Impacto	Probabilidad	Severidad
1	Rendimiento del driver JDBC	3	2	6
2	Llamar a los proc. Almacenados de Oracle desde el servicio Web	2	2	4
3	Distribución remota de las app. Hacia las PDAs	3	2	6
4	Estabilidad del almacén de Oracle	2	2	4
5	Estabilidad del sistema Legacy	2	1	2
6	Acceso a la comunidad de usuarios	2	1	2
7	Disponibilidad de la arquitectura	2	2	4
8	Necesidad de actualización del servidor	1	2	2
9	Lanzamiento del navegador Oracle	3	1	3
10	Cambios del PST para British Columbia	0	0	0
	Total			33

Tabla 9: Riesgos identificados para un proyecto ficticio y sus puntuaciones de probabilidad, impacto y severidad asociadas.

Tal y como el proyecto progresa, expandiremos la tabla para grabar como están funcionando nuestros intentos de gestionar el proyecto. En la siguiente tabla se muestra el progreso de los riesgos a través de los primeros 4 meses del proyecto.

		Enero			Febrero			Marzo			Abril		
ID	**Nombre corto para el riesgo**	**Imp**	**Pro**	**Sev**	**Imp**	**Pro**	**Sev**	**Imp**	**Pro**	**Sev**	**Imp**	**Pro**	**Sev**
1	Rendimiento del driver JDBC	3	2	6	3	0	0	3	0	0	3	0	0
2	Llamar a los proc. Almacenados de Oracle desde el servicio Web	2	2	4	2	0	0	2	0	0	2	0	0
3	Distribución remota de las app. Hacia las PDAs	3	2	6	3	1	3	3	0	0	3	0	0
4	Estabilidad del almacén de Oracle	2	2	4	2	3	6	2	2	4	2	0	0
5	Estabilidad del sistema Legacy	2	1	2	2	1	2	2	0	0	2	0	0
6	Acceso a la comunidad de usuarios	2	1	2	2	2	4	2	1	2	2	1	2
7	Disponibilidad de la arquitectura	2	2	4	2	3	6	2	2	4	2	0	0
8	Necesidad de actualización del servidor	1	2	2	1	1	1	1	0	0	1	0	0
9	Lanzamiento del navegador Oracle	3	1	3	3	1	3	3	3	9	3	1	3
10	Cambios del PST para British Columbia	0	0	0	0	0	0	2	2	4	2	1	2
	Total			33			25			23			7

Tabla 10: Tabla expandida

Durante estos cuatro meses, muchos de los riesgos del proyecto fueron mitigados o evitados por completo. Por ejemplo, el primer riesgo, acaba no siendo un problema ya que las características de la base de datos y el rendimiento del testeo fueron deliberadamente incluidos en la primera iteración (mes uno). Como resultado, la probabilidad de que ocurra el riesgo se redujo de 2 a 0 y la severidad del riesgo pasó de 6 a 0.

Este tipo de detalles pueden ser difíciles de determinar únicamente mirando las gráficas. Para hacerlas más claras, podemos convertir la información en un diagrama de "Risk Burn Down", el cual es una excelente herramienta para mostrar la tendencia y el status.

Los diagramas "Risk Burn Down" son esencialmente gráficos de áreas amontonadas de severidad de riesgo acumulada del proyecto. Las puntuaciones de severidad para cada riesgo se grafican una encima de otra para mostrar el perfil de severidad acumulada del proyecto. Cuando los riesgos y su historial de severidad se muestran en este formato, es mucho más fácil interpretar el status del riesgo global del proyecto y su tendencia.

En los gráficos "Risk Burn Down" también es fácil observar los riesgos que empeoran, así como los nuevos riesgos.

Los diagramas "Risk Burn Down" informan rápidamente a los stakeholders si los riesgos se están moviendo en la dirección adecuada (descendiendo) o si están incrementándose. Estos gráficos proveen de una manera de interpretar fácil para comprobar la salud de nuestro proyecto y se pueden generar rápidamente en herramientas como Microsoft Excel. Aunque el riesgo del proyecto no es un indicador principal, como el de las características entregas, es útil para realizar el seguimiento. Necesitamos intentar controlar los riesgos de forma activa, ya que nuestra efectividad en esto, impactará en si conseguimos entregar un proyecto de manera exitosa.

2.2.5.4. Tablas de tareas/Kanban

La última herramienta que veremos son las tablas de tareas o tablas Kanban. Aunque inicialmente fueron pensadas como herramientas de planificación y monitorización también pueden ser una gran manera de seguir y reportar el valor.

Uno de los beneficios de reunirse frente a la tabla y actualizarla diariamente es que el equipo ve un movimiento gradual de los elementos de izquierda a derecha tal y como éstos se mueven de “Para hacer” a “En progreso” y a “Hecho”. Es gratificante para el equipo – y tranquilizador para el sponsor – ver una colección en aumento de trabajos acabados tal y como se acerca la fecha de finalización. También centra la atención de todos en los elementos que quedan, que realmente empiezan a llamar la atención al ser los últimos en moverse.

Herramientas como la tabla de tareas, ayudan a reportar y seguir el valor, y sus resultados se muestran a menudo en diagramas mucho mayores que se enseñan en lugares prominentes.

Responsabilidad y éticas profesionales.

Antes de continuar con el siguiente capítulo, veremos como el Código PMI de conducta ética y profesional, aplica a los conceptos de entrega basada en el valor. Hay cuatro áreas principales en este código – Responsabilidad, Respeto, Justicia y Honestidad – las cuales veremos por turnos, viendo ejemplos de como se relacionan con los tópicos vistos en este capítulo.

RESPONSABILIDAD.

Tomar decisiones basándose en los mejores intereses para la compañía. Este deber casa muy bien con el concepto de entrega y priorización basada en el valor. Establecemos el orden de trabajo del proyecto de acuerdo con el valor del negocio. En otras palabras, nuestros proyectos se rigen por la razón por la cual la compañía está llevando a cabo el trabajo, en lugar de por las preferencias del gestor o el equipo del proyecto.

Este aspecto de responsabilidad profesional también habla del deber del equipo de desarrollo de identificar y ayudar a priorizar los riesgos técnicos que de otra manera no podrían conocerse. No estaremos tomando decisiones basándonos en los mejores intereses para la compañía si dejamos que los problemas técnicos interrumpan nuestra habilidad para entregar valor.

Proteger la información propietaria. Aunque para mucha gente sea de sentido común, debemos pensar sobre las implicaciones de introducir requisitos en una herramienta web de priorización. ¿La información está segura y protegida? ¿Si utilizamos las herramientas, quien obtiene la información que se incluye en ella? ¿Quién tiene derecho a ver esa información? Debemos ser cuidadosos cuando adoptemos nuevas herramientas para no comprometer la información propietaria.

Reportar comportamientos no-éticos y violaciones. Con suerte no veremos comportamientos de este tipo en nuestros proyectos, pero tenemos que estar vigilantes a cosas como un propietario del producto agresivo forzando a los miembros del equipo de desarrollo a construir requisitos no aprobados para la iteración en curso.

RESPETO.

Mantener una actitud de mutua cooperación. Este deber profesional tiene que ver con equilibrar los requisitos del negocio con la necesidad de reducir los riesgos técnicos. Los representantes del negocio y el equipo de desarrollo deben colaborar para seleccionar los elementos para una iteración, equilibrando la entrega de nuevas características para el negocio con las acciones de respuesta a los riesgos.

Negociar de buena fe. Cuando creamos estimaciones y negociamos las prioridades, el presupuesto, y la programación, necesitamos actuar de buena fe y con transparencia para que el proceso funcione. Las estimaciones de relleno y la creación de espacios confortables en los planes de iteración y entrega socavan la estrecha colaboración que los representantes del negocio y el equipo de desarrollo quieren crear. Este tipo de acciones también hacen difícil seguir estimaciones de manera precisa y medir el ratio real de progreso.

Otro aspecto de este deber profesional afecta a nuestras interacciones con los vendedores externos. Cuando solicitamos ofertas para trabajo y negociamos contratos ágiles con vendedores externos, necesitamos estar seguros de que compartimos la información de manera limpia entre vendedores.

No usar tu posición para influenciar a otros. En los proyectos ágiles, el gestor del proyecto no tiene poder de vetar que es lo que entra en una iteración, ni de decidir la prioridad de un elemento o decidir cuanto trabajo puede realizar el equipo dentro de una iteración.

JUSTICIA.

Buscar y revelar conflictos de interés. Este punto es particularmente importante cuando se negocian contratos para proyectos ágiles. Tenemos la responsabilidad de revelar las relaciones con los vendedores y cualquier posible conflicto de intereses.

HONESTIDAD.

Comprender la verdad. En términos de entrega basada en el valor, esto significa que necesitamos comprobar que el negocio o el cliente están obteniendo lo que quería.

Ser sincero en todas las comunicaciones. No debemos participar en o condonar el comportamiento que está pensado para engañar a otros, como hacer engaños o falsas declaraciones, medias verdades, proporcionar información fuera de contexto, u ocultar información. Esto puede ser especialmente relevante cuando trabajamos con los representantes del negocio en las prioridades y estimaciones y también cuando trabajamos con vendedores en contratos y negociaciones. Otro punto a considerar es que cuando negociamos con vendedores no debemos participar en comportamientos deshonestos con la intención de obtener ganancias personales o a expensas de otros.

EJERCICIOS

En el Anexo F.2, se encuentra un examen tipo sobre lo tratado en este punto.

2.3. IMPLICACIÓN DE LOS STAKEHOLDERS

Los Stakeholders, tal y como se define en la guía PMBOK, son cualquier persona o grupo que puede resultar impactado o impactar en el proyecto. Éste es un término bastante amplio que incluye gente en el papel de cliente, sponsor, y /o representantes del negocio, así como el gestor de proyecto, el equipo de desarrollo, cualquier vendedor involucrado en el proyecto, y otras personas en la organización afectada por el trabajo del proyecto o sus resultados, incluyendo a los usuarios finales del proyecto.

Conseguir implicar a los stakeholders – en otras palabras, involucrarlos en el proyecto – es absolutamente esencial para el éxito de un proyecto ágil.

¿Por qué ponemos la atención en los Stakeholders?

Los proyectos son llevados por personas y para personas. Los proyectos de trabajadores con conocimiento a menudo no tienen un producto tangible; por lo tanto, una comunicación efectiva y la involucración de los stakeholders es crítica para asegurar que los miembros del equipo saben que están construyendo y que está demandando el cliente. Por ejemplo, si un equipo está desarrollando un gran sistema de negocio que operará en un ambiente de trabajo con frecuentes cambios, es vital para el equipo hacer participar a los clientes durante todo el proyecto para asegurar que el equipo comprende los detalles correctos para cada componente y que son capaces de incorporar los últimos cambios. Solo con esta implicación de los Stakeholders puede el equipo estar seguro de que están entregando un producto o servicio óptimos.

En los proyectos de ingeniería físicos, el trabajo es visible, tangible y familiar; por lo tanto la distancia entre lo que te piden y lo que produces es pequeña y se puede acortar rápidamente. En contraste, en los proyectos con trabajadores del conocimiento, el trabajo es a menudo invisible, intangible y nuevo; esto lleva a una mayor distancia en la evaluación, y los malentendidos son más comunes. Los métodos ágiles apuntan a solucionar este problema con demos, sesiones conjuntas de diseño, múltiples bucles de feedback, y repetidas discusiones sobre a qué se debe parecer el término “hecho”.

El examen PMI-ACP comprobará tu competencia y comprensión de los siguientes aspectos sobre la implicación de los stakeholders. Intentaremos recordar este resumen mientras continuamos, y pensaremos como cada concepto se relaciona con estos elementos.

Conseguir los stakeholders correctos. Los proyectos no resultarán exitosos sin los stakeholders apropiados, por lo tanto debemos esforzarnos en conseguir reunir a la gente que necesitamos. Por supuesto, necesitamos a la gente correcta en el equipo de desarrollo, pero también necesitamos tener acceso a gente en la organización que pueda ayudarnos con más efectividad a entender los requisitos del proyecto y tomar las decisiones necesarias para hacer avanzar las cosas. A menudo la mejor gente es la más ocupada. Por lo tanto, podría ser beneficioso para el proyecto pagar para aumentar estos recursos (utilizando a más gente para ellos) para ayudarlos a completar sus otras obligaciones, de manera que tengan más tiempo para contribuir al proyecto.

Cementar la implicación de los stakeholders. Es esencial para los stakeholders el estar comprometidos con el proyecto, por lo tanto debemos hacer todo lo necesario para que su implicación sea sencilla. Por ejemplo, podemos documentar en el gráfico del proyecto como de importante es la implicación de varios de los diferentes stakeholders. Podemos también indicar los beneficios o los problemas derivados de la implicación de los stakeholders (o de la falta de ella) en los informes de situación del proyecto y otros documentos para la dirección del proyecto. Para cualquier método específico que elijamos, debemos asegurar que la implicación de los stakeholders es bien visible y se sigue durante el proyecto.

Gestiona activamente el interés de los stakeholders. Para mantener a los stakeholders comprometidos, debemos involucrarnos activamente. Esto significa que tomamos acciones para reconocer y recompensar la implicación de los stakeholders, como por ejemplo celebrando los hitos del proyecto con ellos, hablando con sus responsables sobre como recompensar sus contribuciones, y asegurándose que el feedback relacionado con el proyecto, forma parte de sus revisiones de rendimiento.

Discutir frecuentemente el significado de "hecho". Ya que los proyectos con trabajadores del conocimiento, a menudo, crean productos y servicios intangibles, hay una gran probabilidad de que exista una gran brecha entre lo que quiere y describe el cliente y lo que el equipo de desarrollo interpreta. Es crítico salvar esta brecha anticipadamente y a menudo, con cada nueva idea presentada en el proyecto. Las discusiones frecuentes sobre el significado de "hecho" son esenciales si el equipo quiere evitarse sorpresas, brechas en las expectativas o productos mal aceptados.

Muestra el progreso y las capacidades. Los proyectos llevan mucho tiempo, y las personas, generalmente, quieren las cosas lo antes posible después de haberlas descrito. Por este motivo, es importante mostrar frecuentemente a los stakeholders como ha avanzado el proyecto. Estas demos o presentaciones, no solo nos permiten comprobar que estamos desarrollando lo correcto antes de llegar demasiado lejos, sino que además, muestran el progreso a nuestros clientes o sponsors. A la vez, los stakeholders continúan comprometidos e informados sobre cuando se van completando las cosas.

Discutir abiertamente las estimaciones y capacidades. Una de las ventajas de comprometer al cliente y mostrarle partes del proyecto, tal y como éstas se construyen es que el ratio real de progreso del proyecto es altamente visible. En el lado negativo, el ratio real de progreso es, a menudo, inferior al deseado por el sponsor. Si se entrega lo suficientemente pronto, no obstante, las malas noticias pueden ser valiosas. Siguiendo y discutiendo las características entregadas versus a las que habría que haber entregado, el equipo y el cliente pueden tomar importantes decisiones basándose en datos sólidos.

2.3.1. Alineando la comprensión de los Stakeholders

Los métodos ágiles reconocen la importancia y dificultad de salvar la distancia entre lo que los clientes piden y como el equipo de desarrollo interpreta lo que han pedido. Se ha creado una gran variedad de técnicas para ayudar a facilitar las discusiones, llevar cualquier malentendido rápidamente a la superficie, y trabajar hacia el consenso y el acuerdo. En los proyectos ágiles, fallar está bien, siempre que el fallo sea rápido y barato y continúe siendo rápido el recuperar el proyecto. Idealmente, buscamos descubrir cualquier desconexión en el entendimiento al mínimo coste posible. Haciendo esto, ahorramos dinero a la organización, que puede ser invertido en otros proyectos.

2.3.1.1. Diagramas de estructura

Los modelos de diagramas de estructura son una popular manera de crear una rápida maqueta del producto. Por ejemplo, en el desarrollo de software, un diagrama de estructura representa las pantallas individuales y los cambios entre pantallas. Este diagrama ayuda a confirmar que todo el mundo tiene la misma comprensión del producto. Si hay discrepancias en la comprensión, el diagrama de estructura sirve como herramienta visual para los stakeholders para referirse y ajustar hasta conseguir el consenso. Los diagramas de estructura son una forma "prototipo de baja fidelidad". En otra palabras, son maneras rápidas y baratas de conseguir feedback sobre algo.

Los equipos ágiles pueden utilizar modelos de diagramas de estructura creados en herramientas como Visio, Powepoint, o pueden dibujar los modelos a mano alzada o pizarras o hojas de papel, que faciliten el rápido reposicionamiento de los cambios del flujo de trabajo. El propósito de estas herramientas es ayudar a clarificar el significado de "hecho" y validar el enfoque que el equipo planea seguir antes de que se utilicen grandes cantidades de tiempo construyendo incrementos del producto (potencialmente erróneos).

2.3.1.2. "Personas"

"Personas" son guías rápidas o recordatorios de los stakeholders, claves en el proyecto y sus intereses. Los proyectos de software, por ejemplo, acostumbran a crear "Personas" para los diferentes tipos de individuos que utilizan el sistema que está siendo construido. "Personas" puede estar basado en perfiles de individuos reales o composiciones de múltiples usuarios. Cuando se utilizan como herramienta de proyecto, "Personas" debe:

- Proporcionar una descripción arquetípica de los usuarios.
- Estar basado en la realidad.
- Estar orientado al objetivo, ser específico y relevante.
- Ser tangible y procesable.
- Generar focus.

"Personas" no es un sustituto para los requisitos, pero los aumentan. "Personas" ayuda al equipo a priorizar su trabajo, permanecer centrado en los usuarios, y ganar una visión interior de lo que los usuarios quieren ser. Estas herramientas ayudan a los miembros del equipo a enfatizar con los usuarios del producto o solución.

Ejemplo de “Persona”

<table>
<tr><td colspan="2">Nombre: Bob el cinéfilo</td></tr>
<tr><td>

Descripción:

Bob ama las películas. Por media, alquila 5 películas en una semana de su videoclub local.

A sus dos hijos también les gusta ver programas de TV. A menudo les gusta ver el mismo programa más de una vez, lo cual significa que Bob a veces tiene que pagar cargos adicionales.

La mujer de Bob tiene diferentes gustos que él para las películas y a veces se pasa mucho tiempo eligiendo una película.</td><td>Valores:

A Bob le gustaría poder alquilar películas desde el confort de su casa. Le gustaría poder buscar películas por título, actor, género y director. También le gustaría saber cómo han calificado a la película otros usuarios.

Está buscando algo con lo que pueda realizar visionados ilimitados de manera que sus hijos puedan ver varias veces los programas, sin tener que pagar cargos adicionales.

También apreciaría una característica de “recomendado” que les ayudara a él y a su mujer a elegir película.</td></tr>
</table>

“Personas” puede ayudar a mantener al equipo centrado en entregar las características que los usuarios encontrarán como valiosas, y esto lleva a una mejor toma de decisiones en el proyecto. Estas herramientas pueden también acortar las discusiones en el equipo. Por ejemplo, los miembros del equipo pueden referenciar una “Persona” que es familiar al resto del equipo. Una pregunta como “¿Necesitará esto Samantha?” o una declaración como “¡Samantha no va a necesitar esto!” pueden servir como un atajo para ayudar al equipo a tomar la decisión más rápido y mantenerlos conectados a la visión del proyecto.

2.3.1.3. Historias de usuario

Las historias de usuario son partes pequeñas, entendibles de funcionalidades del negocio. Los equipos de proyectos ágiles comúnmente confían en las historias de usuarios y una cartera de estas historias para ayudar a alinear las prioridades con las necesidades del negocio. Las historias de usuario a menudo se escriben en este formato:

"Como <puesto>, quiero <funcionalidad>, para que >beneficio para el negocio>".

Ejemplo: "Como cliente de PelículasOnline, quiero buscar películas por actor, para poder encontrar más fácilmente los películas que quiero alquilar"

Las ventajas de esta plantilla con puesto, función y beneficios para el negocio, es que fuerza al proyecto a identificar al usuario (¿Quién está preguntando esto?) y el beneficio para el negocio (¿Por qué estamos haciendo esto?) para cualquier parte de funcionalidad requerida. Pese a todo, los proyectos son bombardeados con requisitos que no tiene un propietario claro o un beneficio claramente conocido.

Otro formato, a menudo utilizado para requisitos no funcionales o basados en el sistema, es "Dar, cuando entonces". Por ejemplo:

Dar la cuenta es válido y la cuenta tiene un balance de MovieCredit mayor de $0.

Cuando el usuario reclama crédito para una película.

Entonces emite la película y reduce el balance de MovieCredit del usuario.

La manera en la que se escriben las historias de usuario es importante, pero hay también otros aspectos de las historias de usuario que pueden contribuir a su valor en el proyecto. El mnemotécnico INVEST, mostrado en la figura 29, se utiliza a menudo como un recordatorio de las características de historias de usuario efectivas.

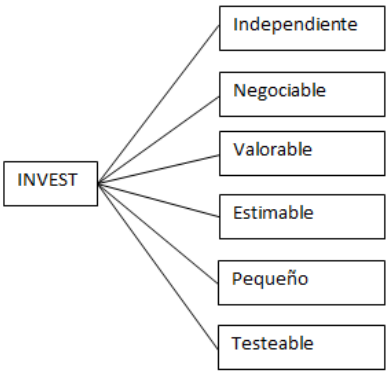


Figura 29: INVEST. (Fuente: Elaboración propia)

Veamos cada uno de los términos con más detalle.

>> I-Independent (independencia): Idealmente, queremos ser capaces de re-priorizar y desarrollar las historias de usuario en cualquier orden. Esto es difícil de conseguir, pero es un objetivo. Intentamos crear historias de usuario independientes que pueden ser seleccionadas por méritos, en lugar de ser arrastradas en la entrega porque otras historias de usuario dependen de ellas.

>> N- Negotiable (negociable): El equipo debe ser capaz de discutir las historias de usuario con los representantes del negocio y hacer concesiones en función del coste y funciones. Por ejemplo, "Imprimir" significa "presiona el botón "PrintScreen" y pega la imagen en un Word", o ¿el negocia necesita un documento total ente formateado con cabeceras y pies de página? Negociar las historias de usuario lleva a una mayor comprensión de los requisitos reales, costes y compromisos aceptables.

>> V- Valuable (Valorable): Si no podemos determinar el valor de un requisito, entonces debemos cuestionarnos porque es parte del proyecto. Incluso aunque una historia de usuario esté escrita de manera diferente a la antes comentada (posición, función, y beneficio para el negocio), una porción de la historia del usuario debe explicar el valor, o beneficio del requisito. Las historias de usuario que no indican claramente los beneficios para el negocio serán difíciles de priorizar, ya que las carteras, normalmente, están clasificadas según el valor del negocio.

>> E- Estimatable (estimable): Este término transmite la idea de que debemos ser capaces de estimar el esfuerzo de una historia de usuario. Si no es posible decir si el requisito necesitará un día o dos semanas de esfuerzo de trabajo, no seremos capaces de priorizar las historias de usuario basándonos en su coste/beneficio.

>> S- Small (pequeño): Las historias de usuario pequeñas son más fáciles de estimar y probar que las historias de usuario más largas. Tal y como la unidad de trabajo se hace más grande, disminuye la habilidad de las personas para estimar. También, en el momento en que una historia de usuario larga se reporta tarde, también llevará un tiempo largo el corregir o rehacer el trabajo. Por lo tanto las historias de usuario se deben mantener pequeñas. El rango común está entre medio día y diez días. Otra ventaja de las historias de usuario cortas es que, normalmente, pueden ser completadas dentro de una sola iteración. Pero esto no significa que la más corta sea la mejor. Las historias de usuario extremadamente cortas, por ejemplo aquéllas que lleven 2 horas o menos para completarse, deben ser evitadas, ya que existe un sobrecoste en crearlas y seguirlas.

>> T –Testable (comprobable): Si no podemos comprobar algo, ¿cómo sabremos cuando está hecho? Tener historias de usuario comprobables es importante para seguir el progreso ya que medimos éste, basándonos en cuantas historias de usuario han sido aceptadas.

Para resumir, estamos buscando partes del trabajo pequeñas, independientes, y valiosas – funciones que pueden ser negociadas con el negocio para encontrar el nivel correcto de coste versus rendimiento – y debemos ser capaces de estimar y probar estas partes fácilmente. Cuando se desarrollan sistemas de software, las historias típicamente deben ser cortes verticales de funcionalidades que pasan a través de todas las capas arquitecturales relevantes del sistema.

Por ejemplo, una historia de usuario que implica salvar y recuperar un balance de un cliente es preferible a una que se centra en imitar la pantalla de la cuenta del cliente. El primer ejemplo utiliza todos los componentes del sistema que la solución final utilizará (la representación, validación, negocio y capas de datos), mientras que el segundo ejemplo solo utiliza la capa de interface de usuario (presentación). A causa de que solo se centra en una capa, crear una imitación de la pantalla de cuenta del cliente no revelará ningún problema en las otras capas. Además, la historia de usuario no actuará como un indicador fiable de la verdadera velocidad de desarrollo, ya que otro trabajo continúa pendiente de ser realizado para poder conseguir la funcionalidad deseada.

Cartera de historias de usuario.

Después de que hayan sido creadas las historias de usuario, pueden ser organizadas en una cartera. Esta cartera de historias de usuario es una lista visible de trabajo a ser realizado, como se muestra en la figura 30:

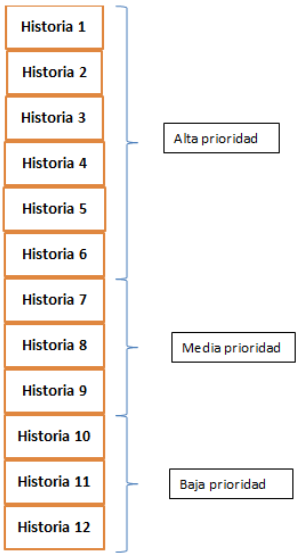


Figura 30: Cartera de historia de usuarios. (Fuente: Elaboración propia)

La cartera de historias de usuario ayuda a guiar las discusiones del equipo sobre las prioridades. También sirve como herramienta de planificación para gestionar las entregas e iteraciones, y ayuda a dirigir el foco del equipo durante las discusiones sobre el alcance y cuando se gestionan los cambios. Esta herramienta ayuda a coordinar el proyecto y mantiene al equipo trabajando hacia la misión acordada, mientras que continúa permitiendo revisiones y actualizaciones conforme llega nueva y mejor información.

Requisitos de jerarquía.

Un proyecto largo es en él mismo, una abrumadora y compleja cantidad de trabajo que no puede ser exactamente planificada o estimada por completo. Por lo tanto, debemos partir el proyecto en pequeñas partes, con requisitos cada vez más refinados, hasta que alcancemos el nivel suficientemente pequeño para estimar y construir los elementos. Esto se hace a través de la planificación iterativa en los proyectos ágiles (lo cual veremos más adelante). Aunque las historias de usuario son la herramienta de planificación más comúnmente discutida, no es lo único que se utiliza en los proyectos ágiles. En la jerarquía de requisitos de un proyecto, las historias de usuario están generalmente en el medio. La figura 31 es un ejemplo de una forma común de estructurar la jerarquía de requisitos:

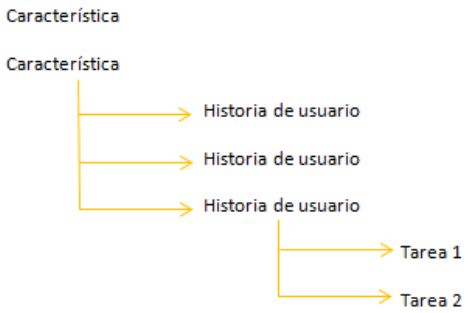


Figura 31: Jerarquía de requisitos. (Fuente: Elaboración propia)

Algunas metodologías extienden este modelo básico con la adición de epopeyas (largas historias de usuarios que necesitan una o más iteraciones). Desafortunadamente no hay una versión válida, única y universal de la jerarquía extendida de requisitos. Algunos equipos de proyecto tienen “epopeyas” por encima del nivel de historia de usuario. Una “epopeya” puede también estar por encima, por debajo, o simplemente reemplazar el nodo de la característica. En las figuras 32, 33 y 34 vemos diferentes maneras en las que se puede posicionar la “epopeya” en las jerarquías de requisitos.

Figura 32: Jerarquía de requisitos con epopeya. (Fuente: Elaboración propia)

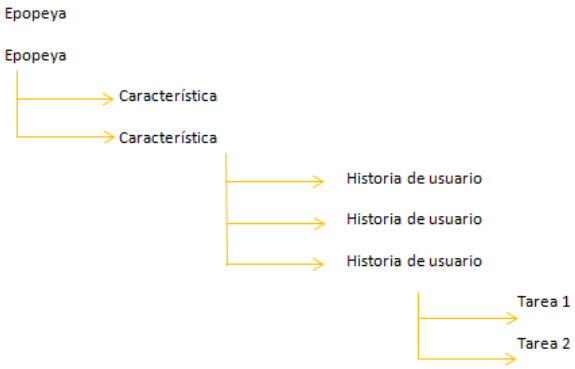


Figura 33: Jerarquía de requisitos con la "epopeya" por encima de las características. (Fuente: Elaboración propia)

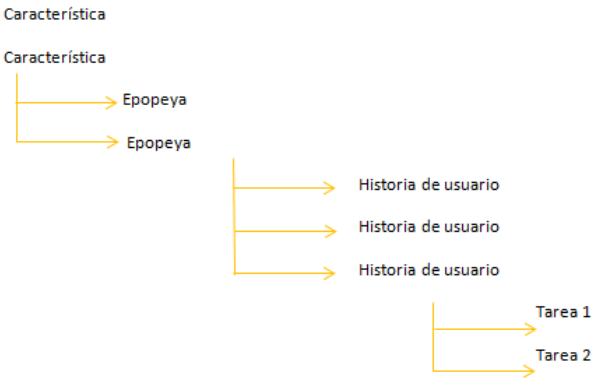


Figura 34: Jerarquía de requisitos con la "epopeya" por debajo de las características. (Fuente: Elaboración propia)

2.3.1.4. Mapas de historias

Como hemos visto con anterioridad con las entregas basadas en el valor, los mapas de historias son útiles herramientas para indicar la priorización del valor para el cliente, pero esta herramienta puede servir también como hoja de ruta del producto, para mostrar como las características se entregaran. Los mapas de historias son grandes herramientas para mostrar que saldrá en la primera entrega, en la segunda, y en las siguientes. Como tal, se utilizan habitualmente en las comunicaciones con los stakeholders y pueden ser publicados en muros como un emisor de información del plan del proyecto. Tienen el beneficio adicional de comprometer a la gente de una forma diferente que si la misma información se representara en un diagrama de Gantt, ya que el formato del mapa de historias es más visual y atractivo.

2.3.1.5. Incorporando los valores de los Stakeholders

Hablaremos de la incorporación de las prioridades de los sponsors y usuarios dentro de las prioridades y ejecución del proyecto. En otras palabras, veremos cómo alinear las prioridades del proyecto con las prioridades de los Stakeholders. Incorporar los valores de los Stakeholders quiere decir que nos aseguramos de que no estamos planificando o iniciando un trabajo que los Stakeholders no apoyan o valoran.

Una manera de incorporar los valores de los Stakeholders en un proyecto es comprometer a los representantes del negocio en la priorización de la cartera. Ejecutando el trabajo en el orden en el que los representantes del negocio lo han especificado (teniendo en cuenta las necesarias dependencias y la mitigación del riesgo, naturalmente), hace que el proyecto entregue los elementos de más alta prioridad primero.

Otra manera de incorporar los valores e intereses de los Stakeholders en las actividades del proyecto es invitar a nuestros Stakeholders a reuniones de retrospectivas y planificación. Al final del día, nuestros clientes y sponsors son los que determinan si el proyecto es un éxito o un fracaso, por lo que es inteligente basar el trabajo y las acciones de los equipos en lo que los Stakeholders valoran.

2.3.1.6. Gestión de los Stakeholders

En la gestión de los Stakeholders se trata de cuidarlos exhaustivamente, incluyendo los esfuerzos iniciales para identificarlos.

Algunos papeles son obvios, por ejemplo, típicamente sabemos quiénes serán los sponsors, usuarios, y los gestores del departamento. No obstante, algunos de los otros papeles pueden ser más complicados de identificar, como los auditores, los proveedores de entradas al proyecto, o los consumidores de las salidas. Es importante identificar a todos los Stakeholders y gestionar efectivamente su involucración en el proyecto, porque excluyendo a cualquiera de ellos puede poner en riesgo el éxito del proyecto.

Si los Stakeholders del proyecto son nuevos en los métodos ágiles, necesitarán cierta formación básica sobre cómo funcionan los proyectos ágiles para ayudarles a comprender el enfoque, abordar cualquier mito sobre éstos, y gestionar sus expectativas. Esta formación debería incluir metas, valores, prácticas y beneficios del enfoque ágil para que entiendan porque se ejecuta el proyecto de esta manera.

Es normal para cualquier cambio el ser realizado con algún grado de escepticismo y cautela, lo cual no es necesariamente algo malo. Después de todo, el mundo sería caótico si adoptásemos cualquier nueva idea, sistema operativo, o sugerencia a cambiar de manera sistemática. Una ligera resistencia al cambio es beneficiosa ya que ayuda a asegurar que solo los cambios que valen la pena prevalecen. Por lo tanto, cuando estamos inicialmente comprometiendo a los Stakeholders en un proyecto ágil, necesitamos reconocer algunas de las preocupaciones que los Stakeholders pueden tener. A continuación veremos algunos ejemplos de preocupaciones típicas:

- **Ejecutivos y sponsors del proyecto.** Los ejecutivos y los sponsors están a menudo preocupados sobre el riesgo de fallo debido al uso de nuevas prácticas y enfoques de planificación contrarios a la intuición, aunque el enfoque ágil provee en la actualidad un mejor feedback para prevenir el fallo.

- **Managers.** Los managers temen la pérdida de control o la erosión de su papel cuando los proyectos asumen un enfoque ágil.

- **El equipo de proyecto.** El equipo de proyecto puede resistirse a tener un nuevo método forzado por la dirección.

- **La comunidad de usuarios.** La comunidad de usuarios a menudo se preocupa de no obtener todas las características que quieren o necesitan, o que las iteraciones tempranas puedan resultar en productos de poca calidad.

- **Grupos de soporte.** Otros grupos pueden estar preocupados sobre una aparente pérdida de control, peticiones continuas para que se involucren, o la falta de un punto claro de final.

Además de los esfuerzos iniciales para identificar a los Stakeholders, formarlos, y reconocer sus preocupaciones, la gestión de los Stakeholders implica continuar comprometiéndolos en el proyecto así como este progresa. Uno de los beneficios de la iteraciones de dos a cuatro semanas es que son lo suficientemente cortas para prevenir la pérdida de interés de los Stakeholders en el proceso. Durante una iteración, nos reunimos con los Stakeholders, nos ponemos de acuerdo en lo que se debe hacer, hacemos parte del trabajo, y dos o cuatro semanas después les mostramos que hemos desarrollado. Ya que las iteraciones breves son buenas para mantener el compromiso de los Stakeholders, son preferibles a las fases de desarrollo largas en las que la visibilidad es menor y en las que el interés de los Stakeholders tiende a disminuir.

Una de las razones principales para mantener a los Stakeholders comprometidos es asegurar que escucharemos las peticiones de cambio lo antes posible. Un diálogo continuo con nuestros Stakeholders nos ayudará también a identificar riesgos potenciales, defectos y problemas. Los métodos ágiles proveen múltiples puntos de contacto con los Stakeholders y eventos de comunicación deliberados designados para facilitar el feedback temprano y continuo.

Por ejemplo, en Scrum, las reuniones de planificación y revisión del sprint actúan como pilares de una iteración, durante la cual el equipo inicia y revisa la involucración de los Stakeholders. Otros métodos ágiles tienen también eventos similares. Un punto para recordar cuando comprometemos a la gente en estos eventos es que no todos los Stakeholders pueden tratarse de la misma manera. Algunas personas pueden causar problemas y ser impedimentos para el progreso del proyecto. En estos casos, el ScrumMaster, el gestor del proyecto, o cualquier otra persona designada, necesita utilizar sus habilidades de inteligencia emocional para intentar comprender los problemas de estos Stakeholders y encontrar un camino positivo para comprometerlos con el proyecto. SI esto no es posible, será necesario intentar escudar al equipo de su corrosiva influencia.

Otro aspecto de la gestión de los Stakeholders es establecer un proceso para escalar los problemas que necesitan un nivel alto de autoridad para resolverse. Lograr el acuerdo sobre este proceso es esencial para mantener el funcionamiento del proyecto sin problemas. Entonces, si el equipo encuentra un problema para el cual no tienen autoridad o influencia suficientes para resolverlo, deben ser capaces de seguir el procedimiento establecido para rápidamente escalarlo al Stakeholder apropiado para su resolución.

2.3.1.7. Gestión de los proveedores

La gestión de proveedores se centra en los Stakeholders externos a la organización pero involucrados en el proyecto ya que están proveyendo algún producto o servicio. Al igual que con los Stakholders internos, debemos ser cuidadosos a la hora de seleccionar los proveedores y podríamos necesitar formarlos sobre los aspectos singulares de trabajar en una ambiente ágil.

Si el proveedor necesita utilizar un enfoque ágil para participar en el proyecto, esto debe ser remarcado en la solicitud de propuesta (RFP). Dependiendo de su papel en el proyecto, no obstante, puede que algunos proveedores no necesiten utilizar o comprender las prácticas ágiles. Antes de establecer como requisito que los proveedores deben estar familiarizados con los métodos ágiles, o desarrollar un evento formativo para ellos, nos debemos preguntar si merece la pena. Si el proveedor desempeña un papel menor, tal vez no lo sea. Para aquéllos que jueguen papeles mayores o críticos, probablemente si merezca la pena. Como todas las decisiones, debemos considerar los beneficios/costes de llevarla a cabo.

Ya que los proyectos ágiles reconocen que los requisitos pueden cambiar y que el campo de aplicación es negociable, los contratos tradicionales de compromiso de los proveedores basados en especificaciones formales son problemáticos. En lugar de eso, los proyectos ágiles, normalmente utilizan los modelos de contratación que hemos visto anteriormente, Entrega Basada en el Valor, incluyendo paquetes de trabajo de precio fijo y contratos de precio fijo graduado.

La frecuente discusión de que se entiende por “hecho”.

Antes de continuar con el siguiente punto, veremos otro concepto esencial sobre alinear la comprensión de los Stakeholders – la definición de “hecho”. Crear una definición compartida de “hecho” es crucial para gestionar las expectaciones de los stakeholders. Esta iniciativa afectará al proyecto desde el nivel de historias del usuario (p.e. “Hecho” quiere decir desarrollado, documentado, y con la aceptación del usuario comprobada), hasta el nivel de entrega (p.e. “La primera entrega se considerará hecha cuando el sistema Alpha se reemplace y no haya defectos de prioridad 1 o peticiones de cambio”), y todo el camino hasta el nivel de entrega final del proyecto (p.e. “Hecho para el proyecto, significa que todas las características de prioridades altas y medias están implementadas, han pasado dos meses de uso sin fallos o problemas, y el proyecto recibe puntuaciones de satisfacción mayores del 70 por ciento desde la comunidad de usuarios”).

Para los proyectos de software, James Shore proporciona la siguiente lista de elementos que se deben discutir y chequear antes de que declaremos que cualquier cosa está “hecha”.

- **Testeado: ¿**Han finalizado todas las pruebas de unidad, integración y cliente?
- **Codificado:** ¿Se ha escrito ya todo el código?
- **Designado:** ¿Ha sido reprogramado el código a satisfacción del equipo?
- **Integrado:** ¿La historia del usuario funciona completamente (normalmente interface de usuario a base de datos) y encaja en el resto del software?
- **Construido:** ¿El script de construcción incluye todos los módulos nuevos?
- **Migración:** ¿El script de construcción actualiza el esquema de la base de datos si es necesario? ¿El instalador migrará datos cuando sea apropiado?
- **Revisado:** ¿Los clientes han revisado la historia de usuario y confirmado que satisface sus expectativas?
- **Fijado:** ¿Han sido reparados todos los fallos o programada su propia historia de usuario?
- **Aceptado:** ¿Los clientes aceptan que la historia de usuario está acabada?

Los métodos ágiles funcionan mejor cuando realizas un pequeño progreso en cada aspecto de tu trabajo cada día, en lugar de reservando los últimos días de la iteración para conseguir acabar las historias de usuario. Tal y como vamos desarrollando las historias de usuario y características, nos debemos asegurar que estamos involucrando al cliente local y que los aspectos varios del proyecto trabajan juntos.

2.3.2. Comunicándose con los Stakeholders

Ya que el trabajo es a menudo invisible en los proyectos con trabajadores del conocimiento, no podemos decir el estado de las cosas con facilidad, mirando alrededor. Por lo tanto, es crítico para el equipo de proyecto comunicarse con frecuencia con los stakeholders para asegurar que todo el mundo está en lo mismo y actualizado. Además, los fallos del proyecto pueden trazarse a las comunicaciones de fallo. En muchos proyectos, a lo largo del proceso, ocurre un problema que no se comunica a tiempo.

En lugar de corregirse el problema, las cosas en el proyecto van peor cada vez. Para ayudar a combatir estos problemas potenciales, los métodos ágiles proveen de una serie de herramientas y eventos de comunicación.

2.3.2.1. Gestión de las comunicaciones

La vía preferida de comunicación en los proyectos ágiles es el cara a cara (F2F). Las comunicaciones F2F tienen el mayor ancho de banda de todas las comunicaciones; esto quiere decir que las comunicaciones F2F pueden transferir la mayor parte de la información en un tiempo dado. La comunicación F2F también permite las preguntas y respuestas inmediatas, mientras que los métodos estáticos, como la documentación en papel, no lo permiten. El diagrama de efectividad de la comunicación de Alistair Cockburn, compara algunos de los métodos de comunicación más efectivos, incluyendo la documentación en papel, los métodos multimedia, intercambios de e-mail, conversaciones telefónicas y comunicaciones F2F.

Las comunicaciones basadas en papel están valoradas más bajo en términos de efectividad de la comunicación. Esta forma de comunicación es más lenta de crear, y por esto, no da opción a la interactividad, los documentos deben escribirse de tal manera que todos los que los reciben puedan entender la información, sin importar su nivel de conocimientos o experiencia. Los documentos en papel son también peores en la escala de riqueza, o temperatura, lo cual significa que no transmiten mucha emoción.

Ahora comparemos las comunicaciones en papel con las F2F, tal como dos personas trabajando juntas en una pizarra en blanco. En las comunicaciones F2F, los dos participantes pueden rápidamente crear un acceso rápido a la información cuando ambos comprenden el concepto, y pueden preguntarse el uno al otro y obtener feedback inmediato. Estas conversaciones también permiten la transmisión de emociones a través de comunicaciones no verbales como gestos, expresiones faciales, y el tono de voz. En las comunicaciones F2F, podemos decir rápidamente si las personas están confundidas, enfadadas, o apasionadas por algo.

No obstante, mientras que preferimos el F2F por su eficiencia y alta transmitividad, no siempre podemos confiar en ellas; hay veces en las que la información debe ser registrada en papel.

2.3.2.2. Radiadores de información

"Radiadores de información" es un término paraguas para un número altamente visible de maneras de mostrar información, incluyendo grandes cuadros, gráficos, y resúmenes de datos del proyecto. Estas herramientas, llamada a veces "controles visuales", informan rápidamente a los Stakeholders sobre el estatus del proyecto, y se muestran habitualmente en áreas de alto tráfico para maximizar su exposición. El término "radiador de información" fue desarrollado por Alistair Cockburn en contraste con la práctica de buscar información del proyecto fuera en "neveras de información", en las cuales nadie sabe que está pasando. En lugar de esto, los radiadores de información utilizan cuadros o gráficos de alta visibilidad para transmitir, o radiar, información sobre el proyecto rápidamente a cualquiera que esté interesado en ella.

El tipo de datos que debe ser mostrada en un radiador de información incluye:

- Las características entregadas hasta la fecha contra las características que restan por ser entregadas.
- Quien trabaja en que.
- Las características seleccionadas para la iteración actual.
- Indicadores de velocidad y defecto.
- Encuentros retrospectivos.
- Registros de riesgo.

Los mapas de historias, vistos anteriormente, y los gráficos "burn down" y "burn up", que veremos ahora, son otros ejemplos comunes de radiadores de información.

2.3.2.3. Gráficos "Burn down y Burn up"

Los gráficos "Burn down y Burn up" se utilizan para mostrar progreso y para ayudar a determinar cuando el proyecto (o una entrega dentro del proyecto) está completado.

Los gráficos "Burn down" muestran el esfuerzo restante estimado en el proyecto, y los gráficos "Burn up" muestran que ha sido entregado. Esto significa que cuanto más trabajo se complete, un gráfico "Burn down" mostrará un indicador de progreso moviéndose hacia abajo para indicar la reducción de la cantidad de trabajo que continua necesitando ser realizado. En contraste, el indicador de progreso en un diagrama "Burn up" se moverá hacia arriba, para mostrar el incremento de la cantidad de trabajo completado.

Los gráficos “Burn down” muestran típicamente el tiempo estimado restante o, si el equipo de proyecto está siguiendo el progreso en puntos, los puntos restantes estimados. El gráfico “Burn down” se puede crear en Excel. Mirando el gráfico anterior podemos estimar que el proyecto se completará (en otras palabras, alcanzará las 0 horas o esfuerzo estimado restante) en Febrero.

Un inconveniente de los diagramas “Burn down” es que pueden mezclar el alcance y las variables. Si volvemos a mirar el primer gráfico, podemos ver que el progreso es lento (la línea es plana) entre el 15 de Diciembre y el 29 de Diciembre. Esta bajada podría haber sido causada por las personas que cogieron vacaciones, o puede ser el resultado de que el equipo esté descubriendo más alcance para el proyecto.

No podemos discernir la razón para este enlentecimiento del progreso mirando el diagrama “Burn down”. En contraste, los diagramas “Burn up” separan normalmente el progreso del alcance, haciendo estos factores fáciles de distinguir.

La figura 35 es un ejemplo de un diagrama “Burn up”, en el cual podemos ver que el alcance se incrementó primero de 300 puntos a 330, y eventualmente a 350.

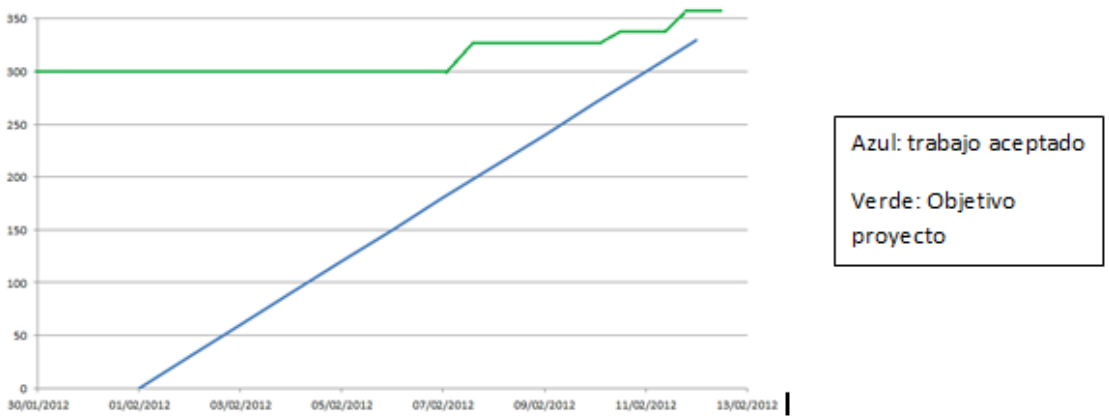


Figura 35: Diagrama burn up. (Fuente: Elaboración propia)

Si los datos en este diagrama “Burn up” se conectan con un diagrama “Burn down”, veremos áreas cuando la línea de ratio de esfuerzo se vuelva plana, pero no seremos capaces de decir a partir del diagrama “Burn down” si deberíamos atribuir el que se vuelva plana a un incremento en el ámbito o un enlentecimiento del progreso. Separando progreso y ámbito, los diagramas “Burn up” ofrecen una visión interna adicional en las varianzas del campo de aplicación.

El diagrama “Burn up” de la figura anterior muestra el trabajo aceptado o completado. ¿Pero qué hay del trabajo en progreso? Si añadimos el trabajo en progreso al diagrama “Burn up”, para seguir tanto el trabajo empezado como el acabado, creamos un diagrama de flujo acumulativo (CFD). La figura 36 es un ejemplo de CFD.

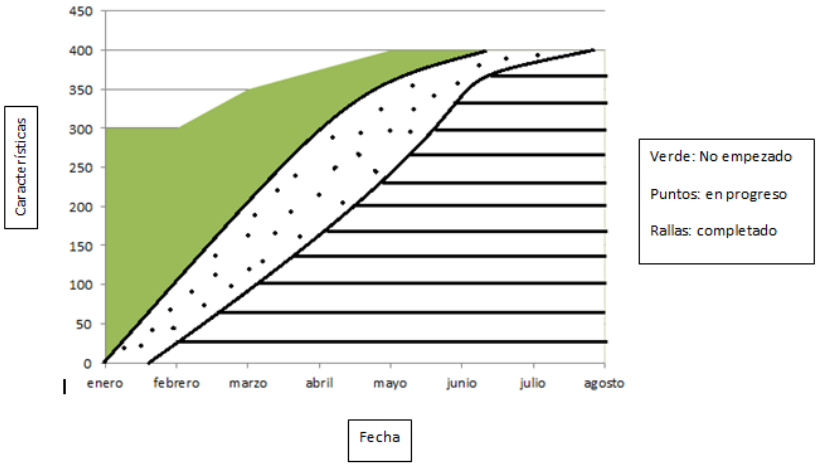


Figura 36: Ejemplo de CFD. (Fuente: Elaboración propia)

En la figura anterior, podemos ver como el ámbito global (la sección verde) se incrementa de 300 características a 400 durante el proyecto. El CFD también muestra el trabajo en progreso (la sección punteada) y el trabajo completo (la sección rallada).

El ratio real de progreso es el gradiente de la línea en lo alto de la sección “Completed”, ya que indica el ratio de características completadas.

2.3.2.4. Velocidad

La velocidad es la medida de la capacidad de un equipo para trabajar en cada iteración. Este indicador nos ayuda a medir cuanto trabajo es capaz de hacer el equipo, basándose en el número de historias de usuario completadas en la pasada iteración. También nos provee de una manera de comunicar que hemos completado, que seremos capaces de completar y cuando esperamos que se complete el proyecto (o entrega). La velocidad se mide en cualquier unidad que el equipo utilice para su trabajo, por lo que estas unidades pueden ser horas, días, puntos, etc…

Cuando la velocidad se sigue a través de múltiples iteraciones, este indicador se puede utilizar para determinar cuándo un proyecto probablemente puede completarse. Por ejemplo, si la velocidad del proyecto tiene un promedio de 50 puntos, y la cartera contiene 500 puntos de trabajo no desarrollado, debemos dividir 500 por 50 (500 / 50 = 10). El dato indica que el proyecto puede ser completado en 10 iteraciones más.

La velocidad usualmente varía mucho en las primeras iteraciones, y entonces empieza a estabilizarse. Esto es así, porque los miembros del equipo deben acostumbrarse a trabajar juntos, familiarizarse con las herramientas del proyecto, y sentirse bien interactuando con los Stakeholders del proyecto.

Mientras que puede parecer lógico predecir que la velocidad siempre tenderá a incrementarse tal y como el equipo gane experiencia, la velocidad normalmente se estabiliza.

Una de las razones que justifican lo que acabamos de comentar es que, así como crece el producto que se está desarrollando, hay más cosas a mantener, refactorizar, y posiblemente se deba ofrecer soporte técnico si las versiones iniciales del producto ya se han desplegado.

2.3.2.5. Modelado ágil

El término “modelado ágil” se refiere a varias técnicas de modelado que son comúnmente utilizadas en proyectos ágiles. Mientras que los modelos son importantes en los métodos ágiles, su valor principal a menudo recae en las discusiones y creación del modelo, más que en la entrega final. Como un reflejo de esto, los modelos ágiles se esbozan a menudo en pizarras y se fotografían entonces como una manera de registrarlos. El valor es la creación, no la preservación del modelo en una herramienta de modelado especializada.

Como los modelos ágiles son típicamente ligeros o “apenas suficientes”, capturan el diseño sin la necesidad de pulirlo más. Podemos crear un modelo para propósitos predictivos para ayudar a clarificar un diseño e identificar los problemas, riesgos y elementos que necesitamos probar. Podemos utilizar el modelado para reflejar los propósitos para investigar problemas y ayudar a encontrar soluciones.

2.3.3. Utilizando habilidades sociales criticas

A menudo es más difícil de dominar las habilidades sociales que las habilidades técnicas. Las habilidades sociales críticas se incluyen en el examen PMI-ACP precisamente porque son críticas. Las habilidades sociales pobres pueden desmoralizar y desencantar rápidamente a un equipo técnicamente fuerte, mientras que alguien que utilice con efectividad sus habilidades sociales, puede obtener resultados increíbles de equipos medios.

Utilizamos el término “Habilidades sociales críticas” en lugar de “talento social crítico” porque a diferencia de los talentos, con los que hemos nacido, las habilidades pueden ser adquiridas y mejoradas a través de la formación y la práctica. A continuación veremos las habilidades sociales críticas que se probarán en el examen PMI-ACP:

- Negociación
- Escucha activa
- Métodos de facilitación
- Globalización, cultura, y diversidad de equipos.
- Resolución del conflicto.
- Equipos distribuidos.
- Modelos de decisión participativos.

2.3.3.1. Negociación

La negociación ocurre a través de los proyectos ágiles, especialmente cuando se discuten los requisitos o prioridades de las características y a que se debe parecer “hecho”. Por ejemplo, digamos que un cliente pide la habilidad de imprimir los códigos de producto. Negociando este requisito, el equipo presenta opciones al cliente.

La opción de más bajo coste puede ser utilizar la función de “imprimir pantalla” en la pantalla de códigos del producto, pegar el pantallazo en un Word, y entonces imprimir este documento para obtener los códigos de producto. Un solución con un coste más elevado puede ser un informe con formato con cabeceras, pies de página, y números de página. También puede haber opciones adicionales que caigan entre estos dos extremos. El equipo y el cliente pueden negociar las ventajas y desventajas entre funcionalidad y coste para llegar a una solución balanceada.

La negociación en los proyectos ágiles no tiene por qué ser – y habitualmente no lo es - un todo o nada (p.e. con un ganador y un perdedor). En lugar de eso, las negociaciones sanas permiten a cada parte investigar las ventajas y desventajas y presentar perspectivas de alternativas. Debe haber una oportunidad para cada punto de vista o caso de negocio para ser descrito totalmente. Las negociaciones son más efectivas cuando las iteraciones entre los participantes son positivas y cada uno puede obtener algo.

2.3.3.2. Escucha activa

La escucha activa consiste en escuchar lo que alguien está realmente intentando transmitir, en lugar de atender únicamente al significado de las palabras que está diciendo. La expresión, "haz lo que quiero decir, no lo que digo" ilustra este concepto. En los proyectos ágiles, necesitamos escuchar el mensaje, no únicamente la cadena de palabras que se está diciendo.

La escucha activa es una habilidad que puede mejorarse con la práctica. De acuerdo con el libro "Co-Active Coaching: Changing business, transforming Lives" nuestra capacidad de escucha progresa a través de tres niveles, como se muestra en la figura 37:

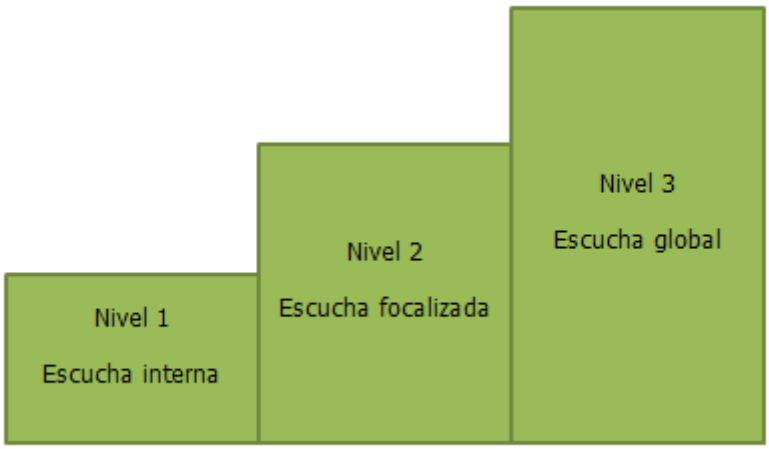


Figura 37: Capacidad de escucha. (Fuente: Elaboración propia)

Nivel 1: Escucha interna.

Escuchamos las palabras que se están hablando, y aunque podemos estar muy atentos, las interpretamos bajo nuestra propia visión. Cuando escuchamos, estamos pensando "¿Cómo me va a afectar esto?" y perdemos el mensaje real del que está hablando. Por ejemplo, si alguien en nuestro equipo comienza a hablar sobre los retos con un nuevo miembro del equipo, podríamos estar pensando en nuestro objetivo de integrar rápidamente al nuevo miembro y ponerlo al día y perder lo que el orador está intentando decir sobre el nuevo miembro.

Nivel 2: Escucha focalizada.

Cuando escuchamos a este nivel, debemos dejar atrás nuestros propios pensamientos y ponernos en la mente del que habla. Enfatizaremos con sus pensamiento, experiencias, y emociones tal y como nos vaya contando la situación. En nuestro ejemplo de la persona que está encontrando desafíos con un nuevo miembro del equipo, seremos capaces de enfatizar con los sentimientos de la persona que habla y reconocer que la formación de este nuevo miembro ha añadido a su carga de trabajo un tiempo estresante. Buscaremos indicadores emocionales en las palabras y pausas, la voz y el tono, y las expresiones faciales de la persona que habla, para obtener más información de cómo se siente la persona con lo que está describiendo.

Nivel 3. Escucha global.

Cuando escuchamos a este nivel, nos basamos en el enfoque del nivel 2 y añadimos un nivel más alto de concienciación, como una función de antena, para recoger cualquier sutil indicador ambiental y físico. Estos indicadores pueden incluir los movimientos o postura del hablante, su nivel de energía, y la atmósfera o ambiente en la sala. Nos daremos cuenta de factores como si el hablante está expresando la información abiertamente frente a los otros o de manera privada, los gestos de otros que están dentro del alcance de oído (p.e. ¿parece que coinciden con lo que se dice, o están apartando la mirada o distanciándose de la conversación?), y muchos otras pistas sutiles que nos ayudarán a entender el contexto más completo de la información que se está compartiendo. En nuestro ejemplo, en el que el hablante nos está hablando sobre el nuevo miembro, reconocemos la importancia y naturaleza sensible de la conversación ya que viene a discutir el problema de manera privada. También notamos que está tensa, cerrando los puños, y poniéndose en pie y caminando mientras habla. Teniendo estas pistas en cuenta, podemos ayudarla a reconocer y explorar sus sentimientos para comprender mejor su reacción a trabajar con ese nuevo miembro.

Formar a los equipos ágiles requiere que escuchemos con atención a las personas. Y no podemos escuchar cuando estamos hablando nosotros, por lo tanto, escucha más y habla menos. Deja hablar a los demás. Intenta contar hasta 10 para dar tiempo a la otra persona para sentirse cómoda y hablar. En general, no estamos aprendiendo mientras nuestras bocas hablan.

2.3.3.3. Métodos de facilitación

Las preguntas sobre métodos de facilitación en el examen PMI-ACP son esencialmente para comprobar si entiendes como dirigir reuniones y talleres. Cuando se facilita una reunión o sesión (o cuando se encuentren estas preguntas en el examen), debemos tener en mente lo siguiente:

- **Metas:** Las personas a menudo sienten que las reuniones les hacen perder el tiempo, especialmente si la reunión trata un amplio abanico de temas y los participantes no entienden porque tienen que estar allí o en que van a contribuir. Establecer una meta clara para cada reunión o sesión de taller puede ayudar a las personas a comprometerse en las discusiones desde el inicio. Además, teniendo una meta clara y manteniendo a todo el mundo centrado en esa meta, en lugar de permitir que la sesión se desvíe del tema, puede acortar el tiempo de la sesión, haciendo que la discusión sea más valiosa para todos los implicados.

- **Reglas.** Establecer unas reglas básicas es otra técnica importante para dirigir sesiones efectivas. Por ejemplo, debe haber reglas con respecto al uso de teléfonos móviles, empezar y acabar las sesiones a tiempo, y respetar los puntos de vista de todos los participantes. No obstante, no es suficiente con establecer las reglas, estas se deben aplicar en cada sesión.

- **Tiempo:** El tiempo es siempre importante cuando estamos intentando tener a un grupo de personas juntas, y puede ser fácil perder la noción del tiempo cuando una sesión está en marcha. Por lo tanto, la duración de la sesión debe ser establecida de antemano, y alguien debe designarse como controlador del tiempo. También es útil determinar anticipadamente cuando se realizará el descanso de la sesión.

- **Asistencia:** El facilitador de la reunión necesita asegurarse de que la reunión es productiva y que todo el mundo tiene la oportunidad de contribuir. Además de mantener al grupo centrado en el objetivo de la reunión y hacer cumplir las reglas, esto puede incluir el asegurarse de que los miembros junior o más tímidos tienen la oportunidad de expresar sus pensamientos, haciendo frente a los participantes más agresivos o dominantes, y manteniendo la sesión fluyendo sin problemas.

2.3.3.4. Globalización, cultura y diversidad de equipos

Así como se expanden nuestras opciones de comunicación y se vuelven menos caras, y nuestras bases de clientes y socios se amplían, factores como la globalización, la cultura, y la diversidad de los equipos se vuelven más importantes. No es poco común tener miembros en el equipo de tres o cuatro continentes trabajando en el mismo proyecto. Como el trabajo de conocimiento es invisible, confiamos principalmente en la comunicación para compartir la información del proyecto. Diferentes culturas aportan desafíos en las comunicaciones, pero también pueden aportar eficiencia a un proyecto ya que tienes un conjunto más amplio de recursos en el que elegir.

Incluso cuando personas de diferentes regiones del mundo hablan el mismo idioma, continúan habiendo diferencias a tener en cuenta.

Una buena manera para ayudar a los miembros de un equipo global a aprender sobre las preferencias y estilos de comunicación de cada uno es mantener una reunión inicial cara a cara. Es mucho más fácil mandar un e-mail, llamar o enviar un mensaje instantáneo a alguien a quien ya has conocido en persona. Y si es posible, extiende el modelo de eventos cara a cara para que los miembros del equipo trabajen juntos en una o dos iteraciones antes de que vuelvan a sus respectivos países.

Este modelo puede ser extendido mediante la organización de reuniones cara a cara de planificación y entrega así como el proyecto progresa. Otros modelos que trabajan bien incluyen una comisión de rotación, o asignación temporal, de los miembros del equipo entre diferentes sitios para dar a los miembros del equipo en cada región una oportunidad de trabajar en persona con personas de otros equipos y experimentar sus culturas.

Naturalmente, la diversidad puede ser un factor en nuestros equipos locales, también. La práctica XP de emparejar puede ayudar a superar los problemas relacionados con la diversidad. El emparejamiento da a los miembros una gran oportunidad de aprender como comunican el progreso los demás, plantean los problemas y las ideas. Cuando se realizan retrospectivas, debemos tener en mente las diferencias culturales. Por ejemplo, gente de algunas culturas puede no sentirse cómoda planteando los problemas del proceso actual cuando están en presencia de sus jefes. En estas circunstancias, puede ser útil pensar “¿Qué es lo que no hice bien?” También puede ser útil dejar las preguntas de retrospección abiertas al final de la reunión de manera que la gente pueda responderlas en salas de reuniones fuera del grupo, vía email o de otras formas anónimas.

2.3.3.5. Resolución de conflictos

El conflicto es una parte inevitable del trabajo del proyecto. Cuando las personas se juntan para solucionar proyectos, puede haber diferencias de opinión e intereses competitivos. Cierto nivel de conflicto es saludable, para asegurar que las ideas se prueban lo suficiente antes de adoptarlas. No obstante, necesitamos asegurarnos de que el conflicto no escala más allá del sano escepticismo y la burla amistosa, o acabaremos con un ambiente negativo y represivo en el equipo.

Crear un ambiente en el cual la gente use el conflicto de forma constructiva es una de las claves para conseguir comprometer a los Stakeholders en un proyecto. Debemos vigilar, por ejemplo, cuando el conflicto se mueva más allá de un debate normal y saludable y se convierta en algo destructivo y dañino para las relaciones entre el equipo. Speed B. Leas ofrece un entorno que ayuda a juzgar al seriedad de un conflicto y comprender mejor el camino que va desde el Nivel 1 (problema a resolver) al Nivel 5 (Guerra Mundial).

Nivel	Nombre	Característica	Tipo de lenguaje	Atmósfera/Ambiente
Nivel 1	Problema a resolver	La información se comparte y hay colaboración	Abierto y basado en hechos	>Las personas tienen diferentes opiniones y malentendidos. >Metas o valores en conflicto >No confortable, pero aun no emocionalmente cargada
Nivel 2	Desacuerdo	La protección personal triunfa sobre la resolución del conflicto	Reservada y abierta a interpretación	>La auto-protección se vuelve importante >Los miembros del equipo se distancian del debate >Las discusiones ocurren a parte (fuera del ambiente del equipo) >Se pasa de bromas sin maldad a bromas hirientes

Nivel	Nombre	Característica	Tipo de lenguaje	Atmósfera/Ambiente
Nivel 3	Concurso	Ganar importa más que resolver el conflicto	Incluye ataques personales	>El objetivo es ganar >Las personas se posicionan >Prospera el culparse unos a otros
Nivel 4	Cruzada	Proteger el grupo de uno mismo se vuelve el centro	Ideológico	>Resolver la situación no es suficiente >Los miembros del equipo piensan que las personas del otro lado no cambiarán y que deben ser eliminadas
Nivel 5	Guerra mundial	Destruir al otro	Poco o inexistente	> ¡Destruir! Es el grito de batalla >Los combatientes deben estar separados >No se puede tener un resultado constructivo.

Tabla 11: Niveles de conflicto

Comprender el entorno de Leas en los diferentes estadios de conflicto nos puede ayudar a mirar a las situaciones de una manera más objetiva, obviando nuestro propio juicio para ver que es lo que realmente está pasando. Identificar el estadio del conflicto también nos puede ayudar a determinar que acciones deberíamos tomar o que herramientas o técnicas podrían funcionar en las situaciones que se den.

Por lo tanto, cuando un equipo está en un estado de conflicto, debemos, en primer lugar, tomarnos cierto tiempo para observar el conflicto y asegurarnos de que estamos considerando las dos partes de la disputa, y no seguimos únicamente una reacción instintiva.

Debemos permitirnos tiempo para una observación adecuada, conversar, e intuir sobre los problemas antes de tomar una acción. Esto significa que al principio, únicamente escucharemos las quejas, sin intentar solucionarlas. Sentiremos la energía del grupo, y evaluaremos los niveles de conflicto. Estaremos atentos a las miradas, ojo entornados, y palabras que frenen conversaciones, para determinar si el conflicto ha salido a la superficie o todavía se encuentra enterrado.

Una vía para evaluar el nivel de conflicto es centrarse en el lenguaje que utiliza el equipo y comparar la conversación con los niveles de 1 a 5.Vemos en el lenguaje que se utiliza en cada nivel con más detalle:

- **Nivel 1 (problema a resolver):** El lenguaje es generalmente abierto y constructivo, y la gente usa, habitualmente, afirmaciones de hechos para justificar sus puntos de vista. Por ejemplo, los miembros del equipo pueden hacer declaraciones como, “Oh, veo lo que dices, pero continúo prefiriendo el otro enfoque, ya que en el pasado hemos visto menos fallos y menos trabajos rehechos usando esta técnica”.

- **Nivel 2 (Desacuerdo):** El lenguaje empieza a incluir la auto-protección. Por ejemplo, los miembros del equipo pueden hacer declaraciones como “Sé que piensas que mi idea no funcionará, pero ya hemos probado tu enfoque y nos dio muchos problemas”.

- **Nivel 3 (Concurso):** Los miembros de equipo comienzan a utilizar un lenguaje distorsionado, como las sobre generalizaciones, presunciones, y posiciones magnificadas. Pueden hacer declaraciones como, “Si no estuviera en el equipo…”.

- **Nivel 4 (Cruzada):** El conflicto se vuelve más ideológico y más polarizado. Los miembros del equipo empiezan a hacer declaraciones como “Están planificando mal” y “No vale la pena ni siquiera hablar con ellos”.

- **Nivel 5 (Guerra mundial):** El lenguaje es completamente combativo. Los miembros de equipos opuestos, raramente se hablan los unos a los otros, solo hablan con los miembros de su “lado”, expresando sentimientos como “Somos nosotros o ellos” y “Debemos vencerles”.

Una vez que hemos observado y analizado el nivel de conflicto, podemos decidir qué hacer. Si el conflicto está en un nivel del 1 al 3, no tomes ninguna acción para resolverlo. En lugar de eso, primero dale al equipo una oportunidad de arreglarlo ellos mismos. Si el equipo puede solucionar el problema por si mismo, habrá adquirido una habilidad muy importante. Es correcto tener algunas disconformidades durante el conflicto, ya que esta experiencia les preparará para auto gestionarse en conflictos similares en el futuro. No obstante, si la situación no mejora, y al contrario, parece empeorar, las siguientes guías pueden ser útiles para resolver el conflicto:

Nivel 1 (problema a resolver): Para un conflicto de Nivel 1, intenta construir un escenario de colaboración para mostrar los problemas de competencia y utilizar este escenario para construir el consenso alrededor de una decisión que todo el mundo apoye.

Nivel 2 (desacuerdo): En el nivel 2, la resolución del conflicto suele envolver el potenciar a los miembros del equipo relevantes para resolver el problema. Este enfoque crea el apoyo de los miembros del equipo a la decisión y restablece un sentido de seguridad al grupo.

Nivel 3 (concurso): A este nivel, el conflicto se ha convertido en acusador. Para ayudar a resolver el problema, necesitamos acomodar los diferentes puntos de vista de las personas. Aunque esto puede implicar el comprometer el trabajo a realizar, no deberíamos comprometer los valores del equipo.

Nivel 4 (cruzada): Resolver un conflicto de nivel 4 requiere diplomacia. Al haber estado rotas las comunicaciones entre los lados opositores durante mucho tiempo, el equipo necesitará a un mediador para transmitir los mensajes entre las diferentes partes. Nuestro foco debe centrarse en rebajar el conflicto un nivel o dos.

Nivel 5 (guerra mundial): Si un conflicto llega al nivel 5, podría no ser irresoluble. En lugar de intentar arreglarlo, debemos imaginar cómo podemos lograr que las personas vivan con ello. Solo en el Nivel 5 deberemos separar a las partes opuestas para prevenir que se hagan más daño los unos a los otros.

En resumen, el conflicto es normal e inevitable cuando la gente trabaja junta. Coaches, ScrumMasters y gestores a menudo se sienten obligados a resolver el conflicto, pero antes de ponerse a ello, lo mejor es observar la situación para obtener una mejor visión de los problemas. El modelo Leas nos puede ayudar a evaluar la severidad del conflicto. Debemos prestar atención al lenguaje que se está utilizando y dar una oportunidad al equipo de resolver el conflicto por ellos mismos. Si tenemos que intervenir, nos debemos centrar en rebajar el problema separando los hechos de las emociones y buscando maneras de ayudar a las personas a avanzar a pesar de sus diferencias.

2.3.3.6. Equipos distribuidos

Cuando los Stakeholders están distribuidos presentan un desafío para las comunicaciones cara a cara y para herramientas como los radiadores de información que se basan en la cercanía para trabajar mejor. No obstante, éste es un desafío con el que deben tratar muchos proyectos; las encuestas demuestran que más del 50 por ciento de los equipos ágiles tienen como mínimo un miembro en un lugar físico diferente del resto del equipo.

Para ayudar a tratar con este desafío, los equipos distribuidos pueden utilizar tecnologías como las conferencias de video, Skype, y otras herramientas para simular un ambiente de trabajo compartido y permitir a los Stakeholders distribuidos a hablar e interactuar como si sus colegas estuvieran al lado. Además, existen herramientas ágiles online que ofrecen características como tablas de tareas electrónicas y tablas de historias que pueden ser utilizadas por personas en múltiples lugares.

2.3.3.7. Modelos de decisión participativos

Los modelos de decisión participativos presentan diferentes maneras para comprometer al equipo en el proceso de toma de decisiones. La velocidad a la que tomamos decisiones y el nivel de acuerdo de nuestro equipo con esas decisiones impactará tanto en el rendimiento del proyecto como en la cohesión del equipo. También, ya que los proyectos con trabajadores del conocimiento no tienen productos tangibles emergiendo de una cadena de producción, los procesos de comunicación y decisión se vuelven más críticos para mantener a todo el mundo informado y comprometido.

Aunque los métodos ágiles utilizan muchas herramientas para promocionar la comunicación efectiva entre los Stakeholders (p.e. cercanía, reuniones diarias, cursos de planificación, retrospectivas, etc), no hay tantas herramientas para la toma de decisiones. Esto no significa que la participación de los Stakeholder en las decisiones no sea importante, no obstante. Si no preguntamos a los miembros del equipo sus opiniones cuando tomamos decisiones, corremos el riesgo de alejar a algunos miembros del equipo. Esto, a su vez, lleva a reducir los niveles de compromiso y participación de los Stakeholders, y puede acabar en la pérdida de una perspectiva importante que nos ayudaría a evitar obstáculos más adelante en el proyecto.

Los métodos ágiles favorecen el dar más poder al equipo y limitar el control de la dirección en los proyectos. Aunque el que un equipo tenga poder, aumenta la satisfacción y productividad de los Stakeholders, también plantea la necesidad de procesos efectivos de toma de decisión. Por lo tanto, ¿sin un dictador en el proyecto, cómo toman decisiones los equipos y siguen adelante?

La primera cosa que se debe comprender, es que no es realista esperar que el equipo logre el consenso en todos los problemas y todas las decisiones. Antes o después, necesitaremos algún mecanismo para tomar una decisión mientras hacemos que todo el mundo continúe comprometido con el proyecto. Este mecanismo son los modelos de decisión participativos. Veamos algunos ejemplos de estos modelos, así como las ventajas y desventajas de cada uno de ellos.

Votación simple.

Un método sencillo puede ser pedir al equipo que vote "a favor" o "en contra" de una idea levantando la mano. Aunque es una técnica muy sencilla, limita nuestras oportunidades de refinar la decisión resultante. Si buscamos un resultado rápido, este método evitará que el equipo discuta mejores alternativas. ¿Pero qué pasa si alguien tiene una sugerencia para ajustar las opciones que se están votando? La votación simple de "a favor" o "en contra" omite el poder mejorar el proceso como un paso esencial. Para ayudar a hacer frente a esta limitación, los equipos deben discutir sus pensamientos antes de votar – pero para muchas decisiones sencillas, esta discusión será un mal uso del tiempo del equipo. Viendo todo esto, este método no parece muy recomendable.

Pulgares arriba/abajo/de lado.

Pedir que se muestre el pulgar arriba/abajo/de lado alrededor de la sala, es una manera más eficiente de lograr una votación simple mientras se sigue permitiendo algún tiempo para discutir otras opciones. Con esta técnica, preguntamos a aquéllos que están manteniendo el pulgar de lado, porque no se pueden decidir por una u otra opción. A veces estas personas son neutrales a la idea, pero en otras ocasiones tienen un conflicto, preocupación o pregunta que necesita investigarse.

Este método es más rápido que hacer votar a todo el mundo, ya que muchas personas no tendrán preocupaciones y únicamente querrán continuar.

Espectro de decisión de Jim Highsmith.

Jim Highsmith esboza una gran herramienta de toma de decisiones en su libro "Agile Project Management: Creating Innovative Products". Utilizando el modelo de Highsmith, los miembros del equipo indican como se sienten sobre las decisiones colocando una marca en un rango de espectro que va desde completamente a favor, pasando por sentimientos entremezclados hasta totalmente no, o vetado. "El método Higsmith es efectivo ya que permite a la gente tanto indicar su apoyo a una decisión como expresar sus reservas al mismo tiempo. Es importante dar a las personas una oportunidad para expresar sus preocupaciones si esperamos alcanzar un acuerdo para avanzar, mientras continuamos respetando los puntos de vista disidentes y hacemos que todos estén comprometidos.

Este método invita a aquéllos que no están completamente a favor de una opción a compartir sus preocupaciones. A menudo, únicamente dando la oportunidad de registrar sus reservas es suficiente para permitirles comprometerse en una nueva dirección.

Votación "Fist-of-Five"

El método "fist-of-five" tiene la ventaja de la velocidad (similar al método de votación simple), mientras que continúa permitiendo a la gente indicar su nivel de acuerdo (similar al método del espectro de decisión). Utilizando el método "fist-of-five", la gente vota enseñando el número de dedos que indica su nivel de acuerdo.

Un pequeño problema con este método es que han surgido dos normas. Para hacer frente a este problema, debemos dejar claro desde el principio si los cinco dedos significan "completamente de acuerdo" o "no, completamente en desacuerdo". Una versión de este método, que ha sido popularizada por la American Youth Foundation, registra el nivel de apoyo por el número de dedos levantados, un puño (sin dedos) significa que no se apoya, mientras que levantar los cinco dedos significa que se apoya completamente y el deseo de llevar la carga.

La otra versión popular de este método usa los dedos votados para registrar el grado de resistencia u objeción a una idea. Con este método, el número de dedos levantados indica lo siguiente:

- **Un dedo:** "Apoyo totalmente esta opción"
- **Dos dedos:** "Apoyo esta opción con algunas reservas que probablemente no tenemos que discutir"
- **Tres dedos:** "Tengo preocupaciones que debemos discutir"
- **Cuatro dedos:** "Me opongo y quiero discutir el problema"
- **Cinco dedos:** (como la palma de la mano extendida en signo de stop): ""Detente, estoy en contra de esta decisión"

Con los modelos de decisión participativos, un punto clave a recordar es "no estar involucrado significa no estar comprometido". Necesitamos encontrar maneras de mantener a nuestros Stakeholders involucrados en las decisiones importantes del proyecto, incluyendo planificación de iteraciones y entregas, sesiones de estimación, y retrospectivas. Si las personas no están involucradas, no estarán comprometidas con la decisión y, finalmente, no estarán comprometidas con el proyecto.

2.3.4. Dirigir eficazmente

Los métodos ágiles son más humanísticos que mecánicos, como se evidencia por el valor ágil de “individuos e interacción por encima de procesos y herramientas”. El concepto de valorar a las personas por encima de los procesos va más allá de como gestionamos el trabajo a realizar en el proyecto; también afecta como organizamos y motivamos a los miembros de nuestro equipo, y como asumimos nuestro rol como líderes. El liderazgo se basa en aprovechar las motivaciones intrínsecas de las personas. Para ser líderes efectivos, necesitamos descubrir porque nuestra gente quiere hacer cosas, comprender que los motiva, y entonces alinear las tareas de sus proyectos y metas en consonancia. Cuando alineamos los objetivos del proyecto con los objetivos personales es cuando obtenemos los más altos niveles de productividad.

Gestión versus liderazgo

La gestión tiene un enfoque mucho más mecánico que el liderazgo; se preocupa de las tareas, control y velocidad. En contraste, el liderazgo asume un enfoque más humanístico en las personas y sus propósitos; se preocupa más de la potenciación, la efectividad, y hacer lo correcto. La siguiente tabla muestra las diferencias entre en enfoque de la gestión y del liderazgo:

Enfoque de gestión	Enfoque de liderazgo
Tareas/cosas	Personas
Control	Potenciamiento
Eficiencia	Efectividad
Hacer las cosas bien	Hacer las cosas correctas
Velocidad	Dirección
Practicas	Principios
Mandar	Comunicar

Tabla 12: Diferencias entre en enfoque de la gestión y del liderazgo.

Pero, ¿significa esto que el liderazgo es mejor que la gestión? ¿Podemos tener liderazgo sin gestión? No; definitivamente necesitamos los mecanismos de la gestión. Pero para ser verdaderamente efectivos, necesitamos la capa de liderazgo por encima de estos mecanismos. Podemos amplificar la productividad del equipo a través de la combinación de la gestión y el liderazgo.

2.3.4.1. Liderazgo de servicio

Los métodos ágiles promueven un liderazgo de servicio que reconoce que son los miembros del equipo, no el líder, formador, ScrumMaster, o Project manager, quienes realizan el trabajo y logran los valores del negocio. El enfoque del liderazgo de servicio redefine el rol del líder en relación con el equipo. Centra al líder en proveer al equipo de lo que necesiten, eliminando los impedimentos para progresar, y realizando tareas de soporte para maximizar la productividad del equipo.

Existen cuatro deberes principales que el líder realiza en su rol de servir al equipo.

1. Protege al equipo de las interrupciones. El líder servidor necesita aislar y proteger a los miembros del equipo de desviaciones, interrupciones, y peticiones de trabajos que no son parte del trabajo.

Cuando los representantes del negocio están estrechamente involucrados en el proyecto, puede ser tentador para ellos el realizar peticiones secundarias de cambios o mejoras directamente a los desarrolladores que desvían el esfuerzo de desarrollo planificado. Mientras que los proyectos ágiles animan este conocimiento del negocio y estas peticiones, deben realizarse a través de los canales apropiados. Los representantes del negocio deben realizar estas peticiones durante la reunión de planificación de la iteración o presentarlas al responsable del proyecto, el cual gestiona la cartera. Parte de ser un líder servidor, implica el recordar a las personas los canales designados para que el equipo pueda mantener su atención en la iteración y establecer una velocidad segura. Estos indicadores de progreso pueden ser utilizados entonces para ayudar a planificar el plan de trabajo futuro.

Aunque es importante proteger al equipo de desviaciones internas, el gestor del proyecto debe ser especialmente vigilante en la protección del equipo de las desviaciones externas. La fragmentación del tiempo – hacer que la gente deje de estar centrada en el trabajo del proyecto y moverlos arriba y abajo entre las iniciativas – socava la productividad. En contraste, proteger al equipo de demandas externas que no sean críticas mejora la productividad. Colocar cerca físicamente a los miembros del equipo es una manera eficiente de prevenir interferencias externas. Si la gente continúa situada en sus antiguos departamentos o lugares de trabajo, es muy fácil que los involucren en trabajos fuera del proyecto.

2. **Elimina los impedimentos para progresar.** Los líderes sirvientes necesitan eliminar los obstáculos del camino del equipo que podrían causar retrasos o la adición de trabajo sin valor. Estos obstáculos pueden incluir trabajo desperdiciado, actividades de conformidad que distraen al equipo de completar los objetivos de la iteración actual. En el vocabulario Lean, el trabajo de conformidad se refiere a esfuerzos que no contribuyen directamente a entregar valor al negocio. Por ejemplo, esto puede incluir tiempo duplicado guardando tareas, reuniones que no sean del proyecto, y otras actividades administrativas.

En las reuniones diarias en las cuales el equipo reporta su progreso, el trabajo planificado, los problemas, el líder necesita tomar nota de los problemas y trabajar para solucionarlos, el mismo día si es posible. Eliminando o facilitando estos impedimentos permitirá al equipo de desarrollo trabajar más rápido y finalmente entregar más valor al negocio.

Algunas herramientas ágiles de gestión de proyectos actuales soportan las carteras de impedimentos. Estas carteras son una especie de listado priorizado de obstáculos a eliminar. Los líderes servidores pueden utilizar estas herramientas para controlar su trabajo de eliminación de impedimentos.

3. **(Re) Comunicar la visión del proyecto.** Puede parecer un deber extraño de colocar en la categoría de líder servidor, pero comunicar y re comunicar la visión del proyecto es crítico para liderar con éxito un equipo. Únicamente si los Stakeholders tienen una visión clara de las metas para el producto completo y el proyecto pueden alinear sus decisiones con ellas y trabajar hacia el objetivo común del proyecto. En su best-seller “The Leadership Challenge”, James Kouzes y Barry Posner dicen que los líderes necesitan revelar un objetivo hacia el cual otros puedan trazar sus cursos. Simplemente, una visión común ayuda a que todo el mundo tire hacia la misma dirección.

Las visiones divergentes se desarrollan normalmente entre miembros de un equipo bien intencionados. En los proyectos de software, por ejemplo, el deseo de un desarrollador por la simplicidad o una nueva tecnología puede ocasionar que su trabajo diverja de los requisitos del usuario. El deseo de un analista especializado en calidad o seguridad por la exhaustividad y la conformidad puede divergir de los requisitos del gestor y los sponsors del proyecto para el progreso y la finalización del proyecto. Comunicar y recomunicar la visión del proyecto ayuda a los Stakeholders a reconocer estas divergencias y ponerlos otra vez en línea con los objetivos del proyecto.

Los líderes más efectivos (líderes de nivel 5) dedican un porcentaje más alto del tiempo de su trabajo a comunicar y recomunicar la visión del proyecto y la corporación que las personas que están en niveles más bajos de liderazgo. Kouzes y Posner creen que es casi imposible para los líderes el sobre comunicar la visión del proyecto y establecen que es un paso crítico para el liderazgo efectivo.

Por lo tanto, los proyectos ágiles no deben tener únicamente un ejercicio de visión en el inicio del proyecto o cuando se desarrollan las metas de la iteración. Este enfoque limitado no es suficiente. En lugar de eso, los líderes servidores necesitan vigilar continuamente por oportunidades para comunicar la visión del proyecto y encontrar nuevas maneras de ilustrar y reforzar la visión.

4. Llevar comida y agua. Este deber no habla literalmente de agua y comida, sino de proveer los recursos esenciales que el equipo necesita para mantenerlo nutrido y productivo. Estos recursos pueden incluir herramientas, compensaciones, y estímulos. Las personas que están alimentadas por la profesionalidad y el deber únicamente no pueden continuar contribuyendo en la medida de sus capacidades, iteración tras iteración. Los líderes necesitan aprender que motiva a los miembros de sus equipos como individuos y encontrar maneras de recompensarlos por el buen trabajo. Como medida simple, un buen lugar para comenzar es un sincero “gracias” a alguien por su duro trabajo.

Los líderes también necesitan celebrar las victorias – las grandes, por supuesto, pero también las pequeñas – tal y como el proyecto progresa. A menudo es tentador reservar las celebraciones para el final del proyecto, pero si los miembros del equipo no reciben un reconocimiento regular, el proyecto puede no llegar nunca a alcanzar un final exitoso. Las celebraciones y el reconocimiento ayudan a dar impulso, y los líderes necesitan nutrir a sus equipos con estas recompensas de manera frecuente para mantener el avance del proyecto de manera productiva.

La formación y otras actividades de desarrollo profesional son ejemplos de recursos que el equipo necesita para ser productivo. El gestor del proyecto debe interesarse en ello y organizar la formación apropiada para cada individuo del equipo. Mediante el desarrollo de las habilidades de los miembros del equipo, el proyecto no solo ganará los beneficios de sus nuevos conocimientos, sino que además mostrará que queremos que los miembros del equipo crezcan como individuos, no únicamente extraerles trabajo e información.

Doce principios para dirigir proyectos ágiles.

Adicionalmente a los cuatro deberes principales que acabamos de ver, hay otras actividades que los líderes servidores deben mantener en mente. Jeffrey Pinto, en “Project Leadership: from Theory to Practice”, ofrece la siguiente lista de principios para líderes a seguir:

1. Aprender las necesidades de los miembros del equipo.
2. Aprender los requisitos del proyecto.
3. Actuar para el bienestar simultáneo del equipo y del proyecto.
4. Crear un ambiente de responsabilidad funcional.
5. Tener una visión del proyecto completo.
6. Utilizar la visión del proyecto para conducir su propio comportamiento.
7. Servir de figura central en el desarrollo exitoso del proyecto.
8. Reconocer los conflictos del proyecto como un paso positivo.
9. Dirigir con un ojo puesto en la ética.
10. Recordar que la ética no es solo una reflexión, sino una parte integral del pensamiento.
11. Tomarse tiempo para reflejarlo en el proyecto.
12. Desarrollar el truco de pensar al revés.

En el punto 12, pensar al revés significa que visualizamos la meta final y entonces trabajamos hacia atrás para determinar que hace falta que pase para conseguirla y que problemas y riesgos pueden suceder.

2.3.4.2. Herramientas y técnicas de liderazgo

Las herramientas y técnicas de liderazgo empleadas en los proyectos ágiles implican tomar un enfoque de blando, en lugar de una estructura de proyecto basada en la dirección y el control. Hay una conocida cita de Warren Bennis que habla de las diferencias entre estos dos enfoques: “La gestión es lograr que las personas hagan lo que se necesita que hagan. El liderazgo es lograr que las personas quieran hacer lo que hace falta hacer”. En lugar de decirle a la gente lo que debe hacer, necesitamos crear un ambiente en el cual las personas quieran hacer lo que es necesario que se haga. La diferencia entre los dos ambientes es como tratar de tirar de una cuerda (cuando la gente quiere hacer lo que es necesario que se haga) que empujar la cuerda (cuando a la gente simplemente se le dice que debe hacer).

Como líderes, podemos ayudar a crear un ambiente de trabajo productivo utilizando prácticas como el modelado, el comportamiento que queremos que el equipo siga, utilizando diferentes herramientas de comunicación para expresar la visión del proyecto, posibilitando que los Stakeholders actúen, y estando dispuestos a desafiar el statu quo. Veamos estas prácticas de liderazgo con más detalle.

Modelando el comportamiento deseado

En el “The Leadership Challenge”, Kouzes y Posner describen un estudio de diez años en el que preguntaron a más de 75.000 personas, “¿Qué valores, rasgos personales, o características buscas o admiras en tu líder?” Los siguientes fueron los valores más respondidos:

- **Honestidad:** Las personas no seguirán a líderes que saben son falsos, ya que haciéndolo socavan sus propios sentimientos de autoestima. Por lo tanto, los líderes deben prestar especial atención a la transparencia y asegurarse de que siguen adelante con lo que dicen que van a hacer. Por lo tanto no debemos esconder nuestros errores – debemos admitirlos abiertamente. Esto no es solo un buen enfoque para nosotros como líderes, sino que también establece un ejemplo de cómo queremos que nuestro equipo opere. No deberíamos hacer cosas como preguntar por estimaciones a las miembros de nuestro equipo y decir que las vamos a doblar antes de presentarlas a la dirección. Estas declaraciones dañan nuestra credibilidad con nuestro equipo y dan una razón para dudar de nuestra integridad. Un mejor enfoque seria explicar el concepto de añadir una contingencia a las estimaciones y discutir como basar esta circunstancia en expectaciones reales de los riesgos que envuelven el proyecto.

- **Visión de futuro:** Las personas esperan que quien los dirige entienda hacia donde se dirigen. Los líderes deben ser capaces de plasmar la imagen para el equipo de manera que todos entiendan cual es el objetivo último.

- **Competente:** Los líderes no necesitan tener las mejores habilidades del equipo, ya que los miembros del equipo suelen estar contentos proporcionando conocimiento especializado cuando es necesario. Pero los líderes deben ser competentes y no avergonzar o ser un estorbo para el equipo.

- **Inspirador:** Las personas quieren que las inspiren en su trabajo, en lugar de sentir cada día una sensación de fatalidad y tristeza. Por lo tanto, los líderes necesitan encontrar maneras de explicar la visión del proyecto y avanzar con genuino entusiasmo y espíritu.

Cuando encarnamos todos estos rasgos como líderes, no solo haremos que la gente nos siga, sino que moldearemos los comportamientos que queremos que los miembros de nuestro equipo sigan. Estaremos, a todos los efectos, dirigiendo con el ejemplo.

Comunicando la visión de proyecto.

Cuando hemos visto los deberes de un líder servidor, hemos hablado sobre la importancia de comunicar y recomunicar la visión del proyecto para mantener a los Stakeholders alineados con los objetivos del proyecto. Un líder puede utilizar una gran variedad de prácticas para conseguir esto, basándose en lo que es más efectivo para un equipo en particular. Por ejemplo, los equipos XP utilizan metáforas, algunos equipos desarrollan mantras, y otros equipos crean tweets de proyecto, en los cuales explican el propósito del proyecto en 140 palabras o menos. Cualquiera que sea el método utilizado, es importante comunicar frecuentemente las metas y objetivos del proyecto para asegurarse de que todos los Stakeholders están al tanto de ellos y alineados con la visión.

Permitiendo que otros actúen.

Para permitir que los miembros de nuestro equipo se sientan seguros tomando decisiones y acciones que hagan avanzar el proyecto de forma productiva, necesitamos fomentar un ambiente de colaboración. Esto implica el fomentar la confianza entre los miembros del equipo y fortaleciendo a los demás compartiendo el poder. También necesitamos crear un ambiente de trabajo seguro en el que las personas no estén preocupadas por preguntar lo que piensan. Las personas aprenden mucho más rápido cuando pueden elevar preguntas sin miedo a represalias o a hacer el ridículo.

Fortalecer a otros compartiendo el poder significa que el gestor del proyecto, ScrumMaster, o líder, no guarda el plan del proyecto o las estimaciones para el mismo. En lugar de eso, el líder se asegura de que la información y el conocimiento se extienden a través del equipo.

Por ejemplo, esto puede implicar cambiar de herramientas de planificación, como los diagramas de Gantt que únicamente una o dos personas pueden actualizar y mantener, a tablas de tareas que el equipo completo puede utilizar. Haciendo esto, la información de planificación y estatus es más accesible al equipo, y el proyecto se beneficia ya que más personas actualizan y optimizan el plan.

Estar dispuesto a desafiar el Status Quo.

Desafiar el status quo quiere decir que buscamos caminos alternativos para cambiar, crecer y mejorar y por lo tanto experimentamos y tomamos riesgos generando constantemente pequeñas victorias y aprendiendo de nuestros errores. Las iteraciones, son un perfecto microcosmos para experimentar. Podemos probar nuevas ideas durante una o dos iteraciones antes de comprometernos con ellas. Si las ideas funcionan, podemos institucionalizarlas. Si no funcionan, no representa una gran pérdida al menos haberlo intentado, y habremos aprendido algo de la experiencia.

Permitir a los Stakeholders sugerir nuevas ideas de mejora y darles la oportunidad de probar estas ideas es una manera de establecer el concepto de que las ideas de todo el mundo tienen valor. No hay nada más desalentador que tener una buena idea y que caiga en oídos sordos. Si esto ocurre, las personas dejarán de sugerir nuevas ideas y no se preocuparán del proyecto. Por lo tanto, mantener a los Stakeholders comprometidos debería darnos ventajas aprovechando las oportunidades que los proyectos ágiles nos presentan para experimentos a pequeña escala, en un ambiente de bajo riesgo.

2.3.5. Ética y responsabilidad profesional

Antes de pasar al resto de capítulos veremos como el código de conducta ética y profesional se aplica a la hora de comprometer a los Stakeholder. Hay cuarto áreas principales de este código – responsabilidad, respeto, justicia y honestidad – que veremos por turno., buscando ejemplos de como se relacionan con los conceptos vistos anteriormente.

Responsabilidad.

Tomar decisiones basadas en los intereses de la compañía. Este aspecto de la responsabilidad profesional afecta a la gestión y negociación de los proveedores. Cuando seleccionamos proveedores y negociamos los contratos, debemos asegurarnos de que ponemos los intereses de la compañía en primer lugar. Esta regla también influye en la facilitación y resolución de conflictos.

Cuando nos enfrentamos a una situación complicada, debemos preguntarnos a nosotros mismos, ¿Estoy tomando esta decisión porque es la opción más fácil para mí o para el equipo de proyecto, o porque es la mejor decisión para la empresa?

Proteger la información propietaria. Éste es otro aspecto de la responsabilidad profesional que afecta a la gestión y negociación de los proveedores. Cuando trabajamos con proveedores externos, ¿Cuánta información podemos compartir? Debemos asegurarnos de que mantenemos la confidencialidad de la información propiedad de la compañía durante todo el proceso.

Reportar comportamiento no ético y violaciones. Este aspecto de la responsabilidad profesional entra en juego durante los conflictos, especialmente si el conflicto ha llegado a grado 4 o 5. Cuando las personas dejan de comportarse correctamente las unas con las otras, tenemos el deber de reportar este comportamiento inaceptable a nuestro departamento de recursos humanos. Si tenemos constancia de que comportamientos perjudiciales están teniendo lugar, pero no actuamos, estaremos ignorando nuestro deber ético.

Respeto.

Mantener una actitud de mutua cooperación. El equipo puede querer realizar las cosas de la manera más fácil, mientras que el cliente puede querer cosas que son técnicamente difíciles de desarrollar. Mantener una actitud de mutua cooperación significa que aceptamos y respetamos estas diferencias y buscamos maneras de resolver estas disputas de forma amigable.

Respetar las diferencias culturales. Respetar las diferencias culturales tiene una fuerte conexión con la globalización, cultura y diversidad en los equipos. Necesitamos respetar las diferencias de los Stakeholders y encontrar prácticas de trabajo que sean de mutuo acuerdo y productivas.

Negociar de buena fe. Cuando se trata con proveedores, no debemos pedir una cita o entrar en una negociación si no tenemos ninguna intención de utilizar este proveedor. Y no deberíamos llevar a los proveedores a hacernos una oferta únicamente para “hacer números” o satisfacer una motivación política.

Hacer frente al conflicto directamente. Tenemos el deber ético de tratar con los conflictos directamente. Esto quiere decir que cuando veamos que un conflicto sube de intensidad, tenemos que intentar rebajar la tensión trabajando directamente con los que lo han iniciado. Esto no quiere decir que siempre tengamos que intervenir en lugar de permitir a los Stakeholders el intentar solucionarlo ellos mismos, pero debemos siempre controlar cuando una situación se sale de control, e intervenir cuando sea apropiado. Un poco de conflicto en el equipo es normal y saludable, especialmente cuando el equipo se está formando, pero cuando las relaciones entre los Stakeholders comienzan a deteriorarse, debemos intervenir rápidamente.

No uses tu posición para influenciar a otros. La persona al mando, tanto si se llama ScrumMaster, líder o gestor del proyecto, tiene un cierto nivel que poder y autoridad en el proyecto, incluso si su rol es servir al equipo. Cuando se realizan las reuniones y sesiones de toma de decisiones participativas, los líderes deben ser cuidadosos y no influir indebidamente a los otros miembros del equipo; es mejor para el proyecto si cada miembro del equipo da su propia perspectiva y opinión, que si simplemente apoyan el punto de vista del líder sobre un problema.

Justicia.

Actúa imparcialmente sin aceptar sobornos. Este concepto parece obvio, pero ¿Dónde dibujamos la línea? ¿Qué pasa si aceptamos una cesta de Navidad de un proveedor? ¿Es correcto realizar una reunión en un restaurante de lujo? Dependiendo del valor de los artículos recibidos, este tipo de cosas son probablemente aceptables. ¿Pero qué hay de eventos fuera del lugar de trabajo que pueda ofrecer un proveedor, como un viaje? Este tipo de regalos extravagantes, son probablemente un problema. Cuando aceptamos cualquier regalo de un proveedor, debemos tener cuidado de estar cumpliendo tanto con la política de aceptación de regalos de nuestra compañía como con el Código de conducta ética y profesional del PMI.

No discriminar. Este punto se ha vuelto cada vez más importante así como ha crecido la globalización de los equipos. Debemos asegurarnos de que ningún prejuicio personal influencia nuestros deberes o nuestro rol como líder servidor.

No utilices tu posición para obtener ganancias personales o de negocio. Este aspecto de la responsabilidad profesional incluye el no pedir a los proveedores o miembros junior del equipo favores para proyectos en especial o contratos exteriores. Y mientras que si es aceptable el construir redes y conexiones mutuas, no sería ético utilizar nuestra posición como managers para conseguir un trabajo para un amigo o familiar.

Honestidad.

Comprender la verdad. Este punto aplica a todas las discusiones, reuniones, e investigaciones de problemas con nuestros Stakeholders. Tenemos la obligación de descubrir y comprender la verdad, incluso si no es lo que queremos escuchar – por ejemplo como cuando a un cliente no le gusta nuestro diseño o cuando vemos que nuestra falta de comunicación es la causa de un problema. Para poder resolver problemas y tomar decisiones de manera efectiva necesitamos tener una comprensión clara y común de los hechos antes de empezar a pensar en ideas para solucionarlos.

Ser veraz en todas las comunicaciones. Cuando comunicamos el avance del proyecto, debemos enfatizar todo el buen trabajo realizado por el equipo, pero si estamos bajo plan, debemos comunicarlo honestamente. La información sobre el progreso en los proyectos ágiles es muy transparente y anima a ser sincero en las comunicaciones; si el cliente solo ha aceptado un 50% de las características, entonces solo hemos avanzado el 50%.

EJERCICIOS

En el Anexo F.3, se encuentra un examen tipo sobre lo tratado en este punto.

2.4. IMPULSAR LAS PRÁCTICAS DE RENDIMIENTO DEL EQUIPO

En este capítulo hablaremos de los equipos. Para el examen – y para tener éxito en los proyectos del mundo real - es necesario entender cómo y cuándo los equipos trabajan mejor, y cómo crear equipos auto organizados y capaces.

La siguiente tabla muestra las herramientas y técnicas y los conocimientos y habilidades asociados con cada práctica utilizada por los equipos ágiles.

Practica	Herramienta / Técnica	Conocimiento / Habilidad (Nivel)
Comprender el rendimiento del equipo	> Liderazgo adaptativo > Inteligencia emocional	> Crear equipo capaces (Nivel 1) > Crear equipos de alto rendimiento (Nivel 2) > Motivación del equipo (Nivel 1)
Prácticas de equipo	> Reunión inicial diaria > Equipo coubicados > Espacio de equipo > Herramientas ágiles	> Formación y tutoría (Nivel 1). > Técnicas de Brainstorming (Nivel 1) > Equipos distribuidos (Nivel 2)

Tabla 13: Herramientas y técnicas y los conocimientos y habilidades asociados con cada práctica utilizada por los equipos ágiles

2.4.1. Comprender el rendimiento del equipo

Los métodos ágiles hablan mucho sobre procesos con su enfoque en conceptos como las iteraciones, carteras y revisiones. Este enfoque es irónico cuando el primer valor en el Manifiesto Ágil es “Los individuos y las iteraciones por encima de los procesos y las herramientas”. ¿Entonces, por qué ocurre esto? La razón principal es que los procesos y herramientas son simplemente más fáciles de describir, clasificar, y proveer de guía que temas más difíciles relacionados con los individuos y las iteraciones. Los individuos y las iteraciones no pueden ser fácilmente referenciados a términos generales ya que las personas varían mucho – en sus habilidades, actitudes, experiencia, perspectiva, cultura, etc.

Los individuos e iteraciones están menos definidos que los procesos y herramientas; en otras palabras, lo que podría ser correcto realizar con un equipo, puede no serlo con otro equipo.

Mientras escuchamos expresiones como “Lo que parece más fácil acaba siendo lo más difícil”, necesitamos reconocer que no se trata de “lo más difícil” sino de lo más importante. Intentar cuantificar el significado entre un buen proceso contra buenas personas, es problemático en sí mismo, pero puede ayudar a enfatizar la necesidad de poner en enfoque en el lado personal de los proyectos, incluso si este enfoque no está en tu área de conocimiento o en la que te encuentres más a gusto. Veremos un ejemplo de esta cuantificación, y como se relaciona con estimaciones realistas de un proyecto.

Estimar proyectos de software es notoriamente difícil debido a todo el riesgo inherente implicado en desarrollar soluciones para nuevos problemas de negocios que tiene altos ratios de cambio. A pesar de esta dificultad, las compañías de software continúan necesitando una manera de ofertar proyectos y estimar sus posibles presupuestos. Como resultado, mucha gente inteligente ha estudiado el problema extensamente para hallar las soluciones.

Una de las más populares soluciones es el modelo de estimación de software llamado COCOMO (del término “Constructive Cost Model”). Este modelo se creó por ingeniería inversa a partir de las aportaciones de miles de proyectos de software completos que tenían un coste exacto conocido. La idea detrás del modelo era evaluar un gran número de proyectos para comprobar si existía una correlación entre las variables de entrada del proyecto y el coste final, y entonces utilizar estos datos como una base para estimar proyectos futuros. El método se probó con éxito; el modelo COCOMO II se encuentra en el corazón de muchos de los sistemas de estimación comercial utilizados hoy en día. La figura 38, muestra los factores de ponderación del método COCOMO II para siete de las variables de entrada, una de las cuales es “los factores personales” (People Factors).

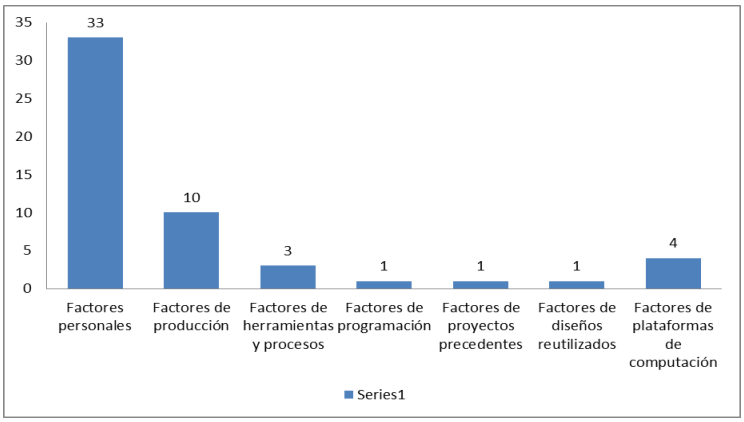


Figura 38: Factores de ponderación para variables de entrada COCOMO. (Fuente: Elaboración propia)

Observando la figura anterior, podemos ver que “People Factors” tiene una puntuación de 33 y que el “Factor de herramientas y procesos” tiene una puntuación de 3. Esto significa que cuando calculamos el coste final de un proyecto de software, el impacto de los factores humanos (como las habilidades y destrezas pobres, medias o altas) es unas 10 veces más significativo que las herramientas y procesos que siguen las personas.

Como se evidencia por estos datos, la idea del Manifiesto Ágil de valorar a los individuos por encima de los procesos es de hecho el camino para el éxito. Buenos trabajadores que tengan pocos o ningún proceso pueden conseguir el éxito incluso en proyectos difíciles, en los cuales equipos con habilidades pobres fallarían, aun teniendo los mejores procesos. Como líderes de los equipos, debemos centrar nuestros esfuerzos en las personas para conseguir el máximo beneficio en el rendimiento. Éste es el motivo por el que el impulsar el rendimiento del equipo es tan importante.

2.4.1.1. Liderazgo adaptativo

Con liderazgo adaptativo nos referimos al concepto de adaptar como dirigimos a un equipo en base a las circunstancias específicas y que el grado de madurez del equipo se encuentra en su formación. La formación del equipo típicamente sigue las fases de Forming, Storming, Norming y Performing. Estas fases son justo lo que parecen – en la fase de Forming, se crea el equipo. Después hay cierta confusión o Storming, mientras la gente aprende a trabajar unida. Después el equipo se normaliza, Norming, empezando a estar a gusto en los diferentes roles y relaciones, antes de pasar a ser un equipo altamente funcional, o Performing, que trabaja junto de manera efectiva. Estas fases, originalmente identificadas por Bruce Tuckman, son seguidas por una fase de desconexión llamada Adjourning o Mourning, ya que las personas a menudo echan de menos estar en un equipo de alto rendimiento después de que éste se disuelva.

En la figura 39 podemos ver las diferentes fases del modelo, mostrando que el equipo puede ir pasando entre Storming, Norming y Performing múltiples veces. Esto es particularmente cierto cuando hay cambios en el equipo del proyecto, como gente que deja o se une al grupo. Los miembros del equipo tienen entonces que restablecer y ordenar los diferentes roles, relaciones y responsabilidades dentro de la nueva estructura.

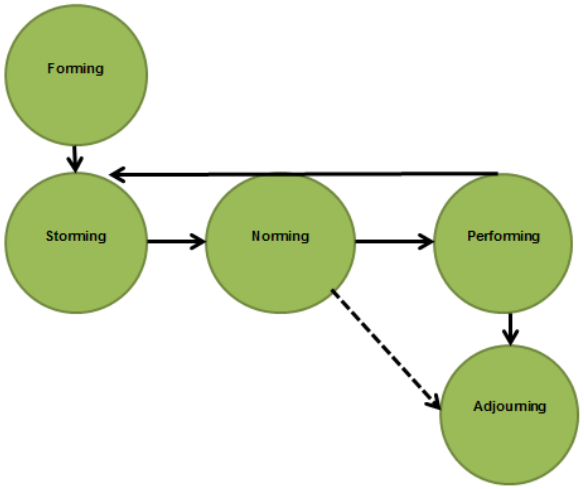


Figura 39: fases de la formación y el desarrollo de un equipo. (Fuente: Elaboración propia)

Otra manera de ilustrar el modelo es desde una perspectiva basada en el rendimiento. Necesitamos llegar a la fase de Performing para obtener el máximo del equipo.

Norming Un equipo potencial que trabajan los unos con los otros y se transforman en un equipo real 3	Storming Pseudo equipo que se desafían unos a otros y se transforman en un potencial equipo 2
4 Performing Un equipo real que trabaja como uno solo y se transforma en un equipo de alto rendimiento	1 Forming Grupo de trabajo que aprenden unos de otros

Figura 40: Fases de la formación y desarrollo de un equipo. (Fuente: Elaboración propia)

En la figura 40, se puede ver que los equipos empiezan en el cuadrante inferior derecho tal y como se van conociendo unos a otros, se mueven a través de la fase de Storming, desafiándose unos a otros y a la fase de Norming, aprendiendo a trabajar los unos con los otros, antes de llegar finalmente a la fase de Performing, en la que trabajan como uno solo. La figura anterior también remarca que inicialmente comenzamos con un grupo de personas en nuestro proyecto, más que con un equipo real.

Durante la fase de Storming, tenemos pseudoequipos, los cuales se transforman en equipos potenciales durante la fase norming, y se convierten en un equipo real en la fase de Performing.

Ahora veremos como este concepto se aplica al liderazgo situacional. Como líderes, podemos ayudar al equipo a través de las fases ajustando nuestro enfoque según la figura 41.

Soporte Miembros del equipo: Media/alta competencia; compromiso variable Líder: Baja dirección; comportamiento de soporte bajo 3	Entrenar Miembros del equipo: Baja/media competencia; bajo compromiso Líder: Alta dirección; comportamiento de soporte alto 2
4 Delegar Miembros del equipo: Alta competencia; alto compromiso Líder: Baja dirección; comportamiento de soporte bajo	1 Dirigir Miembros del equipo: Baja competencia; alto compromiso Líder: Alta dirección; comportamiento de soporte bajo

Figura 41: Modelo de liderazgo situacional. (Fuente: Elaboración propia)

Pronto en la formación de un equipo, el rol de un líder de equipo ágil es ayudar directamente en las actividades del equipo y presentar una imagen clara y táctica de que hace falta hacer. El líder también puede hacer muchas peticiones del tipo “Ayúdame a ver esto” o preguntar cosas como “¿Dónde está el problema?” para ayudar a los miembros del equipo a identificar y articular los problemas.

Cuando el equipo pasa a la fase de Storming, acostumbran a existir muchos desacuerdos, conflictos abiertos, y diálogos ásperos. Durante esta fase, el líder necesita asumir el rol de formador para ayudar a los miembros del equipo a resolver conflictos sin dañar las relaciones. Se debe tener en mente que un poco de conflicto es bueno, por lo que no se debería mimar en exceso al equipo. Hay que dejar que las disputas tengan lugar, pero actuando como árbitro para asegurar que el conflicto no vaya más allá.

Cuando un equipo está en la fase Norming, significa que el equipo ha creado las reglas esenciales para gobernarse a sí mismo. Esta fase no significa que el líder del equipo pueda ponerse en piloto automático no obstante. En lugar de eso, el líder debe desempeñar un rol de Soporte. El equipo continuará necesitando ayuda con la resolución de conflictos, así como recordatorios para hacer cumplir las reglas que se acaban de crear.

Es el momento adecuado para el líder para desafiar al equipo con metas más elevadas como "El equipo es responsable de controlar la velocidad del proyecto", o "Todo el mundo tiene pruebas". Esta fase es también un buen momento para hacer frente a los problemas encontrados en las retrospectivas.

La fase final de Performing no es algo determinado – muchos (sino la mayoría) de los equipos de proyecto nunca alcanzan esta fase ya que las organizaciones hacen demasiados cambios en los equipos, los cuales hacen que los equipos vuelvan a pasar por las fases de Storming y Norming. Los equipos en la fase de Performing son autónomos, capaces, autogestionados, y autocontrolados. Requieren poco más que ser conducidos en la dirección correcta y recibir reconocimiento y valoración por su alto rendimiento. El estilo de delegación de Blanchard y Hersey en esta fase significa que el líder trae trabajo y desafíos al equipo para su resolución.

Parece limpio y ordenado el definir las fases de formación del equipo de esta manera. ¿Quiere esto decir que todos los equipos pasan por estas fases de una manera predecible? No, cada equipo es diferente. Las personas que crean el equipo y factores como si alguno de los miembros del equipo hubieran trabajado juntos en el pasado, afectan a la manera en la que el equipo se desplaza entre las fases. Esto nos lleva a otra pregunta: ¿Pueden los equipos progresar como una unidad completa a través de las fases? No realmente. Las personas y los equipos son complejos y confusos. Lo mejor que podemos hacer es estar al tanto de estos modelos, buscar signos de que el equipo está en una fase concreta, y entonces actuar consecuentemente como líderes. Tales indicaciones generales son útiles, pero no debemos nunca esperar que los equipos actuarán de una manera ordenada y siguiendo las fases tal y como están definidas.

2.4.1.2. Inteligencia emocional

Una de las mejores maneras de permanecer flexible al dirigir o gestionar equipos impredecibles es intentar continuamente mejorar nuestra inteligencia emocional. La inteligencia emocional es nuestra habilidad de identificar, valorar e influenciar las emociones de nosotros mismos y de otros individuos y grupos.

La figura 42 presenta diferentes aspectos de la inteligencia emocional, divididos en cuatro cuadrantes.

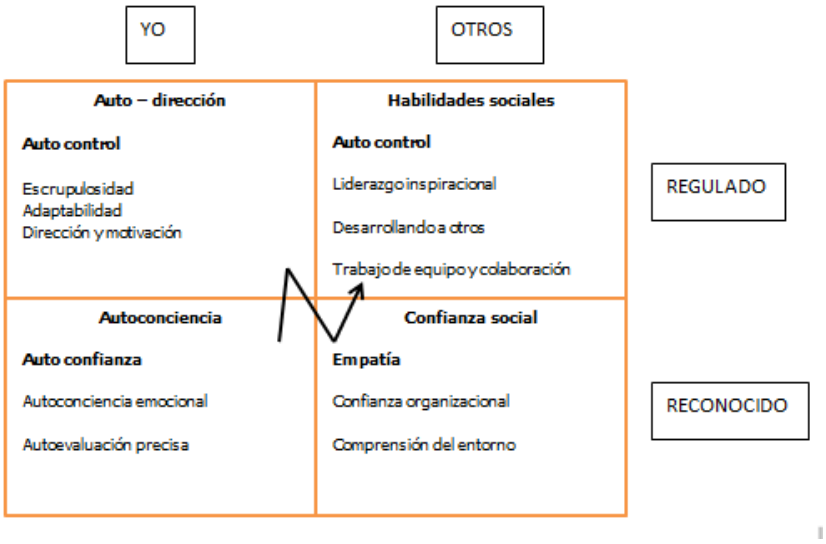


Figura 42: Aspectos de la inteligencia emocional. (Fuente: Elaboración propia)

El diagrama coloca Yo (Self) en el lado izquierdo y Otros (Others) en el lado derecho, y verticalmente se divide en las habilidades de Reconocer (Recognize) y Regular (Regular). Mientras que todo el mundo tiene algún nivel de habilidad en todos los cuadrantes, suele ser lo más fácil comenzar mejorando nuestra inteligencia emocional con nosotros mismos en "Autoconciencia" (Self-Awareness), abajo a la izquierda. Seguidamente podemos aprender como regularnos a nosotros mismos a través de la "Autogestión" (Self- Management), arriba a la izquierda, seguidamente construir Conciencia Social (Social Awareness), abajo a la derecha y finalmente perfeccionar nuestras habilidades sociales (Social Skills), arriba a la derecha.

En otras palabras, primero necesitamos reconocer nuestros propios sentimientos. Una vez que comprendemos nuestras emociones, podemos empezar a controlarlas. Por lo tanto para empezar, reconocemos que es lo que nos hace enfadar, nos frustra, nos hace felices o agradecidos. Entonces necesitamos darnos cuenta de que tenemos el poder de elegir como nos sentimos y responder. Podemos simplemente seguir el patrón normal de estímulo que lleva a la respuesta, pero como humanos, tenemos la habilidad de insertar una decisión entre el estímulo y la respuesta. Tenemos la habilidad de elegir si queremos permitir que algo continúe preocupándonos o si queremos responder a ese algo de manera diferente. Reconocer que tenemos una opción es una de las claves para ser conscientes de nosotros mismos y avanzar en el dominio de la autogestión.

Debemos también tener en mente que, el cómo nos gestionemos a nosotros mismos y nuestra actitud tiene un impacto en los que están a nuestro alrededor, particularmente si estamos en una posición de liderazgo. Como experto en inteligencia emocional, Daniel Goleman explica, "El estado de ánimo y comportamiento del líder rige el estado de ánimo y comportamiento de todos". Un jefe con mal humor y despiadado crea una organización tóxica llena de gente con bajo rendimiento que ignora las oportunidades.

Una vez que nos hemos clasificado a nosotros mismos en las áreas de autoconciencia y auto gestión, debemos trabajar en desarrollar la conciencia social y la empatía de otros. Como formadores o gestores ágiles, debemos ser capaces de identificar cuando los miembros del equipo están atascados, frustrados, o preocupados para poder ayudarlos. Entonces, una vez que somos capaces de reconocer cuando otros necesitan ayuda, utilizamos habilidades sociales, como la habilidad de influenciar, inspirar, dirigir, y desarrollar a otros, para ayudar a los miembros de nuestro equipo a hacer su trabajo y colaborar los unos con los otros.

2.4.1.3. Crear equipos capaces

Los equipos capaces están tanto autoorganizados como autodirigidos. Veamos cada una de estas características por separado.

Equipos autoorganizados.

Que le digan a uno que debe hacer nunca ha sido la receta para conseguir un éxito arrollador. Los que reciben las tareas terminan dudando de la dirección y la secuencia de las actividades entrantes, y los que proveen las instrucciones acaban frustrados con obstáculos imprevistos y problemas técnicos. Empujando las instrucciones como se empuja una cuerda – no es muy efectivo y nunca saca lo mejor de la gente.

En contraste, los miembros de los equipos capaces están liberados de la gestión tipo mando y control y pueden utilizar su propio conocimiento para determinar cómo hacer mejor su trabajo. Los equipos capaces también habilitan a las organizaciones para aprovechar la habilidad natural de las personas para gestionar la complejidad. Gestionamos la complejidad cada día, haciendo malabares con nuestra vida laboral, personal, en los e-mails, llamadas de teléfono y encuentros. No obstante, las empresas a menudo fallan en sacar provecho de esta habilidad cuando se trata de ejecutar tareas de proyectos. En lugar de presentar a los miembros del equipo un número de elementos que debe ser realizado, presentan un conjunto de tareas ordenadas, que en realidad, sería mejor realizar en otro orden. Permitir a los equipos autoorganizarse nos posibilita utilizar las habilidades complejas individuales de gestión que todos poseemos.

Además, los gestores y los programadores no tienen la misma visión técnica interna de la tarea que se está ejecutando, que sí tienen las personas que están desarrollando el trabajo diariamente. Por lo tanto, los gestores y los líderes de los equipos ágiles están realizando una mejor gestión al permitir a los miembros del equipo autoorganizarse su propio trabajo. En lugar de proveer de detalladas listas de tareas, los líderes de los equipos ágiles deben describir las metas de la iteración a alto nivel y dejar que el equipo determine que es lo que sería mejor para cumplir con el trabajo, dentro de las reglas básicas que son aceptables para la empresa.

Este reconocimiento de que el equipo está en la mejor posición para organizar el trabajo del proyecto es liberador y motivador para los miembros del equipo. La gente trabaja más duro y está más orgullosa de su trabajo cuando se les reconoce como expertos en su área. Cuando los equipos autoorganizados seleccionan elementos de la cola de trabajos en espera, tienen la experiencia para elegir los elementos que no se encuentran bloqueados por alguna razón, que son capaces de hacer, y que los llevarán hacia el objetivo de la iteración. Esta práctica aligera muchos de los bloqueos técnicos vistos en los sistemas de empuje en los cuales la lista y secuencia es impuesta al equipo.

Por lo tanto, necesitamos delegar la responsabilidad al equipo para lograr el éxito y permitirles realizar lo que sea necesario para lograr las metas. Éste es el “Servicio a los de abajo” o modelo de liderazgo servicial utilizado por los métodos ágiles. En lugar de un estilo “directivo”, el cual es un enfoque “mando y control” en el cual las instrucciones pasan del gestor del proyecto a los miembros del equipo, los proyectos ágiles toman un enfoque de liderazgo servicial, en el cual el gestor de proyecto o líder escuda al equipo de interrupciones, elimina impedimentos, comunica la visión del proyecto, y provee de apoyo y coraje.

Equipos autodirigidos.

Los equipos autodirigidos son equipos que trabajan de manera colectiva para crear las normas del equipo y toman sus propias decisiones internas. Esto significa que no solo piensan en la mejor manera de cumplir el trabajo que tienen comprometido para la iteración, sino que además resuelven muchos de los problemas del día a día que surgen a lo largo del camino. Los gestores de proyecto o líderes pueden apoyar y reforzar este comportamiento respetando las estimaciones del equipo y sus decisiones y permitiéndoles el cometer errores y solucionarlos.

Esto no significa que el gestor del proyecto abdique la responsabilidad al equipo. En lugar de eso, se le da al equipo libertad dentro de los confines de la iteración. Si la estimación del equipo está alejada del camino a seguir, o si toman malas decisiones técnicas, esto se debe detectar y discutir en la retrospectiva de la iteración. Los problemas deben entonces solucionarse durante la siguiente iteración, y en unas pocas iteraciones, las estimaciones del equipo y sus decisiones técnicas, serán mejores que las que el gestor del proyecto podría haber producido.

Recuerda que los atributos de los equipos autoorganizados y autodirigidos son la meta; no se ha comenzado con ellos. Los equipos, inicialmente necesitan soporte y guía cuando se enfrentan con el ámbito del proyecto, con las herramientas, y con los aspectos Forming y Storming del proyecto. Cuando están en la fase Norming, el gestor del proyecto puede introducir los objetivos de autoorganización y autodirección como objetivos a largo plazo para el grupo.

2.4.1.4. Creando equipo de alto rendimiento

En su libro “The wisdom teams”, Jon Katzenbach y Douglas Smith definen a un equipo como “un pequeño grupo de personas con habilidades complementarias comprometidos con un propósito común, con unas metas de rendimiento y con un enfoque, por los cuales se sostienen mutuamente unos a otros”. Hay algunos aspectos valiosos de esta definición que veremos después. Lo primero a considerar, es que los equipos se describen como algo generalmente “pequeño”. A menudo vemos equipos de 10 o 20 personas. Tal y como crece el tamaño del equipo, es posible que necesite ser dividido en múltiples subequipos e incluir a gente de otros departamentos y divisiones. Manteniendo un equipo pequeño (12 o menos miembros, por ejemplo) permite a los miembros del equipo desarrollar mejor sus relaciones y comunicarse más directamente.

En segundo lugar, los miembros del equipo tienen “habilidades complementarias”. Mientras que los miembros del equipo individualmente pueden no poseer todas las habilidades necesarias para completar una tarea por ellos mismos, el equipo, colectivamente, sí que tiene las habilidades necesarias.

Esto puede significar que el equipo está formado por especialistas cada uno con su rol en el proyecto, pero los métodos ágiles también promueven el uso de especialistas generalizados (personas con habilidades en diferentes campos que puedan rápidamente cambiar de rol). Por ejemplo, un proyecto ágil en un ambiente de creación de software puede tener analistas de negocio los cuales también lleven a cabo tareas de calidad o desarrolladores con habilidades de análisis. Los especialistas generalizados con habilidades cruzadas pueden desarrollar diferentes tareas en los proyectos y pueden ayudar a suavizar los picos y valles en los recursos necesarios.

En tercer lugar, los equipos se definen como que están “comprometidos con un propósito común”. Esto significa que los miembros del equipo están alineados detrás de la meta del proyecto la cual sustituye a sus agendas personales. Los equipos también comparten “metas de rendimiento” y “enfoques” comunes.

En otras palabras, los miembros del equipo están alineados (si no siempre de acuerdo) sobre como se deben medir las metas y sobre como el equipo debe enfrentarse al trabajo. Finalmente, tenemos la idea de que los miembros del equipo "se sostienen mutuamente unos a otros". En otras palabras el equipo comparte el resultado del equipo.

Son numerosas las personas que han investigado como crear estos equipos de alto rendimiento, como por ejemplo Carl Larson y Frank LaFasto, autores del libro Teamwork. Su búsqueda ha tenido influencia en las siguientes directrices para gestores de proyecto:

- **Crear una visión compartida para el equipo:** Haciendo esto se habilita al equipo a tomar decisiones rápidas y a tomar confianza.
- **Establecer metas realistas:** Deberíamos llevar a la gente hasta el éxito, no a que fallen, por lo tanto, las metas deben ser alcanzables.
- **Limitar el tamaño del equipo a 12 personas o menos:** Los equipos pequeños se comunican mejor y pueden apoyar el conocimiento tácito (no escrito).
- **Crear un sentido de identidad de equipo:** Tener una identidad de equipo ayuda a incrementar la lealtad de cada miembro del equipo hacia al equipo y su apoyo a otros miembros del equipo.
- **Proveer de un liderazgo fuerte:** Los líderes deben estar siempre presentes para marcar el camino, y dejar que el mismo equipo cumpla la misión.

Lyssa Adkins ha explorado también el concepto de los equipos de alto rendimiento y ha identificado las siguientes características para este tipo de equipos:

- Son **autoorganizados** en lugar de basarse en los roles o funciones de cada persona.
- Están **capacitados** para tomar decisiones.
- Creen realmente que **como equipo pueden resolver cualquier problema.**
- Están comprometidos con **el éxito del equipo** vs éxito a cualquier coste.
- El equipo **es dueño de sus decisiones y compromisos.**
- **La confianza,** vs el miedo o la ira, los motiva.
- El equipo **se rige por el consenso.** Con divergencia total seguida de convergencia.
- Y viven en un mundo de constantes **desacuerdos constructivos.**

Muchas de estas características se explican por ellas mimas o ya han sido vistas. Veamos los dos últimos puntos en más detalle. El penúltimo, habla de establecer un ambiente seguro en el que los miembros del equipo debatan o discutan sobre los problemas sea visto como algo sano y sea alentado ya que esta práctica lleva en último lugar a mejores decisiones y a que se acepten mejor internamente una vez que han sido tomadas. Divergencia (el discutir y debatir) y convergencia (el acuerdo sobre la mejor decisión) incrementan el compromiso del equipo.

El último punto se relaciona con el penúltimo. El desacuerdo constructivo es vital para entender realmente los problemas y trabajar en ellos. Patrick Lencioni, autor de The five dysfunctions of a team, lista las siguientes disfunciones que dañan y limitan el rendimiento de un equipo.

1. **Ausencia de la verdad:** los miembros del equipo no están dispuestos a ser vulnerables dentro del grupo.

2. **Miedo al conflicto:** el equipo busca una armonía artificial por encima del debate constructivo y apasionado.

3. **Falta de compromiso:** los miembros del equipo no se comprometen con las decisiones del grupo o simplemente fingen estar de acuerdo con ellas.

4. **Evitan la responsabilidad:** los miembros del equipo esquivan la responsabilidad de avisar a sus compañeros de un comportamiento contraproducente.

5. **Descuido de los resultados:** los miembros del equipo priorizan sus necesidades individuales, como el éxito personal, el estatus, o el ego, por encima del éxito del equipo.

Estas disfunciones provienen todas de evitar el conflicto (o desacuerdo constructivo) y de no tener un ambiente seguro en el que esté bien hacer preguntas. Establecer un ambiente seguro para el desacuerdo es la clave del éxito; este tipo de ambiente permite a los miembros del equipo construir un fuerte compromiso con las decisiones. Si poseen este compromiso, cuando encuentren los inviables obstáculos durante el proyecto, serán capaces de superarlos o sortearlos, en lugar de llevar a sus gestores una larga lista de razones por las que algo no puede hacerse.

2.4.1.5. Motivación del equipo

Lo que motiva a las personas va más allá de pagarles un salario. El salario únicamente motiva a la gente a aparecer cada día por el trabajo. Una vez que están en el trabajo los niveles de productividad de los empleados varían desde alguien quien socava el proyecto a alguien quien ofrece innovación apasionada a la empresa.

La motivación del equipo, es el arte de alentar a las personas hacia el lado derecho de la gráfica anterior. Alistair Cockburn ha descrito la motivación de un equipo como el vector de propulsión de los miembros del equipo en una balsa. En otras palabras, si las motivaciones individuales de los miembros del equipo son personales y no están alineadas hacia la meta del proyecto, el vector global del equipo (dirección y velocidad) será menor y no estará bien dirigido hacia la meta del proyecto.

Sí, no obstante, podemos encontrar una manera de alinear las metas personales de los miembros del equipo con la meta del proyecto, haremos que el vector del proyecto sea mayor hacia el éxito de la resolución final del proyecto.

Para conseguir que todos se posicionen en el mismo sentido, necesitamos encontrar y comprender plenamente las motivaciones personales de cada uno de los individuos del equipo y las motivaciones del equipo como grupo, utilizando técnicas como entrevistas individuales. Imaginemos, por ejemplo, que tenemos un miembro del equipo llamado Bob, el cual se encuentra a seis meses de la jubilación y que lo único que quiere es que sus seis últimos meses de trabajo sean lo más fáciles posibles. Tenemos otro miembro del equipo, Tim, que está enojado porque quiere utilizar un tecnología en particular para el sitio web pero el grupo ha elegido otra herramienta que lo impide. Y otro miembro del equipo, Jane, está buscando trabajo porque no le han dado el rol de líder que ella quería.

Una vez que hemos hecho esta búsqueda y hemos encontrado y plenamente entendido que motiva (y desmotiva) a las personas, podemos comenzar a determinar si algunos elementos de lo que ellos quieren, se puede introducir en el plan del proyecto. Por ejemplo, esto puede querer decir, “Bob, sé que te vas a retirar en seis meses pero, ¿no sería una gran manera de ser recordado el ganar el premio a proyecto del año? Tenemos una gran oportunidad de lograrlo”. Para Tim, el motivador podría ser, “Sé que no hemos elegido la tecnología que tu preferías para la nueva página, pero ¿podríamos utilizarla para crear el sistema de ayuda o la base de conocimiento? ¿Te podríamos poner al cargo de esta parte del proyecto?”. Y para Jane, podríamos ofrecer un motivador como, “Jane, ¿Qué tal si te ponemos de líder en las próximas dos iteraciones con una revisión en la retrospectiva? Sé que no es el rol a tiempo completo que deseabas, pero podría ser una buena experiencia y podrías añadirlo en tu currículum”.

Estos ejemplo son sencillos, pero ayudan a ilustrar que muchos proyectos tienen la oportunidad de incorporar las metas personales de los miembros del equipo en las metas del propio proyecto – incluso si es solo temporalmente. Encontrando una manera de alinear las metas personales con las metas del proyecto, las personas ven que hay en él para ellos y sus niveles de motivación y productividad se incrementan sustancialmente.

Las conversaciones francas explicando porque el proyecto es importante para la compañía pueden ayudar también a motivar al equipo. El tener a un miembro de alto rango de la empresa comunicando e inspirando la visión de lo que la compañía espera que el proyecto logre es mejor que las llamadas de teléfono o los e-mails.

Además, dejar que los miembros del equipo sepan que alguien que es significativo para ellos se preocupa de que el proyecto sea un éxito es también un poderoso motivador.

Por último, mientras que es bueno comprender e incorporar los motivadores individuales, nos debemos asegurar de que consolidamos los objetivos de la compañía y del equipo por encima de los objetivos individuales. Es importante hacer cosas como celebrar las victorias como equipo y ofrecer recompensas al equipo completo, en lugar de recompensas individuales. Necesitamos fomentar el aspecto de “responsabilidad mutua” del equipo de trabajo.

2.4.2. Prácticas de equipo

Las prácticas de equipo se refieren a las actividades que el equipo realiza para ser más efectivo. Seguidamente veremos las reuniones diarias, las técnicas de brainstorming, y otras herramientas y técnicas utilizadas para coubicar y distribuir equipos.

2.4.2.1. Reuniones diarias

Las reuniones diarias son una de las prácticas principales en los equipos ágiles. Son reuniones cortas, centradas en algo concreto que niegan la necesidad de más reuniones. Las reuniones diarias están limitadas a 15 minutos o menos y se mantienen en estos tiempos haciendo que los asistentes solo respondan a tres preguntas. Estas reuniones son para el equipo y hechas por el equipo lo que ayuda a mantener a todo el mundo centrado en el ámbito aceptado y en la meta de la iteración.

Las siguientes preguntas son las que debe responder cada miembro del equipo para reportar su estatus:

1. ¿En qué has estado trabajando desde la última reunión?
2. ¿Qué tienes planeado acabar hoy?
3. ¿Existe algún impedimento u obstáculo para tu trabajo?

Cuando las personas reportan problemas, las conversaciones sobre como reparar esos problemas deben tener lugar fuera de línea (p.e. fuera del ambiente de la reunión) para mantener la reunión en marcha y dentro de los 15 minutos de tiempo.

2.4.2.2. Formar y asesorar

Formando y asesorando a los equipos ágiles se les ayuda a mantenerse en la pista del proyecto, superar los problemas, y mejorar continuamente sus habilidades. La formación se realiza simultáneamente en dos niveles, a nivel individual y a nivel del equipo completo.

La formación del equipo completo, ocurre más en los límites de la iteración ya que es cuando tenemos al equipo unido para eventos como la planificación de la iteración, la revisión de la iteración y las retrospectivas. Éstos son buenos momentos para hacer que las prácticas ágiles funcionen bien y ayudar realmente al equipo a incorporar su rol ágil. Los cambios de proceso se realizan entre iteraciones en lugar de en la mitad de la iteración, ya que la iteración es el periodo durante el cual el equipo se dedica a completar la tarea dentro de un ambiente sin cambios.

Durante la iteración, ofrecemos formación y asesoramiento individuales a los miembros del equipo. Esto podría ser el reunirse con un miembro del equipo cara a cara, en un ambiente seguro y confidencial si el miembro tiene problemas o quejas a reportar. Durante estas conversaciones, es importante mantener una conversación sincera, pero siendo positivos y respetuosos. Para resolver problemas, podemos necesitar debatir con otros gestores, establecer comprobaciones frecuentes, y hacer un seguimiento para asegurarse de que los problemas han sido resueltos o como mínimo reducidos.

Adkins remarca las siguientes acciones que ayudan a establecer el trasfondo para la formación y el asesoramiento cara a cara:

- **Reúnelos medio paso por delante:** No intentes empujarlos directamente al punto final. En lugar de eso, entrénalos de manera que se muevan hacia la meta final y den el siguiente paso desde donde están ahora. Como ejemplo, si nuestro objetivo es que los miembros del equipo seleccionen sus propias tareas pero aún no lo hacen, en lugar de decirles que los equipos ágiles deben seleccionar su propio trabajo, intenta llevarlos a medio camino con preguntas como ¿Quién podría hacer esto?

- **Garantiza la seguridad:** Declara a los miembros del equipo que todas las conversaciones se mantendrán confidenciales, y asegúrate de que así es. Las personas están más dispuestas a colaborar y compartir sino tienen miedo a que las cosas se repitan fuera de contexto.

- **Asóciate con los directivos:** A menudo los directivos funcionales miembros del equipo no se encuentran en el equipo del proyecto. Además, estos directivos pueden no estar utilizando o conocer los métodos ágiles, por lo que el cómo se miden los miembros del equipo en sus roles funcionales puede no coincidir con los valores ágiles. Necesitamos asociarnos con los directivos funcionales para alinear las metas de todo el mundo y asegurar que las contribuciones al proyecto de los miembros del equipo se reportan apropiadamente a los directivos funcionales.

- **Crear consideraciones positivas:** Pueden no gustarnos todos los individuos que formamos, pero tenemos que ayudarlos a todos. Si alguien no nos gusta, este sentimiento puede mostrarse a menudo de maneras sutiles. Por lo tanto, es importante desarrollar una compasión real por los demás y un deseo de ayudar a las personas a desarrollar sus roles.

2.4.2.3. Técnicas de Brainstorming

Los equipos ágiles utilizan técnicas de Brainstorming para ayudar a identificar las opciones, resolver los problemas, y encontrar maneras de mejorar los procesos.

La técnica de Brainstorming o lluvia de ideas tiene como objetivo generar la mayor cantidad de ideas posibles en un periodo de tiempo determinado. En este método, los participantes son invitados a pensar ideas rápidamente alrededor de una pregunta, problema u oportunidad.

La única regla para desarrollar una lluvia de ideas son: “ninguna idea es mala”. Esta técnica, que prioriza la cantidad sobre la calidad de las ideas, es usada generalmente como un proceso divergente para estimular la creatividad y la innovación, y posteriormente, se transforma en un proceso convergente al agrupar y evaluar las ideas generadas durante el proceso.

La Lluvia de Ideas, se usa para lograr que un grupo de personas comparta sus ideas acerca de un tema o pregunta particular. Esto podría ayudar a definir una idea, un enfoque o una estrategia. El método también puede ser usado para ayudar al grupo a romper paradigmas, patrones de pensamiento que estén bloqueando el desarrollo de nuevas ideas.

Los pasos para llevar a cabo el Brainstorming son los siguientes:

1. Elegir un coordinador, que apuntará las ideas y será el facilitador.
2. Definir el problema o asunto como un desafío creativo
3. Establecer un tiempo límite por sesión. El tiempo recomendado es de unos 25 minutos.
4. Una vez comienza el Brainstorming, se escriben cada una de las ideas en el menor número de palabras posible. Los participantes dan propuestas y el facilitador será el que tome note en una pizarra en blanco o alguna superficie donde todos las vean. Se deben escribir todas las ideas, a pesar de lo absurdas que parezcan. Se debe fomentar el buen ambiente y las risas y no las críticas. Se establecerá el turno a seguir señalando el participante que comenzará. Cuando se llega a un punto del desarrollo en que el volumen de ideas aportado decrece apreciablemente, se hará una ordenación o una lectura de las ideas aportadas, produciéndose generalmente una segunda fase creativa.
5. Se revisan todas las ideas para verificar su comprensión, solicitando la información que fuera necesaria, y llegar a un consenso sobre los problemas redundantes, no importantes o no negociables. La tormenta de ideas se dará por finalizada cuando ningún participante tenga ideas que aportar.
6. Se revisa la lista de ideas, agrupando las que puedan ir juntas, y eliminando las duplicadas.
7. Se determinarán cinco criterios para juzgar que ideas solucionan mejor el problema o desafío.
8. Se definen cuantas ideas votará cada participante. Debería ser como mínimo 1/3 del total.
9. Cada integrante escribirá las ideas seleccionadas.
10. Se cuentan los votos que ha recibido cada idea.
11. Se eliminan las ideas con un menor número de votos.
12. Se repiten los pasos anteriores con la lista reducida hasta que lleguemos a una lista donde todas las ideas tengan el mismo peso.
13. Una vez obtenida esta lista se discute grupalmente hasta llegar a una decisión final.

Con estas técnicas podremos mejorar la efectividad y rendimiento del equipo, ayudándonos de los mismos miembros del equipo para conseguirlo.

Por ejemplo, el equipo puede pensar:

- Roles del producto que se pueden presentar en las personas
- Los elementos que deberían aparecer en una lista de características mínimas comercializables para su entrega.
- Riesgos potenciales que pueden impactar en el proyecto.
- Soluciones a un problema encontrado en una retrospectiva.

El Brainstorming puede realizarse de diferentes maneras, incluyendo la escritura, una ronda de turnos, o un formato libre.

Escritura

Con este método, los miembros del equipo tienen tiempo de elaborar una lista de ideas individualmente. Este enfoque limita la influencia de los demás, ya que las ideas se generan aisladas antes de que los miembros del equipo las compartan.

Ronda de turnos.

El formato de la ronda de turnos también puede utilizarse. Con este enfoque, todo el mundo en turnos, sugiere sus ideas. Este formato tiene la ventaja de permitir que las ideas se construyan con las de los demás, pero el grupo debe estar cómodo compartiendo sus ideas frente a los demás, para que funcione.

Libertad para todo.

Otro enfoque de Brainstorming es el formato de libertad para todo. Con este método, las personas simplemente gritan sus ideas. Es espontáneo y puede ser colaborativo si los miembros del equipo investigan y mejoran las sugerencias a través de discusiones, pero solo puede funcionar en un ambiente apoyado. Incluso en un ambiente de este tipo, no obstante, los miembros más tímidos pueden no ser escuchados o pueden sentir que no tienen las mismas oportunidades para participar.

Cuando se elige que método de Brainstorming utilizar, es importante comprender la disponibilidad del equipo a colaborar como un grupo. Lyssa Adkins recomienda el modelo de “zona verde, zona roja” como una manera de comprender y diagnosticar cuanto apoyo hay a la colaboración.

Una persona en la zona verde…	Una persona en la zona roja…
Toma responsabilidad en las circunstancias de su vida.	Maldice a otros por las circunstancias de su vida.
No responde a la defensiva.	Se siente amenazado o perjudicado.
No se siente fácilmente amenazado psicológicamente.	Responde a la defensiva.
Intenta crear el éxito mutuo.	Provoca una actitud defensiva en otros.
Busca soluciones en lugar de quejarse.	Es rígido, reactivo y justo.
Utiliza la persuasión en lugar de la fuerza.	Utiliza la vergüenza, la culpa y las acusaciones.
Puede ser firme pero no rígido, sobre sus intereses.	No es consciente del clima de antagonismo que crea.
Piensa tanto en el corto plazo como en el largo.	Tiene poco conocimiento de lo que no es visible.
Se interesa por otros puntos de vista	No busca o valora la retroalimentación
Agradece la retroalimentación	Ve a los demás como el problema o el enemigo
Ve los conflictos como una parte natural de la condición humana.	Ve los conflictos como una batalla y busca ganar a cualquier coste.
Habla calmada y directamente sobre los problemas difíciles.	No deja pasar u olvida las cosas.
Acepta la responsabilidad de sus acciones.	Transmite altos niveles de desaprobación y desprecio.
Continuamente busca profundos niveles de comprensión.	Se centra en las ganancias y ventajas a corto plazo.
Transmite una actitud protectora	Se siente víctima de diferentes puntos de vista.
Busca la excelencia en lugar de la victoria.	Su pensamiento es derecha/izquierda, correcto/incorrecto

Una persona en la zona verde…	Una persona en la zona roja…
Escucha bien	No escucha bien

Tabla 14: Zona verde y zona roja

Los formadores y facilitadores efectivos están en la Zona Verde y demuestran un comportamiento de Zona Verde en todas las iteraciones. También fomentan y solicitan el pensamiento y comportamientos de la Zona Verde a los miembros del equipo. Únicamente cuando las personas pasan la mayoría de su tiempo en el estado mental de Zona Verde pueden llegar a ser realmente útiles en las sesiones de Brainstorming. Es aceptable – y humano – el tener algunos pensamientos de la Zona Roja en algunos momentos, pero debemos mantenerlos a raya y volver a la Zona Verde lo antes posible.

Una vez que las ideas han sido capturadas a través de las sesiones de Brainstorming, los siguientes pasos serían ordenar las ideas, priorizarlas y actuar. Ordenar las ideas se hace habitualmente colocándolas en una pizarra, consolidando las ideas similares, y eliminando las duplicadas, hasta que se logra una lista con diferentes sugerencias.

Los equipos ágiles pueden, entonces, utilizar lo siguiente para priorizar las técnicas:

- **MosCow:** Esta técnica se utiliza para la priorización jerárquica. Designaciones como “1, 2, 3”, o “Alto, medio, bajo”, pueden también utilizarse.

- **Votación por punto o multivotación:** Con esta técnica, la priorización pública o privada de los votos se propaga a través de toda la constelación de elementos.

Veamos estas dos técnicas en más detalle.

MoSCoW

Como vimos en capítulos pasados, el termino MoSCoW proviene del método ágil DSDM y se deriva de las primeras letras de las frases “Must have” (debe tener), “Could have” (podría tener), “Would like to have, but not this time” (Nos gustaría que lo tuviera, pero no esta vez). Cuando utilizamos el MoSCoW para priorizar las ideas del Brainstorming, el equipo categoriza las sugerencias debajo de las cabeceras de las frases que acabamos de ver, hasta que consiguen el consenso en la jerarquía.

Votación por puntos o multi votación

Con esta técnica, la sabiduría del equipo emerge a través de las prioridades individuales. Todo el mundo obtiene un determinado número de puntos (o marcas de verificación, pegatinas…) para distribuir por las opciones presentadas.

Para ilustrar como funciona esto, veamos un ejemplo. Imagina que hemos completado la sesión de Brainstorming para hallar riesgos potenciales para el proyecto. La sesión ha dado como resultado una lista de 40 riesgos únicos que ahora necesitan ser priorizados.

A cada persona se le dan ocho votos en la forma de marcas de verificación que pueden utilizar para indicar que elementos piensan que son los más importantes. Cada miembro del equipo está limitado a ocho marcas, pero depende de cada uno el cómo distribuirlas. Puede ser una única marca en ocho elementos diferentes, cuatro marcas en un solo elemento y dos marcas en dos elementos, o cualquier otra combinación que refleje la evaluación de las prioridades para ese miembro del equipo.

El líder de la sesión de Brainstorming suma entonces los votos para cada elemento y crea una lista ordenada basada en cuantos votos ha recibido el elemento. La votación puede ser pública, o puede ser privada con alguien realizando el recuento para prevenir las luchas y la votación estratégica.

Cuando se decide cuantos votos dar a cada persona, una buena regla de oro es el 20 por ciento del total de elementos a ordenar. Por lo tanto, si hay 40 riesgos a votar, podríamos calcular 40 x 0,2 = 8, y todo el mundo obtendría 8 votos para distribuir.

2.4.2.4. Espacio del equipo

El espacio del equipo es el ambiente designado en el cual los miembros del equipo llevan a cabo su trabajo diario. Como los métodos ágiles recomiendan las interacciones cara a cara como la mejor manera de comunicarse, no es sorprendente que también recomienden equipos coubicados con un área común de trabajo para colaborar y compartir información.

Los equipos ágiles a menudo requisan un espacio abierto como una sala de conferencias para que sirva como su espacio de trabajo. Este tipo de espacios pueden también conocerse como sala de guerra. Cualquiera que sea la manera en la que se llame, debe tener mucho espacio en las paredes para pizarras, que se utilizaran en las discusiones colaborativas, y espacio para publicar información de los indicadores del proyecto.

2.4.2.5. Equipos coubicados

Aunque los métodos ágiles promueven utilizar equipos coubicados creando un espacio de equipo colaborativo, este enfoque también tiene sus desafíos. Veamos algunos aspectos importantes de los equipos coubicados que se deberían conocer para el examen, incluyendo algunos de los desafíos, y maneras de superarlos.

Cuevas y comunes.

El ambiente abierto de trabajo es genial para escuchar información útil sobre el proyecto que puede ayudarnos en nuestro trabajo, pero también son buenos para escuchar conversaciones privadas telefónicas con su cónyugue y directivos financieros.

Para ayudar a resolver este problema, muchas compañías dan acceso a oficinas privadas en las cuales la gente puede ir a realizar llamadas privadas. Estas oficinas pueden también ser utilizadas por los miembros del equipo que quieran trabajar aislados por un corto periodo de tiempo si están teniendo problemas de concentración en un ambiente ruidoso.
Este modelo se llama “cuevas y comunes”. Cuevas, refiriéndose al espacio en el cual los miembros del equipo pueden retirarse cuando necesitan tranquilidad o privacidad para llamadas de teléfono, y común, sería el área en la cual los miembros del equipo trabajan como grupo.

Comunicación osmótica.

Por comunicación osmótica nos referimos a la valiosa información que fluye de los miembros del equipo como parte de las conversaciones y preguntas diarias cuando trabajan cerca unos de otros. Por ejemplo, imagina que Bob pregunta a Jim, un miembro coubicado, como reiniciar el servidor. Mary, otro miembro del equipo, está a punto de utilizar esa máquina y escucha la conversación.
Ella puede intervenir en el reinicio y salvar un conflicto potencial. Esta habilidad de recoger cosas que de otra manera se perderían en el mayor beneficio de los equipo coubicados.
Cuanta más separación tengamos entre los miembros del equipo, más difícil será que fluya esta comunicación osmótica. Por lo tanto, cuando nos enfrentamos al desafío de trabajar con un equipo separado físicamente, los métodos ágiles recomiendan que eliminemos cuantas más barreras posibles para la comunicación osmótica cara a cara.
Alistair Cockburn une la comunicación osmótica a los campos de energía que radian de las personas. Si estás demasiado lejos, recibirás muy poco, pero si estás trabajando en la proximidad, recibirás el beneficio pleno, como vemos en la siguiente figura. En otras palabras, si podemos tener a la gente sentada y trabajando juntas con pocas barreras entre ellos, la comunicación osmótica mejorará.
El conocimiento tácito, el cual se refiere a información que no ha sido escrita pero que se mantiene a través del conocimiento del grupo, es también más efectivo cuando los equipos están coubicados. Por ejemplo, como reiniciar la impresora puede no estar escrito en ninguna parte, pero como todo el mundo está cerca y ha visto o escuchado a otros hacerlo antes, la información se comparte y apoya por el equipo.
Tal y como los equipos crecen en tamaño, se vuelve más difícil físicamente el mantener conversaciones cara a cara con todo el mundo, y el conocimiento tácito empieza a romperse.

Como resultado, el equipo necesita empezar a poner más cosas por escrito para mantener a todo el mundo informado. La documentación escrita es mucho más lenta de crear, por lo que muchos de los ahorros de tiempo provenientes de la colaboración de equipos pequeños, se pierden.

Para facilitar la comunicación cara a cara y mantener el conocimiento tácito, lo métodos ágiles recomiendan limitar el tamaño del equipo a 12 personas o menos. Así como el proyecto crece en tamaño y necesita más recursos, el enfoque preferido es dividir los equipos grandes en subequipos más pequeños. Los representantes de los subequipos necesitan entonces verse cada día para coordinar y sincronizar el trabajo a lo largo del proyecto.

2.4.2.6. Equipos distribuidos

Los equipos distribuidos son aquellos que tienen como mínimo un miembro del equipo trabajando fuera de sitio. La “Version One State of Agile Surveys” de 2010 y 2011, que recibió feedback de miles de proyectos, reportó que aproximadamente el 50 por ciento de los proyectos ágiles tienen como mínimo una persona trabajando fuera del lugar habitual de trabajo. Esto significa que, en lugar de ser la excepción, los equipos distribuidos son la norma. Una razón por la cual los equipos distribuidos se utilizan tanto es que Internet ha abierto nuevas opciones de comunicación y reducido los costes de comunicación, lo que hace que formar equipos distribuidos geográficamente no solo sea posible sino que sea también rentable.

Un gran desafío para los equipos distribuidos es encontrar las maneras de replicar los beneficios de la colaboración cara a cara, la comunicación osmótica, el conocimiento tácito, y las relaciones mejoradas que vienen de personas trabajando en proximidad unas con otras. Afortunadamente, los mismos factores que han hecho de los equipos distribuidos algo común permiten herramientas que pueden ayudarnos a restablecer algún nivel de los beneficios que se obtienen de la colaboración cara a cara. Lo siguiente son ejemplos de estas herramientas:

- **La videoconferencia** puede ser especialmente útil para las reuniones diarias y las retrospectivas para proveer de presencia visual para los miembros del equipo distribuidos.
- **Moderadores de reuniones basados en la web** ayudan a seguir la participación y proveer de un concentrador para conectar la información.
- **Las aplicaciones de encuesta** pueden ser una manera efectiva de votación en tiempo real o fuera de línea, para conseguir respuestas a preguntas y opiniones.
- **La mensajería instantánea (IM) y los auriculares VoIP (Voice over Internet Protocol)** pueden hacer que personas que están en la otra punta del mundo parezca que

están mucho más cerca. Con los auriculares de Skype, por ejemplo, los miembros del equipo pueden hablar los unos con los otros como si estuvieran en la misma habitación.

- **Las aplicaciones basadas en la presencia** extienden las capacidades de la IM gestionando el estado "actualmente online" de los participantes para crear un ambiente de oficina virtual para compartir información. Estas aplicaciones usualmente ofrecen servicios de gestión de documentos y archivos, así como capacidades rudimentarias de planificación de proyectos, para ayudar a los miembros del equipo a colaborar.
- **Las pizarras interactivas** pueden compartir contenidos con múltiples lugares y permitir a los participantes el colaborar en un ambiente visual que es mucho más rico que una conversación telefónica.

Más allá de las herramientas que la tecnología puede ofrecernos, podemos también dar el paso de reunir al equipo al menos una vez al inicio del proyecto. Como discutimos al inicio del capítulo anterior, las comunicaciones son mucho más sencillas una vez que las personas se han conocido cara a cara.

Otro aspecto de los equipos distribuidos a tener en cuenta es que las fases de formación de los equipos de Storming y Norming son más difíciles cuando los miembros del equipo no están coubicados. Algunas personas son más propensas a desvincularse de un email o un debate por VoIP, en lugar de permanecer atentos como lo harían si estuvieran en la misma habitación que la otra persona. No obstante, otras personas parecen más cómodas hablando por email, tal vez porque se sienten protegidos por la distancia y por un sentido de anonimato. En contrate, cuando te reúnes en persona, estos individuos pueden de repente ser agradables y dóciles.

Storming y Norming son críticas para ayudar a los equipos a construir compromiso a las decisiones y resultados. Por lo tanto, los líderes de los equipos ágiles distribuidos necesitan asegurar que hay suficiente debate y creación de decisiones colectivas pronto en el proyecto para poder atravesar estas fases. Esto puede significar el introducir ciertas partes de trabajo antes en el proyecto justamente para mantener al equipo hablando y trabajando de los problemas.

Jean Tabaka, autor de Collaboration Explained, ofrece los siguientes consejos para gestionar equipos distribuidos:

- **Mantener una metáfora:** las metáforas pueden ayudar al esquipo a estar centrado en la misión o visión del proyecto. Por ejemplo, una metáfora para un proyecto sobre construir un sistema de seguridad podría ser "Estamos creando la Gran Muralla de China".
- **Aplicar comunicaciones frecuentes:** Cuando los miembros de un equipo no están cerca unos de otros, programar comunicaciones frecuentes puede ayudar. Por ejemplo, un equipo distribuido puede tener dos reuniones de inicio al día además de llamadas uno a uno programadas para compensar la falta de comunicaciones espontáneas.

- **Intensificar la facilitación:** Esto puede significar realizar más preguntas, repitiendo las respuestas más frecuentemente en las conferencias, y trabajando para mantener a todos comprometidos.

- **Prácticas de colaboración para conferencias:** También es importante mantener las conferencias efectivas y productivas de manera que las personas estén dispuestas a atender y contribuir. Aquí vemos algunas guías básicas a seguir cuando se facilitan las llamadas de conferencia:

  - ✓ Mantener el rumbo: no deberían haber agendas difusas.
  - ✓ Mantenerse dentro de tiempo: Como regla general, mantener las llamadas dentro del límite de una hora.
  - ✓ Mantener el seguimiento de quien está en la llamada: esto puede realizarse manteniendo un control de las ubicaciones.
  - ✓ Mantener el flujo de decisiones: enviar las agendas por adelantado puede ayudar a conseguir esta meta.
  - ✓ Mantener la entrada de ideas: Para realizar esto, necesitamos comprometer a los participantes con preguntas.
  - ✓ Mantener la justicia: Esto significa que necesitamos mantener un control telefónico justo.
  - ✓ Mantenerlo fácil: En otras palabras, no tomar el control de decisiones.
  - ✓ Mantenerlo documentado: No solo debemos documentar la conferencia, también debemos enviar el feedback adecuadamente.

Jim Highsmith señala que existe una importante diferencia entre los proyectos distribuidos y subcontratados. Como él explica, “Los proyectos distribuidos tienen múltiples localizaciones de desarrollo que pueden abarcar edificios, ciudades, o países. Los proyectos subcontratados involucran a múltiples entidades legales, por lo tanto contrataciones, contratos con las administraciones, y hacer frente a diferentes infraestructuras de desarrollo son añadidas a la carga de trabajo del equipo.

Así como crece el proyecto y requiere más recursos, es inevitable que los equipos se vuelvan distribuidos. Simplemente, no es posible acomodar a una gran número de personas en un único espacio.

Pero incluso con equipos distribuidos, Highsmith afirma que los métodos ágiles son mejores para conseguir el éxito que los enfoques no ágiles a causa de los siguientes factores:

- Las iteraciones cortas en el desarrollo ágil fuerzan a una colaboración y coordinación cercanas y continuas.

- El control de los proyectos ágiles distribuidos es mejor ya que en cada iteración se crea un proyecto distribuible.

En otras palabras, si tienes que dirigir un equipo distribuido, es mejor emplear prácticas ágiles, ya que el feedback frecuente es más útil que únicamente añadir documentación para mantener el equipo en marcha.

2.4.2.7. Herramientas ágiles

Además de las herramientas que acabamos de ver las cuales ayudan con las comunicaciones distribuidas, hay otras herramientas ágiles las cuales pueden aprovechar los equipos ágiles. Justamente como los valores del Manifiesto Ágil “Individuos e interacciones sobre procesos y herramientas”, los equipos ágiles tienden a preferir las herramientas sencillas y accesibles por encima de los modelos informáticos sofisticados.

Herramientas sencillas y accesibles.

Una de las razones principales por la que los métodos ágiles valoran este tipo de herramientas es que promueven la comunicación y la colaboración, que es donde se transfiere realmente el conocimiento y el aprendizaje en un proyecto. En contraste, las herramientas sofisticadas pueden producir informes y gráficos muy vistosos pero pierden usuarios debido a su complejidad y a la curva de aprendizaje necesaria para dominarlas.

Esta preferencia por las herramientas sencillas y accesibles no exclusiva de los métodos ágiles. La coordinación de batallas militares es a menudo llevada a cabo manipulando fichas físicas de barcos y tropas, a pesar de los enormes presupuestos de los modelos por ordenador, ya que el desplazar las fichas involucra más a los participantes y lleva a una menor confusión.

De acuerdo con este enfoque, vemos en uso extendido de herramientas manuales como las tablas de tareas, historias de usuario escritas en cartillas de 7,62cm x 12,7 cm, y cartas de juego con números en ellas utilizadas en la estimación de planificación póker.

Herramientas digitales.

Hay también un conjunto de herramientas ágiles y aparatos que pueden ser utilizados con los equipos distribuidos, incluyendo:

- **Software de gestión de proyectos ágiles:** Estas herramientas pueden ayudar con la creación de carteras y la priorización. También producen estadísticas de control de la velocidad y ayuda al equipo a asimilar el proceso de seguir la velocidad.

- **Muros con tarjetas virtuales:** Estos muros virtuales pueden imitar los muros de cartas físicas, pero están accesibles en cualquier sitio para los miembros del equipo con conexión a internet.

- **Tablas inteligentes:** Estas herramientas son geniales para capturar los resultados de las sesiones de diseño sin tener que gastar tiempo extra creando documentos formales.

- **Sitios Wiki, herramientas de gestión de documentos, y sitios web colaborativos:** Estas herramientas, ayudan a los stakeholders a crear, ver, modificar y discutir los elementos del proyecto.

- **Herramientas de prueba automatizadas, herramientas de construcción automatizadas, y señales estilo semáforo:** Estas herramientas son típicamente utilizadas en los proyectos de software para enseñar el estatus de la compilación. Por ejemplo, si la compilación es estable y todas las pruebas funcionan, las herramientas dan una luz verde. Si algún código se prueba y hace fallar la compilación, dan una luz roja.

- **Herramientas CASE:** Estas herramientas fueron populares en los 90s para generar código desde los diseños. Ahora se están utilizando para documentación de ingeniería inversa desde código fuente y modelos de datos. Por lo tanto, en lugar de gastar tiempo valioso en mantener actualizada la documentación del sistema, los equipos pueden utilizar las herramientas CASE cada semana para desarrollar tareas como los nuevos modelos de bases de datos por ingeniería inversa, indicadores y estadísticas de código a partir de la evolución del sistema. Las capacidades de estas herramientas liberan tiempo al desarrollador para centrarse en añadir nuevas características, en lugar de mantener la documentación.

2.4.3. Ética y responsabilidad profesional

Antes de pasar al resto de capítulos veremos como el código de conducta ética y profesional se aplica a impulsar las prácticas de rendimiento del equipo. Hay cuatro áreas principales de este código – responsabilidad, respeto, justicia y honestidad – que veremos por turno., buscando ejemplos de como se relacionan con los conceptos vistos anteriormente.

Responsabilidad.

Tomar decisiones basadas en los intereses de la compañía. Los equipos capaces son muy efectivos, pero continuamos necesitando confrontar las decisiones del equipo contra los valores de los stakeholders. Por ejemplo, el equipo puede recomendar comprar un nuevo servidor para conseguir un tiempo de respuesta más rápido, pero si el servidor es muy caro, comprarlo para el proyecto puede no interesar a la compañía.

Proteger la información propietaria. Los equipos distribuidos hacen un uso extensivo de la colaboración basada en herramientas Web, Pero debemos asegurarnos de que el albergar la información de la compañía en estos servicios ajenos no viola las políticas de protección de datos de la compañía. Antes de utilizar estas herramientas, necesitamos comprender sus modelos de seguridad, así como sus términos y condiciones relacionados con el uso de la información y su venta.

Respeto.

Mantener una actitud de mutua cooperación. Cuando tenemos miembros de equipos distribuidos, debemos asegurarnos muy bien de que “fuera de la vista” no sea igual a “fuera de la mente”. Antes de tomar decisiones, nos debemos preguntar a nosotros mismos si hemos comprometido adecuadamente a los miembros desplazados en el diálogo.

Respetar las diferencias culturales. No todo el mundo puede sentirse cómodo con las sesiones de Brainstorming “libre para todo” en la presencia de compañeros y managers. Debemos imaginar cómo comprometer a los miembros del equipo de manera que respetemos sus diferencias culturales.

Negociar de buena fe. En cualquier momento durante el proyecto, pero especialmente en las primeras fases de formación del equipo, debemos asegurarnos que todo el mundo negocia de forma justa cosas como que enfoque tomar después de la sesión de Brainstorming o que características incluir en una iteración.

Hacer frente al conflicto directamente. El conflicto puede ser bueno y se debe permitir que ocurra, pero el potenciar el rendimiento de los equipos puede ser imposible si los conflictos no se tratan de una manera directa y productiva y en cambio se deja que vuelvan a resurgir a lo largo del proyecto.

No uses tu posición para influenciar a otros. Los líderes deben facilitar, en lugar de dictar. Esto significa que dejaremos debatir los problemas a los miembros del equipo y los capacitaremos para tomar decisiones locales.

Justicia.

Busca y revela los conflictos de interés. Debemos estar siempre alerta de los conflictos de interés entre las partes internas de un proyecto, pero este concepto también aplica cuando se escoge que herramientas se utilizan en nuestros proyectos, particularmente en los sistemas alojados remotamente. ¿Son estas herramientas seguras? ¿Protegerán los datos corporativos y no los venderán a otras partes?

No discriminar. Con los equipos distribuidos, la discriminación puede ser el tratar a los miembros del equipo que están fuera de manera diferente a los locales. Por ejemplo, si organizamos una comida para el equipo local para celebrar un evento, debemos encontrar una manera para reconocer también a nuestros miembros del equipo que están fuera.

Honestidad.

Comprender la verdad. Parte de la inteligencia emocional es comprender nuestras fortalezas y debilidades. Cuando formamos y colaboramos, debemos buscar el comportamiento en “zona roja” en nosotros mismos y en los demás y trabajar para ajustarlo.

Ser veraz en todas las comunicaciones. Los equipos rinden mejor cuando los miembros del equipo reportan el progreso y los problemas de forma abierta y honesta durante las reuniones diarias y otros encuentros. Además, cuando hacemos de tutores, no debemos dejar a un lado los problemas difíciles por miedo a dañar los sentimientos de alguien; en su lugar necesitamos encontrar una manera apropiada de discutir el asunto.

EJERCICIOS

En el Anexo F.4, se encuentra un examen tipo sobre lo tratado en este punto.

2.5. PLANIFICACIÓN ADAPTATIVA

Este capítulo trata de la planificación, pero no de cualquier planificación; los proyectos ágiles llaman planificación adaptativa. Un enfoque adaptativo comprende que la planificación es un proceso en curso y tiene múltiples mecanismos para actualizar el plan. La planificación adaptativa difiere de los enfoques de planificación más estáticos que crea mucha de la planificación anticipada. Los enfoques más estáticos son reactivos, más que proactivos; después de crearse el plan inicial, la planificación únicamente se realiza en respuesta a excepciones al plan y cambios de peticiones.

Como hemos visto antes, los métodos ágiles se basan en el valor. Esto significa que tienen el objetivo de maximizar la entrega de valor de negocio al cliente. Por lo tanto la cartera se prioriza con los elementos de alto valor para el negocio en la parte superior, y las entregas se programan para maximizar el valor de la funcionalidad que se está entregando.

Planificación adaptativa.

Como parte de poner la atención en la entrega de valor, los métodos ágiles también intentan minimizar cualquier añadido de trabajo no valioso. Ya que planificar actividades no añade directamente funcionalidad al negocio, se puede considerar como un gasto. Por lo tanto, si queremos minimizar el gasto, debemos tomar el enfoque más eficiente a la hora de planificar. Desde el punto de vista de la eficiencia, realizar la planificación necesaria para el proyecto una sola vez y no volver a replanificar parece lo mejor. Desafortunadamente, para lo proyectos basados en el conocimiento de sus trabajadores, no es muy efectivo ya que, debido a los altos ratios de cambio de este tipo de proyectos, no es seguro o responsable planificar una sola vez. La única manera de conseguir en éxito en estos proyectos, es planificar y replanificar.

La planificación adaptativa es la aceptación consciente de que los primeros planes pueden ser erróneos; por lo tanto, las actividades de replanificación y adaptación deben ser programadas en el proyecto. La incertidumbre lleva a la necesidad de replanificar. Para ayudar a ilustrar este concepto, intentemos ver un proyecto como un viaje. Si estamos yendo hacia un destino que conocemos bien y a través de un terreno que también conocemos bien, podemos utilizar mapas o unidades GPS para crear planes detallados y seguros para el viaje completo. Nuestro ratio de progreso puede variar respecto a la programación inicial, pero ello se debe a que tenemos revisiones del estatus – para comparar donde estamos y donde pensábamos que íbamos a estar ahora, siempre refiriéndonos al plan inicial. Tomando las variaciones del plan en cuenta, podemos recalcular nuestro consumo de tiempo y la estimación del tiempo de llegada.

Tal vez encontremos un bloqueo en el camino, o una distracción, o tal vez debamos volver atrás al Comité de Dirección para pedir permiso e ir en una dirección diferente, pero estos eventos son excepciones; son desviaciones infrecuentes del plan.

En contraste, la incertidumbre de llevar a cabo un proyecto basado en trabadores con conocimientos totalmente nuevo, es más como un viaje a través de una isla desierta que muy poca gente haya visitado alguna vez, utilizando un mapa del tesoro aproximado como guía. Es muy probable que nadie de nuestro equipo haya estado en la isla antes. No hay funciones de GPS para ayudarnos a planificar nuestro viaje. En lugar de eso, como equipo discutimos la meta a conseguir, nos ponemos de acuerdo en un plan general, y partimos en la dirección que creemos que con más probabilidad será la correcta. Si llegamos a un obstáculo, como un lago o un precipicio, que no está en nuestro mapa, no seguimos ciegamente el plan. En lugar de eso, aceptamos que nuestro plan era erróneo y que es necesario actualizarlo, y adaptamos nuestro viaje basándonos en la nueva información que hemos aprendido.

Y lo mismo ocurre con muchos proyectos ágiles; la naturaleza sin precedentes del trabajo nos lleva a que frecuentemente descubramos problemas y experimentemos altos ratios de cambio. Por lo tanto, debemos afrontar el proyecto con la idea de que tendremos que ir adaptando la planificación.

En esta parte, se discutirán las siguientes áreas:

Practica	Herramienta/técnica	Conocimiento / Habilidad (nivel)
Planificación de conceptos	> Timeboxing > Elaboración progresiva > Proceso de adaptación > Característica mínima comercializable (MMF)	> Análisis basado en el valor (Nivel 2) > Descomposición y priorización basada en el valor (Nivel 1) > Juegos ágiles (Nivel 3)
Estimación	> Wideband Delphi y planificación póker > Tiempo ideal > Tamaño relativo / punto de historia > Estimación de afinidad	> Estimación de tiempo, presupuesto y coste (nivel 1) > Principios de contabilidad de proyectos ágiles (Nivel 3)
Planes Ágiles	> Planificación de la iteración y la entrega.	> Cartas ágiles (Nivel 2) > Caso de desarrollo de negocio (Nivel 2)

Tabla 15: Áreas a desarrollar

2.5.1. Planificación de conceptos

La planificación de conceptos incorporada por los métodos ágiles promueve y acepta que los detalles aparecerán así como el proyecto progrese, que es necesario adaptar el proyecto basándonos en el feedback, y que la repriorización frecuente es la norma. Esta aceptación de que la cosas cambiarán y de que mejor aceptar la incertidumbre que resistirse al cambio es lo que gobierna las prácticas ágiles que veremos en este capítulo.

El examen PMI-ACP probará el conocimiento y la habilidad para aplicar los siguientes conceptos de planificación:

1. Planificar a múltiples niveles

2. Comprometer al equipo y al cliente en la planificación.

3. Gestionar las expectaciones demostrando frecuentemente el progreso y extrapolando la velocidad.

4. Adaptar los procesos a las características del proyecto.

5. Actualizar el plan basándonos en las prioridades del proyecto.

6. Asegurar que las estimaciones tienen en cuenta los riesgos, distracciones y la disponibilidad del equipo.

7. Utilizar los rangos de estimación apropiados para reflejar el nivel de incertidumbre en la estimación.

8. Basar las proyecciones en los ratios de finalización.

9. Factores de distracciones y trabajo fuera del proyecto.

2.5.1.1. Timeboxing

Timeboxes son periodos de tiempo cortos y fijos en los cuales las actividades o el trabajo se llevan a cabo. Si el trabajo planificado para la Timebox no está completado cuando el tiempo para ésta se acaba, detenemos lo que estemos realizando y recolocamos el trabajo no completado en una nueva Timebox. Ejemplos de Timebox incluyen:

- Reuniones diarias que tiene un Timebox de 15 minutos.
- Iteraciones que tienen un Timebox de (típicamente) dos semanas.

Veamos un poco más el concepto de Timebox para una iteración. Imaginemos que planificamos completar 10 historias de usuario dentro del Timebox de una iteración, pero solo tenemos 8 completadas cuando finaliza el tiempo de Timebox. Aunque no hayamos completado todas las historias de usuario, no extenderemos el tiempo de Timebox. En lugar de eso, reportaremos que 8 elementos han sido completados y devolveremos los 2 restantes a la cartera para considerarlas en la siguiente iteración.

Los proyectos ágiles a menudo tienen mucha incertidumbre cuando el dominio del proyecto es nuevo para la organización o cuando se utilizan nuevas tecnologías en el proyecto. El Timebox, ayuda a incorporar cierto nivel de orden y consistencia a lo que de otro modo sería un ambiente de trabajo altamente variable. Ofrecen una oportunidad de evaluar los resultados, obtener feedback, y controlar los costes y riesgos asociados a un esfuerzo.

El concepto de Timebox ha sido referido como “el control en el caos”. Proporcionan puntos de control frecuentes para calibrar el progreso y replanificar el enfoque. Otra manera de ver las iteraciones con Timebox es imaginar una caja física que representa la capacidad del equipo para cumplir un trabajo.

El trabajo se carga en la Timebox por orden de prioridad. Si algo no cabe en la caja – en otras palabras, si el equipo no tiene la capacidad de completar una pieza dentro de la iteración – tendrá que esperar a las siguientes iteraciones.

Una punta de arquitectura, es un periodo de tiempo dedicado a probar un concepto. La punta de arquitectura es otra actividad que a menudo también se incluye en un Timebox en los proyectos ágiles. En un ejemplo de un proyecto de software, esto significaría “Vamos a destinar una semana para comprobar el rendimiento de la base de datos nativa antes de tomar una decisión para la conectividad”.

Las Timebox, son también poderosas herramientas de motivación para completar el trabajo. Por ejemplo, cuando trabajamos individualmente, muchas personas utilizan los cronómetros Pomodoro (cronómetros de 25 minutos de duración, a menudo con forma de tomate), para ayudarse a sí mismos a mantenerse centrados en la tarea para periodos cortos de tiempo. El ajuste de 25 minutos permite dos sesiones por hora, con unos pocos minutos para ver emails, tomar una bebida, estirar las piernas, etc.

La razón de que la técnica Pomodoro sea efectiva es que muchas personas se distraen y realizan la multitarea de forma ineficiente. Para muchas actividades, en lugar de responder cada email o mensaje tal y como llega, es mejor trabajar duro durante 25 minutos sin distracciones o interrupciones, y entonces tomar un pequeño descanso.

2.5.1.2. Elaboración progresiva

Elaboración progresiva es el nombre que se le da al proceso de añadir más detalle así como la información aparece. Utilizamos la elaboración progresiva para evolucionar y crear más precisión en:

- Planes
- Estimaciones
- Evaluación de riesgos
- Definiciones de requisitos
- Diseños arquitecturales
- Criterios de aceptación
- Escenarios de test

Elaboración progresiva en planes y estimaciones.

En el inicio de un proyecto, necesitamos planificar y estimar el trabajo involucrado para determinar como de grande será el esfuerzo a realizar y para crear una estrategia y plan de ejecución razonables. No obstante, necesitamos también comprender que al inicio de un proyecto es cuando conocemos menos del esfuerzo a realizar. En este primer punto, aún no hemos podido aprender de lo realizado en el proyecto. Sería imprudente limitar nuestras actividades de planificación y estimación al inicio del proyecto. En lugar de eso, debemos refinar continuamente nuestros planes y estimaciones así como el proyecto progresa y nuevos detalles aparecen. Este proceso de actualizar continuamente es la esencia de la elaboración progresiva.

El concepto de elaboración progresiva se ilustra en la figura 43, la flecha con “ahora”, indica que estamos al inicio del proyecto y estamos creando los planes iniciales. Las líneas en las primeras iteraciones muestran como de detallados están los planes para estas iteraciones. Como puede verse, la primera iteración tiene muchas líneas, pero cada vez hay menos y menos detalle para las siguientes iteraciones.

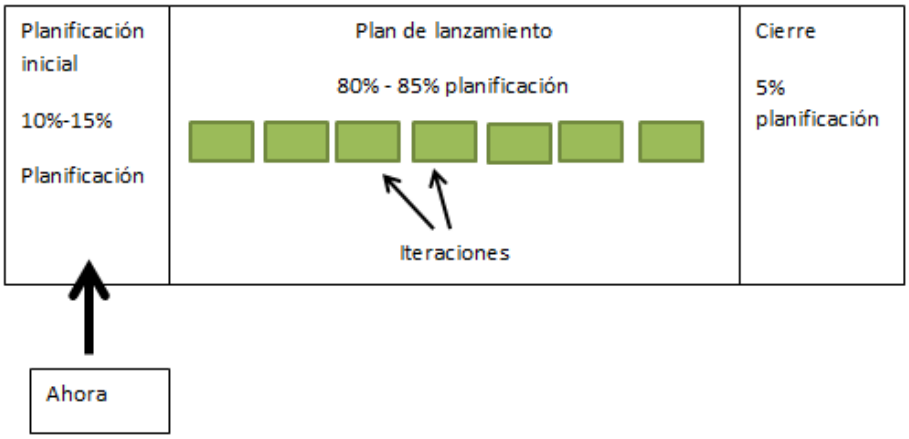


Figura 43: El nivel de planificación inicial en un proyecto cuando se utiliza elaboración progresiva. (Fuente: Elaboración propia)

La figura 44 muestra que, así como el proyecto avanza y nueva información surge, los planes se elaboran progresivamente y se añaden detalles. En este diagrama, la flecha con el “Now” indica que nos encontramos en la tercera iteración. La cuarta iteración se encuentra planificada en detalle, mientras que la quinta, sexta y séptima iteraciones tienen niveles decrecientes de detalle. Este proceso de refinar los planes así como nos acercamos a la iteración continúa a lo largo del proyecto.

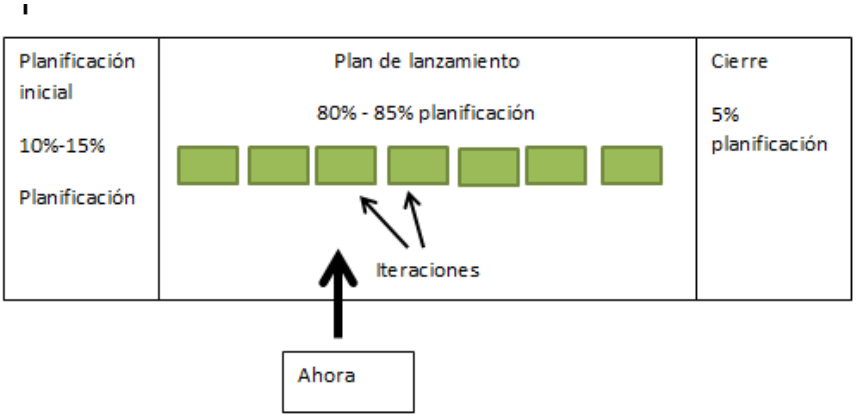


Figura 44: El nivel de planificación así como progresa el proyecto. (Fuente: Elaboración propia)

2.5.1.3. Proceso de adaptación

Así como nuestros planes evolucionan tal y como aprendemos más del proyecto, su ambiente y sus stakeholders, también lo hace nuestro enfoque. Las revisiones retrospectivas son el principal desencadenante que lleva a los procesos de cambios.

Al final de cada iteración, nos reunimos con el equipo extendido (nuestro equipo de desarrollo y otros stakeholders del proyecto) para plantear las siguientes preguntas:

- ¿Qué está yendo bien?
- ¿Qué áreas necesitan mejora?
- ¿Qué se debería estar haciendo diferente?

Como los problemas están identificados, comprometemos al equipo para buscar soluciones. Entonces nos comprometemos a probar las soluciones seleccionadas durante una o dos iteraciones antes de reunirnos otra vez para discutir si la situación ha mejorado. Si las acciones ayudan, las adoptamos como parte de nuestros procesos en el proyecto. Si no han ayudado, consideramos el esfuerzo como una oportunidad de aprender y decidimos si probamos algo más o volvemos al proceso inicial.

A través de este ciclo de inspección regular, reflexión y adaptación, adaptamos los procesos del proyecto a una única situación del proyecto y la organización.

2.5.1.4. Característica mínima comercializable (MMF)

Cuando planificamos una entrega de características a un cliente, la entrega tiene que tener sentido, ser útil, y ser valiosa. Esto aplica a todo tipo de proyectos ágiles, si se trata de una entrega de software, un nuevo producto eléctrico, o un incremento de ingeniería. El término “característica mínima comercializable” (MMF) se refiere a este paquete de funcionalidad que es lo suficientemente completo para ser útil a los usuarios o al mercado, pero que no representa al proyecto completo.

Para un teléfono móvil, por ejemplo, la característica mínima comercializable podría ser un teléfono que pueda ser utilizado para hacer y recibir llamadas, guardar los nombres y números, y acceder al buzón de voz, pero no tenga una cámara, conectividad a internet, o un reproductor de música en su primera entrega. En su lugar, este kit de funcionalidades puede ser añadido en las siguientes entregas y evaluado independientemente.

Hay que tener en cuenta, no obstante, que la funcionalidad del teléfono que se ha entregado en la MMF necesita ser completa. Por lo tanto, todos los atributos para realizar llamadas de teléfono deben estar presentes como parte del MMF para permitir al cliente o al negocio el revisar de manera comprensiva esta funcionalidad.

En el desarrollo de software, puede ser posible transferir anticipadamente incrementos del producto final a la comunidad de usuarios, para que el negocio pueda comenzar a obtener beneficios de la aplicación antes de que se complete el proyecto completo. Por ejemplo, si el sistema de entrada de pedidos y de facturación están completos, pero los enlaces de marketing y gestión de reportes no han sido construidos aún, la empresa puede obtener beneficios entregando esta versión temprana del sistema. La entrega incremental puede permitir algún retorno de la inversión mientras que el equipo desarrolla el resto de funcionalidades. Ésta también provee de una oportunidad para probar la funcionalidad de los sistemas implementados. Estos ensayos de campo, pueden dar como resultado un cambio de los requisitos que pueden ser incorporados en el producto final mientras que el equipo de desarrollo continúa en su sitio.

2.5.1.5. Análisis basado en el valor

El análisis basado en el valor es el proceso de considerar el valor de negocio de los elementos funcionales y actuar consecuentemente. Esto afecta al ciclo de vida completo de los proyectos ágiles y es integral en el sentido de que el valor del negocio impacta en como determinamos el ámbito del proyecto, planificamos, programamos, desarrollamos, probamos, y entregamos trabajos. En cada fase del proyecto, nos preguntamos, “¿Cuál es valor para el negocio de este elemento o práctica?” y “¿Qué elementos de este conjunto tienen el mayor valor para el negocio?” Entonces priorizamos el trabajo para entregar los elementos de mayor valor los primeros.

Para comprender completamente el valor de un elemento, también necesitamos comprender el coste de desarrollo y entrega. Por ejemplo, una característica que está valorada en $5.000 para el negocio pero que tiene un coste de desarrollo de $4.000 (o incluso $6.000) no es tan valioso como algo que entregue un valor de $3.000 al negocio y únicamente cueste $1.000 de desarrollar.

Por lo tanto necesitamos factorizar los costes probables de desarrollo cuando se realiza el análisis basado en el valor. Ésta es una razón por la que estimamos los elementos de la cartera que estarán a alto nivel al inicio del proyecto. Hacer esto nos permite hacer comparaciones beneficio contra coste. Esta información es entonces utilizada para priorizar elementos para su desarrollo.

Por ejemplo, para maximizar el valor proporcionado en una entrega, un gestor de proyectos puede elegir algunas características de valor medio que el equipo puede desarrollar rápidamente en lugar de elementos de alto valor que consumirán todo el tiempo de desarrollo restante.

El otro componente que debemos considerar es la frecuencia de recuperación de la inversión. Podemos analizar si la característica genera valor para el negocio cada semana o mes (como un ahorro de tiempo para el personal) o solo una vez (como un control de conformidad). Por lo tanto, cuando evaluamos el retorno sobre un periodo de uno a cinco años, la visión extendida del retorno debería ser utilizada para evaluar el verdadero valor del negocio.

Teniendo esto en mente, analizamos el valor real para el negocio de las características, y por lo tanto entonces podemos priorizarlas apropiadamente. No obstante, necesitamos recordar que algunos elementos de alto valor para el negocio pueden ser dependientes de algunos elementos de bajo valor, y que por lo tanto estos elementos de bajo valor deben ser desarrollados antes, para permitir al equipo entregar los elementos de alto valor lo antes posible.

2.5.1.6. Descomposición y priorización basada en el valor

La descomposición y priorización basada en el valor es el proceso de obtener requisitos de los stakeholders, clasificar estos requisitos, y entonces colocar los requisitos priorizados en el proceso de desarrollo. Este proceso se ilustra en la figura 45.

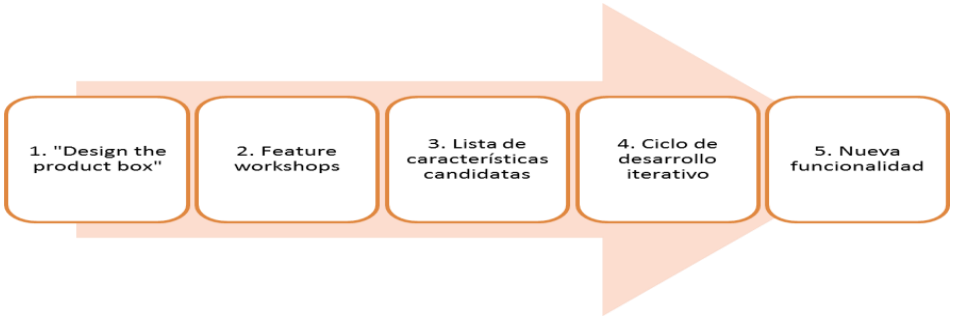


Figura 45: descomposición y priorización basada en el valor. (Fuente: Elaboración propia)

La forma en la cual el proyecto se inicia puede variar de un proyecto a otro, pero independientemente del enfoque que tomemos, debe haber un esfuerzo inicial para definir la visión. En la figura anterior, veíamos una actividad llamada “Design the Product Box” en el inicio del proyecto. Esta actividad captura la visión a alto nivel del proyecto y ayuda a alinear a los stakeholders detrás de una misión, objetivo y criterio de éxito comunes. En el paso 2, tenemos una serie de talleres de características en los cuales la visión del proyecto se divide en características potenciales del sistema.

Estos talleres resultan en una lista de características candidatas (paso 3), la cual es entonces priorizada basándose en el valor del negocio y en el riesgo y conforma la lista priorizada de características para el ciclo de desarrollo iterativo (paso 4).

En este nivel macro, podemos ver ejemplos de descomposición y priorización basada en el valor. Las tres cosas principales que el proyecto debe entregar están capturadas en la caja de visión (el límite de tres es intencionado, para mantener el equipo centrado en priorizar las características y a realizar operaciones). Los principales elementos funcionales de la visión son dibujados en los talleres de características. Las características son entonces priorizadas y consumidas por el ciclo de desarrollo. Así como nos metemos en el desarrollo e intentamos implementar las características, es probable que descubramos la necesidad de apoyar características y otros elementos que previamente no habían sido considerados. El proceso de identificar requisitos adicionales, agrupándolos o separándolos en elementos funcionales, y priorizando estos elementos se repite una y otra vez a un nivel cada vez más detallado y refinado, similar al patrón de una hoja de helecho.

Este es otro ejemplo de cómo los métodos ágiles abordan lo que la guía PMBOK llama “elaboración progresiva”. En la descomposición y priorización basada en el valor, empujamos elementos del proyecto – en cualquier nivel de detalle y tamaño que sea apropiado para la fase del proyecto en la que nos encontremos y cuanto sepamos actualmente sobre el proyecto – en un formato con el que podamos trabajar para el siguiente proceso. Continuamos, entonces, refinando y elaborando estos elementos, añadiendo niveles de detalle más alto y transformándolos en un formato que puede ser refinado más aún en los siguientes procesos. El producto final de la elaboración progresiva es un entregable altamente detallado que todavía se asemeja a los objetivos iniciales del diseño.

Con los métodos ágiles, no tratamos de especificar requisitos totalmente detallados de inicio. En lugar de eso, inicialmente tratamos unos requisitos a groso modo, y progresivamente los refinamos así como el proceso progresa. Este enfoque tiene una serie de ventajas:

- Ayuda a mantener el diseño global balanceado de manera que el producto no se desequilibre por un sobredesarrollo en un área particular.
- Retrasa decisiones o detalles de implementación hasta el último momento viable. Esto significa que no nos estamos precipitando en desarrollar cosas que tal vez más tarde necesitarán ser modificadas como resultado de nueva información o peticiones tardías de cambio.

Como nota aparte, ésta es una de las múltiples paradojas o decisiones balanceadas involucradas en los proyectos ágiles. Nuestro objetivo es tomar decisiones tarde para incorporar cambios, mientras que construimos nuevos incrementos del sistema pronto para obtener feedback, el cual puede resultar en cambios en el sistema. Puede parecer tonto, pero este enfoque es realmente una manera de mitigar riesgos y encontrar problemas y cambios dentro del ambiente amigable del proyecto, en el cual los cambios son menos costosos, que en los ambientes más hostiles de producción, en los cuales son mucho más costosos de realizar.

2.5.1.7. Juegos ágiles (juegos colaborativos y de innovación)

Los juegos de innovación, también conocidos como juegos colaborativos, son técnicas de facilitación de talleres que los equipos ágiles utilizan para ayudar a los stakeholders a entender mejor problemas complejos o ambiguos y alcanzar consenso en una solución acordada.

Lo siguiente son ejemplos de juegos colaborativos utilizados en proyectos ágiles:

- **Recuerda el futuro:** Éste es un ejercicio de posicionamiento de la visión y obtención de requisitos.
- **Podar el árbol de producto:** Este ejercicio ayuda a obtener y dar forma a los requisitos.
- **Lancha rápida:** El propósito de este ejercicio es identificar amenazas y oportunidades (riesgos) para el proyecto.
- **Comprar una característica:** Éste es un ejercicio de priorización.
- **Bang-for-the-Buck:** Este ejercicio busca el valor contra los costes clasificados.

Vamos a ver ahora los juegos más utilizados en detalle, empezando con el ejercicio de “Recuerda el futuro”.

Recuerda el futuro.

Este ejercicio de facilitación compromete a los stakeholders del proyecto en imaginar que la entrega del proyecto o iteración se ha completado. Entonces describen que imaginan que ha ocurrido para que la iteración o entrega haya resultado exitosa.

COMO FUNCIONA. Para ilustrar esto, utilizaremos el ejemplo de planificar una entrega a seis meses. Reunimos a los stakeholders del proyecto, incluido el equipo, usuarios, y sponsors, todos a la vez y les pedimos que se imaginen que estamos a seis meses y dos semanas de la fecha actual. La razón por la que pedimos que imaginen dos semanas más de los seis meses es porque es el tiempo que normalmente tarda en aceptarse e implementarse el producto.

El ejercicio comienza con cada stakeholder trabajando independientemente. Durante los siguientes 20 minutos, el trabajo de cada persona consiste en listar que se completó en la entrega para que fuera exitosa. Una manera de explicar esta parte del ejercicio es pedir a las personas que imaginen que están escribiendo un informe para su jefe o departamento sobre como ha ido la entrega, en el cual listen todas las cosas que se completaron y entregaron. Los stakeholders deben grabar los elementos completos y entregados en notas, con un elemento en cada nota.

Una vez que terminan los 20 minutos, todo el mudo pone sus notas en un muro. Entonces, como un equipo, los stakeholders trabajan juntos para agrupar las notas en grupos asociados y eliminan cualquier duplicado. Este proceso puede llevar otros 20 minutos, mientras la gente clarifica el significado de sus notas y crean cabeceras para identificar correctamente cada grupo de elementos.

TEORIA DETRÁS DEL JUEGO. El juego se ha diseñado de esta manera a causa de las investigaciones realizadas en numerosos estudios de psicología cognitiva. Cuando se formula la pregunta abierta de “¿Qué debería un producto o sistema realizar? las personas se esfuerzan en generar una lista de características completas y medidas provisionales. No obstante, si variamos la pregunta solo un poco y pedimos a las personas que imaginen que se encuentran en un punto posterior a la fecha de entrega y que “recuerden” todo lo que el proyecto o sistema ha realizado para acabar siendo exitoso, obtenemos resultados significativamente diferentes. A causa de que el evento se encuentra ahora “en el pasado”, los participantes deben generar mentalmente una secuencia de eventos que lleven a este resultado. Este ejercicio lleva a mejores definiciones y descripciones más detalladas de los pasos intermedios.

Naturalmente, el propósito del juego de “Recuerda el futuro” no es realmente predecir el futuro. En lugar de eso, estamos intentando comprender mejor las definiciones de éxito de los stakeholders y como podemos alcanzar ese éxito.

Podar el árbol de producto.

Este ejercicio implica a los participantes en pensar ideas sobre las características y funcionalidades de un producto.

COMO FUNCIONA. Para este juego, comenzamos dibujando un gran árbol con tronco y ramas en una pizarra en blanco. La habilidad artística no es un problema aquí – solo estamos creando un lugar en el que colocar las características. Invitamos entonces a los participantes a añadir las características como hojas del árbol de producto. Una buena manera de explicar esta parte del ejercicio es decir que el árbol es el producto, en el cual el tronco representa lo que conocemos ahora o hemos ya construido, y las ramas representan las nuevas funcionalidades que aún deben ser diseñadas.

Animamos a los stakeholders a agrupar las características relacionadas cerca las unas de las otras. Las características de soporte deben estar cerca del tronco, y las características que dependan de ellas deberían colocarse más lejos o en la parte superior del árbol.

TEORIA DETRÁS DEL JUEGO. Pensando sobre como las características se relacionan las unas con las otras y con la funcionalidad ya existente, los stakeholders pueden entender mejor el proceso de establecer prioridades y de definir las secuencias de desarrollo. Este ejercicio ayuda a los participantes en el proceso de la elaboración progresiva. Animamos a los stakeholders a añadir nuevas notas al árbol identificando las características adicionales. Entonces expandimos en más detalle el producto tal y como las características se dividen en historias de usuario y aparecen otros requisitos de soporte.

Lancha rápida.

Una vez que las características y las historias de usuario han sido identificadas en nuestro árbol de producto, podemos vernos tentados de pasar directamente a la tareas de priorización y programación, pero hay un paso muy importante que debe ocurrir antes de esto. Necesitamos identificar y planificar para mitigar (reducir) o evitar problemas potenciales para el proyecto y tomar ventaja de las oportunidades potenciales. Este paso debe ocurrir antes de que prioricemos las historias de usuario ya que muchas de las respuestas a los riesgos pueden necesitar incluirse en el proceso de priorización. Si dejamos la mitigación del riesgo hasta después de haber acabado la priorización de las historias de usuario, nos podemos encontrar intentando introducir tareas importantes en iteraciones ya completamente planificadas, lo cual no funcionará.

El juego de la “Lancha rápida”, que también se conoce como “Barco de vela”, utiliza las características e historias de usuario identificadas en el juego de “Podar el Árbol de producto”. Se centra en obtener los riesgos del proyecto – tanto los riesgos negativos, que serán amenazas para el proyecto, como los positivos, que serán oportunidades potenciales.

Este ejercicio es muy rápido de preparar, y normalmente se obtiene una buena lista de riesgos para el proyecto.

COMO FUNCIONA. Para empezar, colocamos una pizarra en blanco al lado del diagrama de Árbol de Producto del ejercicio anterior. Dibujamos entonces una línea de agua y una figura de un barco, con el barco encarado en la dirección del árbol de producto. Explicamos a los participantes que el barco representa el proyecto. Por ejemplo, podemos decir "Aquí está el proyecto en dirección a la meta que hemos desarrollado. ¿Cuáles son las anclas (o amenazas) que pueden hacer que vayamos más lentamente o incluso nos paremos? Y ¿Cuáles son los otros factores (u oportunidades) que nos impulsarían y nos ayudarían a impulsarnos hacia nuestra meta?"

Los stakeholders, trabajan entonces como grupo creando notas "ancla" para las amenazas e impedimentos para el proyecto. Las notas "ancla" se posicionan debajo de la línea de agua. Los stakeholders, crean también notas "viento" para las oportunidades, que posicionan por encima de la línea de agua. Para distinguir aún más las amenazas del proyecto de las oportunidades, los equipos utilizan a menudo notas de color rojo o amarillo para las amenazas y notas azules o blancas para las oportunidades.

TEORIA DETRÁS DEL JUEGO. Este juego provee de un beneficio más allá de identificar las amenazas y oportunidades para el proyecto. Algunas personas necesitan una manera de articular sus preocupaciones antes de sentirse bien comprometiéndose a trabajar. Una vez que sus preocupaciones han sido registradas y reconocidas, se encuentran menos lastrados por ellas y más felices de contribuir. Han expresado su parte y llevado a cabo su deber de identificar el riesgo, por lo que pueden avanzar con mayor efectividad que si se hubieran guardado sus preocupaciones.

Esto no quiere decir que simplemente registrando los riesgos las amenazas vayan a desaparecer y las oportunidades vayan a ocurrir. Al registrar los riegos es justamente cuando comienza el duro trabajo de reducir y evitar las amenazas o de llevar a cabo las oportunidades de mejora. Todavía tenemos que llevar a cabo el esfuerzo de determinar cómo responder a los riesgos, evaluar esas acciones, y construir la respuesta en el proyecto. No obstante, este juego ayuda a las personas a eliminar obstáculos mentales por lo que pueden avanzar más felizmente, conociendo que sus preocupaciones están siendo consideradas y gestionadas.

2.5.2. Estimación

La estimación es un tema muy largo. Para el examen, serás avaluado en tu conocimiento de la teoría de la estimación ágil y tu habilidad para realizar técnicas sencillas de estimación. Estimar proyectos basados en trabajadores con conocimientos es especialmente difícil, ya que estos proyectos son complejos y, a menudo, nuevos para la empresa. Esto significa que la empresa puede no haber llevado a cabo proyectos similares con anterioridad, o que la tecnología que se utiliza puede ser nueva, y que por lo tanto es probable que aparezcan ciertas incógnitas. Esta combinación de complejidad e incertidumbre hace más problemático estimar proyectos basados en trabajadores con conocimiento que otro tipo de proyectos.

Lo siguiente son algunos puntos generales y buenas prácticas para estimación ágil que se deben tener en mente, tanto para el examen como para proyectos reales:

- **¿Por qué estimamos?** La estimación es necesaria para dimensionar y aprobar proyectos, calcular el ROI y la TIR, y determinar que partes del trabajo pueden ser realizadas dentro de la entrega o iteración.

- **¿Cómo se crean las estimaciones?** Las estimaciones se crean progresando a través de las fases de determinación del tamaño del proyecto, del esfuerzo, de la programación y finalmente el coste. Para crear una estimación global, se debe incluir también el desarrollo del proyecto, despliegue y costes de sustentación.

- **¿Cómo se establecen las estimaciones?** Utilizamos el término estimación y no predicción ya que hay cierto grado de incertidumbre en una estimación. Por lo tanto, las estimaciones se deben establecer como rangos (p.e. “$4.000 a $4.500” o “16 a 18 meses”) para gestionar las expectaciones sobre la incertidumbre del proyecto.

- **¿Cuándo estimaremos?** No deberíamos reservar la estimación para cuando conozcamos menos del proyecto – al inicio. Las estimaciones iniciales son ciertamente necesarias, pero son las menos precisas. Esto significa que necesitamos estimar de manera continua durante el proyecto, introduciendo los costes actuales o las duraciones hasta la fecha para crear mejores estimaciones para que el proyecto avance.

- **¿Quién estima?** Igual que en el caso anterior, no debemos reservar la estimación para la persona que menos conozca sobre la ejecución o aceptación del proyecto – el gestor del proyecto. En lugar de eso, debemos involucrar a los miembros del equipo que están realizando el trabajo en el proceso de estimación. Después de todo, ellos son los que mejor conocen la tecnología, y su involucración incrementará su aceptación de las estimaciones.

Veamos ahora las técnicas utilizadas para la creación de las estimaciones.

2.5.2.1. Wideband Delphi y planificación póker

Wideband Delphi es un enfoque de estimación basada en el grupo. Esta técnica pide a un grupo de expertos que representen estimaciones anónimamente para que nadie sepa a quien pertenece cada estimación. El enfoque anónimo produce estimaciones mejores ya que minimiza tanto que las personas tiendan a estar de acuerdo con el punto de vista predominante como el efecto “halo” en el cual las personas están de acuerdo con las opiniones de los expertos o superiores, en lugar de juzgar sus propias ideas.

Una sesión de estimación Wideband comienza con un esfuerzo de planificación para definir el problema. En lugar de estimar el proyecto completo en una reunión, el grupo rompe el proyecto el problema mayor en partes más manejables. El equipo crea una especificación del problema, identifica los supuestos y limitaciones, y resalta el proceso para siguientes rondas de estimación. Por ejemplo, detalles como si las personas deberían omitir tiempo de documentación de las estimaciones, unidades de estimación (p.e. semanas persona, horas persona, dólares persona, etc), y el criterio de salida (p.e, queremos obtener el +/- 20 por ciento de tolerancia en el rango de estimación) están incluidas en el plan, y se programa una reunión inicial para el equipo de estimadores.

Antes de comenzar a crear estimaciones, los participantes leen la especificación del problema y tienen la oportunidad para plantear y discutir preguntas de cualificación. También reciben hojas de papel con espacios en los que pueden introducir sus estimaciones para diferentes tareas. El facilitador recoge las estimaciones y las representa en un gráfico, sin identificar a quien pertenece cada estimación.

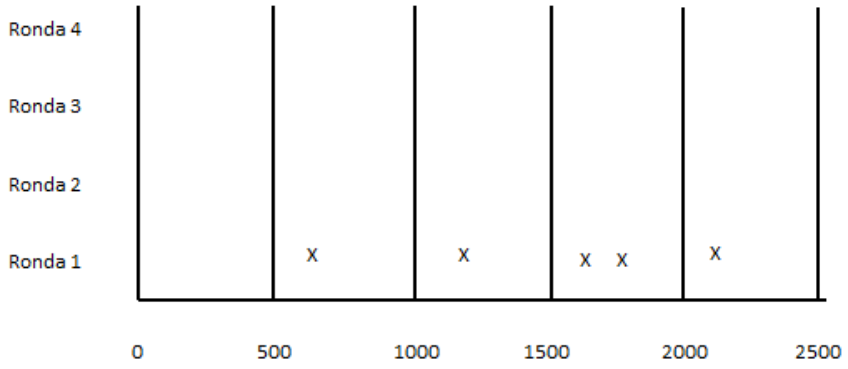


Figura 46: Técnica Wideband Delphi – Estimaciones de la ronda 1. (Fuente: Elaboración propia)

En la figura 46, podemos ver que después de la primera ronda del proceso de estimación, la estimación va de aproximadamente 600 horas por persona a 2.100 horas por persona. No importa si la estimación de 600 horas persona viene del gestor del proyecto o si la estimación de 2.100 horas provienen del desarrollador más experimentado. Toda esta información se mantiene anónima.

Los participantes discuten entonces las diferentes tareas y cualquier supuesto u otro factor significativo que influya en sus estimaciones antes de repetir el proceso de estimación. Por ejemplo, Bill podría informar “He añadido dos semanas para el testeo de la regresión de las aplicaciones, ya que estamos anexando la tabla de contabilidad; la última vez que lo hicimos, rompió el sistema de facturación, por lo que necesitamos permitir cierto tiempo adicional de testeo y posible resolución.” Una vez que todas las tareas, supuestos, y factores significantes han sido discutidos, el grupo repite el proceso de estimación anónima.

Después de varias rondas de este proceso, normalmente empezamos a ver más consenso en las estimaciones, como se muestra en la figura 47.

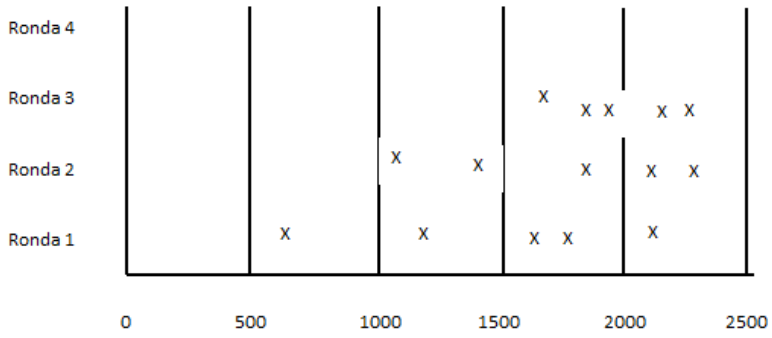


Figura 47: Técnica Wideband Delphi – Varias rondas. (Fuente: Elaboración propia)

En este gráfico, vemos como en cada ronda de estimación, ha habido una deriva de los puntos hacia el este. Esta deriva, demuestra el consenso de que el problema particular o parte del proyecto que está siendo estimado, requerirá un rango más alto de horas por persona. Una vez que las estimaciones han convergido lo suficiente para que el grupo alcance su criterio de salida (en este ejemplo, la estimación más alta y la más baja deben estar dentro de una rango del +/- 20 por ciento de la estimación media), el proceso se para. Una única lista de tareas maestra se crea entonces a partir de la lista de tareas de todo el mundo. Si alguna tarea se excluye de la estimación (como la documentación en este caso), se añade a la lista maestra. El equipo revisa los resultados para asegurarse de que todo el mundo está de acuerdo con la lista final de tareas y con el rango de estimación.

La estimación Wideband Delphi es una buena técnica para ser utilizada por equipos ágiles ya que es:

- **Iterativa:** El proceso se repite varias veces.
- **Adaptativa:** Basándose en el feedback de los otros participantes, los miembros del equipo tienen una oportunidad de actualizar y mejorar la siguiente ronda de estimaciones.
- **Colaborativa:** Es un proceso de equipo basado en la colaboración que mejora la aceptación de los resultados por parte de los participantes.

Planificación Poker.

La técnica de Wideband Delphi se implementa a menudo como “planificación póker”. Esta variación de la técnica combina todos los elementos esenciales del wideband Delphi en un proceso rápido y colaborativo. La planificación póker utiliza cartas numeradas. Los números, que aparecen es estas cartas, que a menudo están basados en la secuencia de Fibonacci, representan unidades dimensionadas, como días de desarrollador o puntos de historia.

Un conjunto de cartas, se entrega a cada participante de la planificación.

Una vez que las cartas han sido distribuidas, un moderador (a menudo el propietario del producto o cliente) lee una historia de usuario, que el equipo discute entonces de manera breve antes de que cada estimador seleccione una carta para representar su estimación de la historia de usuario. Los participantes dan la vuelta a sus cartas, todos a la vez de manera que todo el mundo pueda ver los números. Por ejemplo, imagina que el moderador lee, “Crear una funcionalidad de cambio de contraseña”. Una vez que el equipo discute la historia de usuario, alguien cuenta hasta tres y todo el mundo revela la carta que cree mejor representa el esfuerzo necesario para desarrollar la historia de usuario.

Si hay un grupo de cuatro personas, y tres de ellas dan la vuelta a una carta con el número 5 y la otra da la vuelta a una carta con el número 3, la tarea se registra como 5. Ya que el rango es pequeño y hay un pequeño debate sobre la estimación, el proceso pasa a la siguiente historia de usuario para mantener el juego en marcha rápido. Sí, no obstante, hubieran tres cartas con el número 5 y una con el número 13, se debería discutir el valor extremo.

El proceso entonces se repite – todo el mundo recoge sus cartas y vuelve a estimar, teniendo en cuenta la nueva información. Haciendo esto, veremos emerger un nuevo consenso alrededor del número 13.

Como la técnica de la votación de puño o cinco dedos, vista antes, la planificación póker es un ejemplo de modelo de decisión participativo. Así como se incrementa la involucración de los stakeholders en el proceso, también se incrementa su participación en los resultados.

La planificación póker, es fiel al enfoque del wideband Delphi en que combina los siguientes elementos:

- Tiene múltiples iteraciones de estimación, donde se necesitan.
- Requiere que los participantes presenten estimaciones al mismo tiempo para ayudar a contrarrestar el efecto de que se unan a las estimaciones de sus superiores o el efecto halo.
- Permite la convergencia del grupo en las estimaciones apoyadas.

Como añadido, estudios en Motorola y Microsoft muestran que muchos equipos de proyecto encuentran la planificación póker, no solo más precisa que los enfoques de estimación previos, sino también más rápidos y entretenidos. Dicho esto, el objetivo del ejercicio no es crear estimaciones precisas. En lugar de eso, esta técnica ayuda al equipo a conseguir de forma rápida y barata consenso alrededor de estimaciones razonables para hacer avanzar el proyecto.

2.5.2.2. Tiempo ideal

Cuando pedimos a un equipo que estime un trabajo, el tema de cuál es la mejor manera de incorporar las interrupciones, distracciones, y el trabajo ajeno al proyecto, normalmente aparece. Estas actividades fuera del proyecto pueden incluir el atender a las reuniones de la plantilla, revisar emails, o la visita ocasional al médico. Por ejemplo, para los miembros asalariados del equipo con una jornada de 40 horas semanales, ¿podríamos estimar 35 horas semanales de trabajo en el proyecto y permitir 5 horas para estas otras actividades? ¿Y qué hay del personal externo? Tienen menos reuniones y viene incluso cuando están enfermos, ¿podemos contar 38 horas para ellos?

Dejando los estereotipos aparte, podemos simplificar el debate hablando del tiempo ideal. Esto significa que pediremos a los miembros del equipo que estimen considerando que no habrá interrupciones. En una jornada de 8 horas, pedimos a los miembros del equipo que asuman que las ocho horas estarán disponibles para el trabajo. Obviamente, éste no es un enfoque muy realista, pero el propósito de hablar de tiempo ideal es simplificar el proceso de estimación y sacar la variable de disponibilidad de la ecuación. Haciendo esto, podemos obtener un sentido más preciso del esfuerzo involucrado en el trabajo.

Por lo tanto, el tiempo ideal es cuanto tardara en llevarse a cabo algo, si todo el trabajo periférico y las distracciones son eliminados. Se asume que la historia de usuario que está siendo estimada es en lo único en lo que se está trabajando, que no habrá interrupciones, y que tenemos todo lo necesario, significando que no estamos esperando a que alguien entregue un trabajo o entregue información.

2.5.2.3. Tamaño relativo / punto de historia

El tamaño relativo y los puntos de historia ayudan a resolver dos problemas comunes con la estimación:

1. Las personas no son muy buenas prediciendo el tamaño absoluto del trabajo.

2. El proceso de estimación es difícil e impopular.

Frecuentemente, en los proyectos, tareas que parecen triviales, llevan mucho más tiempo para ser completadas de lo que se creía, y a veces aparecen nuevas tareas imprevistas. El proceso completo lleva más de lo esperado. Se puede intentar introducir estos problemas en todas las estimaciones que se realicen, pero se corre el riesgo de recibir críticas por inflar las estimaciones. Esta, aparente, situación sin salida es la causa debido a la cual realizar estimaciones es altamente impopular entre los miembros del equipo.

¿Por lo tanto, como debemos contrarrestar esta frustración? Resulta que mientras las personas no son muy precisas realizando estimaciones absolutas, son mejores (o al menos se sienten más a gusto con ello) realizando estimaciones comparativas. La diferencia entre estimaciones absolutas y comparativas puede ilustrarse con el ejemplo de dar direcciones.

Podemos indicar como se va al supermercado diciendo que debes viajar 1,3 millas al sur (una aproximación absoluta al dar la dirección), pero es más fácil tanto estimar como de comprender si damos unas explicaciones de cómo llegar más precisas, sigue recto durante 9 manzanas hasta el parque y una vez allí el supermercado está cinco manzanas más allá.

Al igual que estimar una distancia a través de manzanas en lugar de en millas, si tenemos partes conocidas de trabajo ya realizadas, podemos estimar nuevas piezas de trabajo de manera más rápida y precisa teniendo en cuenta las entidades conocidas. Utilizando la creación de un sistema de software como ejemplo, imaginemos que hemos desarrollado una ventana de entrada simple y le hemos dado a la tarea la puntuación relativa de 2 puntos de historia. Podemos entonces estimar otras tareas en referencia a la puntuación de la pantalla de entrada. Por lo tanto, podríamos asignar 1 punto de historia a una reparación simple o un cambio a una pantalla, considerando que solo será necesario la mitad del trabajo que hacía falta para desarrollar esa misma ventana. Podríamos también estimar el desarrollo de otras ventanas simple con 2 puntos de historia, y las piezas de trabajo mayores con 3 – 5 puntos de historia. En este ejemplo “2 puntos de historia” es el equivalente a la “manzana” que veíamos antes, y nos sirve como la referencia que podemos utilizar para estimar otras cosas.

Naturalmente, tomar este enfoque comparativo, no evitará que pasen cosas extrañas o que las actividades no sobrepasen el tiempo que se planificó para ellas inicialmente, pero el cambiar horas por puntos de historia, hace que esto sea más fácil de aceptar. En lugar de decir que somos malos estimando, reconocemos que un par de historias llevaron más tiempo del que pensábamos. Las estimaciones no están basadas en nuestro propio sentido del tiempo; están basadas en como de bien comprendemos como esas actividades se comparan con las que ya han sido completadas en el proyecto.

Otra ventaja de estimar en puntos de historia en lugar de estimar en horas o días, es que eliminamos la techo artificial de “horas por semana”. ¿Si un miembro del equipo completa historias de usuario que suman en total 40 horas de tiempo estimado antes del mediodía del viernes, qué incentivo tendrá para pasar a la siguiente historia de usuario y trabajar en ella el viernes por la tarde? Las estimaciones por horas y días, están demasiado cercanas a nuestra semana de trabajo y a nuestro sentido del deber; esta conexión puede traer problemas de productividad. Por lo tanto, en lugar de eso decimos “La pasada semana el equipo entregó 42 puntos de historia, y esta semana han entregado 45”. Es diferente a decir que el equipo realizó 120 horas de trabajo y que éste ha realizado 130 horas.

El término utilizado para las medidas relativas de trabajo no importa.

La idea es ir más allá de la estimación en horas, la cual no solo crea techos artificiales sino que además crea problemas cuando ciertas personas reciben otras peticiones en las que destinar su tiempo, y no pueden, de ninguna manera, entregar 40 horas de trabajo productivo durante la semana.

En su libro "User Stories Applied: For Agile Software Development", Mike Cohn afirma que el mejor enfoque para estimar historias de usuario es uno que:

- Nos permita cambiar de idea siempre que obtengamos nueva información sobre una historia.
- Funcione tanto para las historias de usuario grandes y pequeñas.
- No nos lleve mucho tiempo.
- Nos provea de información útil sobre nuestro progreso y sobre el trabajo restante.
- Sea tolerante a las imprecisiones en la estimación.
- Pueda ser utilizado para planificar entregas.

Además, deberíamos tener en cuenta los siguientes puntos a la hora de estimar historias de usuario:

- **La definición de punto de historia debe ser de los equipos:** La dimensión de punto de historia, utilizada en el proyecto debe ser creada y debe pertenecer al equipo. Por ejemplo, el equipo puede decidir que 1 punto de historia equivale al esfuerzo involucrado en la creación de una nueva pantalla simple, o que 1 punto de historia equivale a la cantidad de trabajo realizada en un día ideal (sin interrupciones para reuniones, emails, etc). Cualquiera que sea la unidad de tamaño que el equipo escoja, debe ser la que se utilice. Aceptando las decisiones del equipo sobre la unidad de estimación, reforzamos su posesión de las estimaciones. Y no importa si el punto de historia de otro equipo representa una semana ideal de un desarrollador, en lugar de un día. La unidad no tiene por qué ser consistente a lo largo de la organización; únicamente necesita ser utilizada de forma consistente dentro del proyecto. No deberíamos intentar comparar velocidades entre equipos, ya que la simple composición y definición del punto de historia de cada equipo, hace que esa comparación carezca de sentido.
- **Las estimaciones de puntos de historia deben incluirlo todo:** No deberíamos necesitar añadir tiempo al proyecto por unidad de testeo o refactorización. En su lugar, la estimación del punto de historia debe incluir todas las actividades conocidas. De otra manera acabaríamos intentando hacer caber tareas extra o recurriendo a multiplicar la estimación inicial por un factor preseleccionado (p.e. de 1,5) para tener en cuenta trabajo adicional. Los estimadores por puntos de historia inclusivos, son preferibles a los "multiplicadores mágicos", ya que son más precisos y transparentes.

En contraste, los multiplicadores pueden ser aplicados erróneamente multitud de veces u olvidados, y son difíciles de defender.

- **Los tamaños deben ser relativos:** Una historia de 2 puntos de usuario debe ser equivalente más o menos a dos veces el esfuerzo de trabajo necesario para una historia de 1 punto. Una historia de 3 puntos debe ser equivalente a 3 veces el esfuerzo necesario para una historia de 1 punto y también equivalente a la combinación de una historia de 1 punto y otra de 2. Aunque esto parezca obvio, merece la pena recalcarlo. En algunas escalas, los números pueden indicar el posicionamiento en un rango de bajo a medio y a alto, o de pequeño a medio y a grade o enorme. Pero con una escala relativa, la unidad base necesita ser matemáticamente equivalente de manera que podamos utilizar estas unidades para calcular la velocidad, comparar las iteraciones, y validar nuestro plan de entrega. Por lo tanto, completar cuatro historias de usuario de 5 puntos debe ser equivalente a completar veinte historias de usuario de 1 punto.

- **Se debe tener en cuenta la complejidad, esfuerzo de trabajo, y riesgo:** El tiempo total requerido para completar un trabajo es una función de la complejidad de ese trabajo (¿Necesita análisis el trabajo o es posible que hayan sorpresas?), del esfuerzo involucrado (¿tenemos que añadir un único campo de datos o 30?), y de su riesgo (si no podemos completar el trabajo utilizando el plan A, debemos repensar esta parte del proyecto). Necesitamos asegurar que estos tres atributos son evaluados cuando el equipo estima las historias de usuario.

La secuencia de Fibonacci

Las cartas de planificación póker y los puntos de historia son a menudo estimados utilizando la secuencia de Fibonacci, o variantes de esta secuencia. Los primeros números de la secuencia de Fibonacci son “0, 1, 1, 2, 3, 5, 8, 13, 21…” Esta secuencia se consigue añadiendo la suma de los dos números previos para conseguir el siguiente número de la secuencia. Por lo tanto, 0+1 =1; 1+1 = 2, 1+2 = 3; 2 + 3 = 5, y así seguidamente. Los equipos utilizan la secuencia de Fibonacci para estimar el tamaño. Es una secuencia que sucede naturalmente y aparece frecuentemente en conexión con cómo las cosas crecen. Las poblaciones de conejos, las conchas y las ramas de los árboles siguen la frecuencia de Fibonacci, como hacen los problemas de dimensionamiento y el esfuerzo de resolverlos.

2.5.2.4. Estimación afín

La estimación afín es el proceso de agrupar requisitos en categorías o colecciones. Para los proyectos ágiles, esta técnica se utiliza a menudo para agrupar historias de usuario de tamaño similar. La estimación afín es una forma de triangulación; provee de una vista comparativa de las estimaciones y da al equipo la oportunidad de realizar un chequeo real. Posicionando las historias de usuario en categorías por tamaño, es más fácil ver si las historias de usuario a las que se les ha asignado estimaciones similares, son realmente comparables en tamaño. Esta técnica ayuda al equipo a asegurarse de que no han alterado de forma gradual el valor de medida de un punto de historia.

Una buena manera de realizar esta triangulación es establecer columnas en una muro que represente diferentes tamaños de historias de usuario, y entonces colocar cada carta de historia de usuario en la columna apropiada. Tal y como se estiman nuevas historias de usuario, deben colocarse en la columna apropiada y compararse con las cartas ya existentes. Si la nueva historia de usuario parece encajar con el resto de historias ya colocadas en la columna, perfecto – el esquema de estimación no se ha corrompido.

Si es totalmente diferente, no obstante, necesitamos realizar un punto de control con el equipo para discutir qué es exactamente un punto de historia y recalibrar sus estimaciones.

2.5.2.5. Tiempo, presupuesto, y estimación de coste (vía tamaño, esfuerzo, y estimación de coste)

Hemos visto diferentes técnicas y herramientas utilizadas en la creación de estimaciones. Veamos ahora los pasos involucrados en la estimación:

1. **Determinar el tamaño** del proyecto en puntos de historia o tiempo ideal.

2. **Calcular el esfuerzo** para el trabajo en horas o días persona (o semanas / meses persona) a través de determinar la disponibilidad y capacidad del equipo.

3. **Convertir el esfuerzo en una programación** utilizando el tamaño del equipo, los recursos requeridos y las dependencias.

4. **Calcular el coste** aplicando tasas de mano de obra y añadiendo otros elementos de coste del proyecto.

Veamos cada uno de ellos en detalle:

Paso 1: Determinar el tamaño del proyecto en puntos de historia o tiempo ideal.

Para completar este paso, utilizamos las técnicas que hemos visto antes, incluyendo la estimación "Wideband Delphi" y la planificación póker.

Paso 2: **Calcular el esfuerzo** para el trabajo en horas o días persona (o semanas / meses persona) a través de determinar la disponibilidad y capacidad del equipo.

Para completar este paso, necesitamos determinar la disponibilidad y capacidad del equipo. Esto significa que si hemos elegido estimar el trabajo en tiempo ideal en aras de la simplicidad y la velocidad, sabemos que tenemos que determinar cómo será la disponibilidad. Hacemos esto calculando la disponibilidad media esperada de los miembros del equipo y dividiendo entonces el tiempo ideal estimado por esta media.

Por ejemplo, imaginemos un pequeño equipo de tres personas – Bob, Bill, y Mary. Bob, típicamente, está disponible para trabajar en el proyecto el 80% del tiempo, Bill está disponible el 70% del tiempo, y Mary está disponible el 75% del tiempo. Utilizando esta información, calculamos la disponibilidad media del equipo: (0.80 + 0.70 + 0.75) / 3 = 0.75; o 75%.

Por lo tanto, si nuestro tiempo ideal estimado para el proyecto es de un total de 500 días, podemos calcular el esfuerzo probable dividiendo el tiempo real por la disponibilidad media: 500 / 0.75 = 670 días persona.

Tal y como el proyecto avance, necesitamos comprobar la disponibilidad y capacidad del equipo contra este cálculo para determinar si las aproximaciones iniciales son correctas. Si el mismo equipo de tres personas completa de forma repetida un total de 100 horas de trabajo válido por una semana de trabajo de 40 horas, su capacidad de trabajo actual es de 100 / (40 x 3) = 83%. En otras palabras, actualmente trabajan a un ritmo superior al estimado.

Tal y como el trabajo progresa, aumentamos la confianza en la velocidad actual del equipo cuando medimos el progreso futuro del proyecto. Utilizando el ratio de disponibilidad del 83 por ciento (el cual corresponde a la velocidad actual del equipo), podemos recalcular el esfuerzo de las 500 – día – ideal del proyecto: 500 / 0.83 = 603 días persona.

Paso 3: Convertir el esfuerzo en una programación utilizando el tamaño del equipo, los recursos requeridos y las dependencias.

Ahora tenemos que trasladar el esfuerzo en una duración programada y determinar si nuestro proyecto de 603 días puede ser completado por nuestro equipo de tres personas en 201 días (603 / 3 = 201) o en alguna otra duración.

Cuando creamos la programación, tenemos que plantear preguntas como:

- ¿Puede el trabajo realizarse independientemente de otras partes del trabajo?
- ¿Puede ser realizado por cualquier desarrollador en cualquier momento?
- ¿O hay dependencias de programación o problemas de habilidades que puedan requerir una duración mayor?

Como se representa por la I en el nemotécnico INVEST, "independiente" es una de las características de una buena historia de usuario. Esto quiere decir que las buenas historias de usuario no deben estar estrechamente unidas a otras historias. Naturalmente, la realidad tiene maneras de echar a perder este objetivo, pero en el examen PMI-ACP se tomará el enfoque más simple. Para el examen, se asume que las historias de usuario son totalmente independientes, lo que hace los cálculos tan simples como dividir el esfuerzo por el número de recursos.

Utilizando el ejemplo previo, vemos que se espera que la duración del proyecto sea de 201 días. Si queremos trasladar esta duración a meses, solo tenemos que dividir el número total de días por los días laborables del mes. Si asumimos una jornada laboral de 5 días, hay unos 21 días laborables en un mes típico, por lo tanto 201 / 21 = 9.6 meses. Por lo tanto, la duración programada de nuestro proyecto es de 201 días o 9.6 meses.

Paso 4: Calcular el coste aplicando tasas de mano de obra y añadiendo otros elementos de coste del proyecto.

Una vez que hemos calculado la duración del proyecto, es fácil calcular los costes en términos monetarios. La fórmula para este cálculo es:

Coste total = (Tiempo x ratio de recursos) + Otros costes del proyecto.

Veamos un ejemplo, utilizando el tiempo de proyecto estimado anteriormente de 9.6 meses.

Para realizar el cálculo, necesitamos conocer los ratios de los miembros del equipo. Digamos que el ratio de Bob es de $50 por hora, el ratio de Bill de $80 por hora y el ratio de Mary de $95 por hora. Después, necesitamos convertir el tiempo estimado para el proyecto a horas. Con una jornada diaria de 8 horas y con aproximadamente 21 días por mes, hay unas 168 horas por mes. Por lo tanto, 9.6 x 168 = 1.613 horas de proyecto.

Utilizando esta información, podemos calcular el coste de trabajo:

(1.613 x $50) + (1.613 x $80) + (1.613 x $95) = $80.650 + $129.040 + $153.235 = $362.925

Ahora debemos añadir otros costes del proyecto, como recursos a tiempo parcial, hardware, software, formación, suministros, viajes, relleno de representantes del negocio para liberarlos a contribuir al proyecto, el periodo de garantía después de que el producto salga a la venta, y cualquier otra cosa que el proyecto necesite. Una vez hemos calculado estos costes adicionales, tendremos nuestro coste total. En este punto, deberíamos añadir los fondos para contingencias al coste total, para obtener el presupuesto de nuestro proyecto. La cantidad de estos fondos se basa en el riesgo e incertidumbre del proyecto y en las políticas de la empresa en cuanto a contingencias. Por ejemplo, algunas compañías utilizan un rango del 10 al 20 por ciento de los costes totales del proyecto, para contingencias.

2.5.2.6. Principios de contabilidad en proyectos ágiles

Comprender el valor y beneficios de un proyecto es uno de los lados de la ecuación, mientras que comprender los costes seria el otro lado. Cuando consideramos los costes de realizar un proyecto basado en trabajadores del conocimiento, los costes salariales a menudo comprenden una gran parte de los gastos. Por lo tanto, aunque existirán costes asociados con la equipación, viajes, licencias, servicios especializados, etc., los costes salariales constituyen normalmente el mayor segmento.

Hay diferentes maneras de contabilizar los costes salariales del proyecto. Muchas compañías cogen el salario anual de un miembro del equipo y lo dividen por 52 para determinar el coste del salario semanal. Entonces utilizan este valor para las estimaciones del proyecto y para la realización del presupuesto. Otras compañías consideran el coste de los salarios totalmente gravados, formados por el salario del miembro del equipo más el coste de los metros que ocupa en la oficina, ordenadores, beneficios, etc. Este salario totalmente grabado puede ser fácilmente más del 50% del gasto de salario del empleado.

Como gestores de proyecto intentando estimar el proyecto, necesitamos saber cuál es la política de la empresa sobre la utilización de costes grabados sobre los no grabados. También necesitamos considerar la involucración del miembro del equipo en el proyecto. ¿Está dedicado al proyecto a tiempo completo, solo el 50%, u otro porcentaje? Una vez hemos comprendido estas variables, podemos calcular los costes salariales del proyecto, como se indica en la siguiente tabla:

Rol	Salario anual	Costes laborales totalmente gravados	Coste gravado por iteración	Tiempo en el proyecto	Coste ajustado por iteración
Propietario del producto	$120.000	$180.000	$6.925	100%	$6.925
Desarrollador	$100.000	$150.000	$5.770	100%	$5.770
Desarrollador	$80.000	$120.000	$4.615	50%	$2.310
Analista	$100.000	$150.000	$5.770	100%	$5.770
Tester	$90.000	$135.000	$5.195	100%	$5.195
				Total	$25.970

Tabla 16: Costes salariales del proyecto

Esta tabla nos indica que nuestro coste real probable por iteración es de unos $26.000. Por lo tanto si nuestro proyecto tiene 10 iteraciones, el coste salarial será de unos $260.000.
Utilizando estos datos y los indicadores de velocidad del equipo, podemos también calcular el coste por punto de historia. Por ejemplo, si el equipo tiene una media de 26 puntos de trabajo por iteración y cada iteración cuesta $26.000, entonces cada punto de historia viene costando $26.000 / 26 =$1.000 el desarrollarse.

Conocer el coste por punto de historia puede ser útil para validar rápidamente nuevas funcionalidades. Por ejemplo, si el responsable del producto pregunta sobre el posible esfuerzo y coste necesario para desarrollar un nuevo informe y el equipo lo estima en 5 puntos de historia, podemos reportar que el coste probable de desarrollo será de $5.000.

Rangos de estimación.

Las estimaciones deberían representarse en rangos para indicar nuestro nivel de confianza en la estimación y para gestionar las expectativas de los stakeholders. Cuando reportamos con estimaciones, necesitamos ser cautelosos con la calidad de nuestras variables de entrada – es fácil aplicar las matemáticas a un montón de estimaciones altamente especulativas y creer que hay más precisión en nuestras estimaciones de la que realmente hay.

También querremos evitar puntos de estimación únicos. Por ejemplo, diciendo “El proyecto costará $784.375,32” da una impresión errónea de exactitud. En lugar de eso, deberíamos presentar las estimaciones en rangos, como “Creemos que el proyecto costará entre $775.000 y $820.000”.

Los rangos de estimación serán más estrechos cuando tengamos más confianza en las estimaciones y más amplios cuando nuestra certeza sea menor.

Estimar proyectos ágiles tiene importantes similitudes y diferencias respecto a la estimación de proyectos tradicionales y no iterativos. Por ejemplo, cerca del comienzo de un proyecto ágil cuando sabemos lo mínimo sobre el proyecto, nuestras técnicas de estimación son similares a los métodos de estimación tradicionales. Aplicamos enfoques heurísticos (basados en conocimiento experto), como el revisar datos de proyectos que crearon productos similares o que tuvieron un tamaño o complejidad similar, para ver cuánto llevo el realizar estos proyectos. También podemos llevar a cabo estimaciones paramétricas (basadas en cálculos), como las estimaciones "bottom-up" que se basan en el número y complejidad de las historias de usuario.

Tal y como el proyecto ágil progresa, las iteraciones proveen evidencias del avance real del proyecto. Podemos empezar a introducir la velocidad de las iteraciones completas para juzgar mejor nuestro progreso real y estimar el resto del proyecto.

La razón por la cual la velocidad de las iteraciones completadas puede ser utilizada para estimar con precisión el progreso de un proyecto es porque las iteraciones contienen todos los componentes del sistema que el producto final utilizará. Esto quiere decir que nos exponemos a los diferentes aspectos del esfuerzo de desarrollo en cada iteración, por lo que pronto, tenemos la oportunidad de confiar menos y menos en las estimaciones tradicionales del inicio y en su lugar confiar mucho más en la velocidad emergente.

También podemos comparar nuestra velocidad con la cartera restante para estimar el calendario para completar el proyecto.

Por ejemplo, si después de las tres primeras iteraciones, nuestra velocidad se ha estabilizado y da una media de 50 puntos de historia por cada iteración de dos semanas, y la cartera de trabajos restantes contiene 500 puntos de valor de funcionalidad, es razonable asumir que nos llevará 10 iteraciones más el completar el trabajo restante (500/50 = 10). Ahora calcularemos el ratio "burn" por iteración, utilizando de nuevo a Bob, Bill y Mary como miembros de nuestro equipo:

- Para Bob, multiplicamos 10 días por 8 horas por su ratio de $50 por hora, para un total de $4.000.

- Para Bill, multiplicamos 10 días por 8 horas por su ratio de $80 por hora, para un total de $6.400.

- Para Mary, multiplicamos 10 días por 8 horas por su ratio de $95 por hora, para un total de $7.600.

Si ahora sumamos estos números entre sí ($4.000 + $6.400 + $7.600), obtenemos un ratio "burn" por iteración de $18.000. Ya que asumimos que nos llevaría 10 iteraciones el completar el trabajo restante, podemos calcular cuánto costará completar las características restantes: $18.000 x 10 iteraciones = $180.000.

Esta frecuente retroalimentación de los resultados actuales en las estimaciones es un punto de control muy valioso para el proyecto. El bajo rendimiento se descubre pronto, ya que los proyectos ágiles miden el progreso en función del número de historias de usuario aceptadas, más que en estimaciones del tanto por ciento completado de los entregables de diseño. Descubrir el bajo rendimiento nunca es agradable, pero es mejor descubrirlo pronto en el proyecto, ya que tendremos más tiempo para tomar acciones correctoras.

Ley de Parkinson y Síndrome del estudiante.

Las prácticas ágiles ayudan también a mitigar los efectos de la Ley de Parkinson y el Síndrome del Estudiante. La Ley de Parkinson establece que el trabajo tiende a expandirse para rellenar el tiempo disponible. En otras palabras, si tenemos tres meses designados para obtener los resultados, definitivamente gastaremos los tres meses para conseguir esos resultados. A causa de que los proyectos ágiles se centran en cortos periodos de tiempo, como iteraciones de una o dos semanas, hay "menos tiempo a rellenar".

La idea detrás del Síndrome del Estudiante es que cuando a las personas se les da una fecha límite, tienden a esperar hasta que casi han llegado a esa fecha antes de empezar el trabajo. De nuevo, las iteraciones de dos semanas de los proyectos ágiles hacen que esa fecha límite no es nunca demasiado lejos, lo que ayuda a mantener a las personas centradas.

2.5.3. Planes ágiles

Ante de hablar de los documentos y herramientas de planificación actuales utilizados en los proyectos ágiles, compararemos los enfoques de planificación de los proyectos ágiles y tradicionales. La planificación ágil varía respecto a la tradicional en tres maneras principales.

En los proyectos ágiles:

1. Las pruebas y las demostraciones descubren los requisitos reales, los cuales requieren entonces planificación.

2. La planificación ágil no requiere un gran esfuerzo inicial, pero si un esfuerzo continuado a lo largo del proyecto.

3. Los ajustes a medio plazo son la norma.

Veamos estos puntos con más detalle.

1. Las pruebas y las demostraciones descubren los requisitos reales, los cuales requieren entonces planificación.

Cuando llevamos a cabo ciertas iniciativas como los proyectos con trabajadores del conocimiento, conseguir el apoyo de los stakeholders es crítico para el éxito del proyecto. ¿Pero cómo conseguimos que los stakeholders estén de acuerdo con los requisitos? A menudo es mejor tener una discusión inicial sobre la visión del producto y progresar iterativamente desde ahí con un prototipo tangible. Entonces podemos permitir a los stakeholders ajustar el proyecto basándose en su experiencia con el prototipo, en lugar de esperar a que describan completamente con palabras a que se debería parecer el producto y que debería hacer cuando aún existe incertidumbre sobre el ámbito y la solución propuesta.

Por lo tanto, en lugar de crear especificaciones y planes muy detallados, los proyectos ágiles construyen un prototipo para entender mejor el dominio y uso de ese prototipo como la base para planificaciones y elaboraciones posteriores.

2. La planificación ágil no requiere un gran esfuerzo inicial, pero si un esfuerzo continuado a lo largo del proyecto.

La guía PMBOK pone un énfasis muy fuerte en la planificación inicial; sugiere centrarse en planificar esfuerzos para el ámbito del proyecto, el calendario, el presupuesto, la calidad, los recursos humanos, las comunicaciones, los riesgos, las contrataciones, los cambios de gestión, la configuración de la gestión, y los procesos de mejora, todo antes de empezar a trabajar en el proyecto. En contraste, los métodos ágiles reconocen que el nivel de riesgo e incertidumbre en los proyectos con trabajadores del conocimiento pueden hacer problemática la planificación inicial, por lo que extienden la planificación mucho más, incluso durante todo el ciclo de vida del proyecto. Extender el esfuerzo permite al proyecto ajustarse mejor a la información emergente.

Como hemos visto repetidamente, los proyectos con trabajadores del conocimiento son a menudo intangibles y los requisitos son difíciles de articular. Rara vez se construye el mismo sistema dos veces, lo cual hace difícil crear analogías a funcionalidades existentes. Estos problemas pueden llevar a la “evaluación de las dificultades”, donde se desarrollan los desajustes entre las interpretaciones del equipo y los requisitos reales que el cliente quiere.

Ejecutar un proyecto de trabajadores del conocimiento es un desafío complejo, creativo y de alto riesgo. A diferencia de los proyectos de fabricación predecibles, el trabajo a menudo es un proceso de búsqueda y desarrollo sin precedentes. Intentar crear planes detallados y orientados a tareas para trabajadores del conocimiento probablemente lleve a planes frágiles y que serán abandonados pronto. O, si los planes no se abandonan, puede haber una gran cantidad de tiempo de gestión gastado en actualizar estos grandes planes, en lugar de gestionando el proyecto.

Desafortunadamente, muchos proyectos llevan a cabo la mayor parte de su planificación demasiado pronto en los ciclo de vida, cuando se conocen pocos datos de forma concreta sobre el dominio del proyecto, el ambiente del negocio, o la dinámica del equipo. El énfasis de la planificación con la gestión de proyectos tradicional se muestra en la figura 48. Como se indica por la región verde, la planificación ha sido una de las actividades predominantes en el inicio del ciclo de vida y actividad del proyecto.

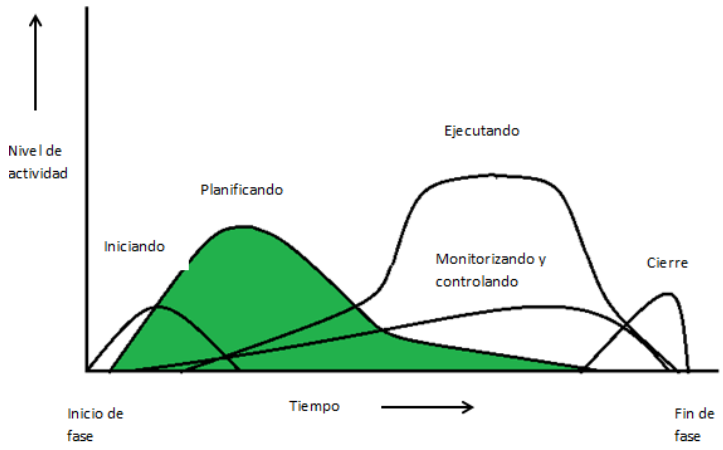


Figura 48: Foco de la planificación en un enfoque tradicional de proyectos. (Fuente: Elaboración propia)

Los métodos ágiles aceptan las realidades de los proyectos de trabajadores de conocimiento y deliberadamente hacen de la planificación un componente más visible e iterativo en el ciclo de vida del proyecto. En la figura 49, la sección verde repetida indica que la planificación se lleva a cabo a través de todo el ciclo de vida.

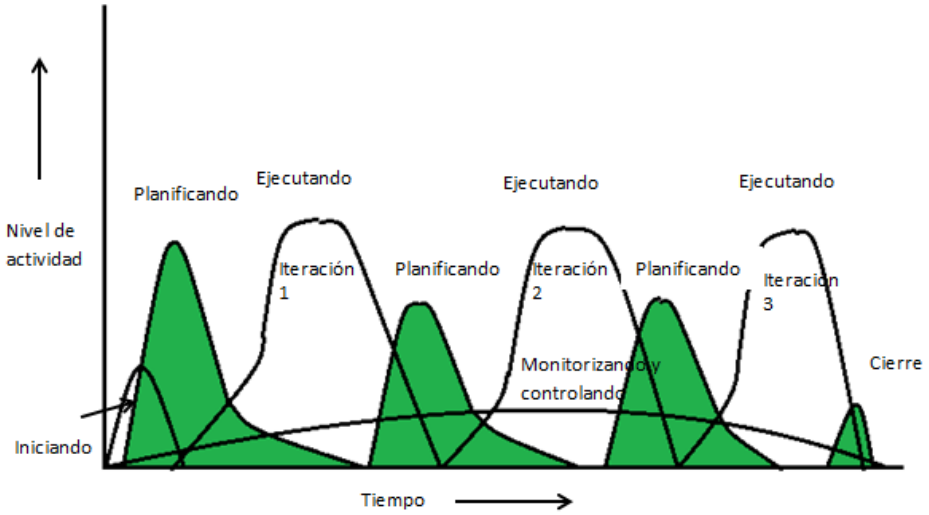


Figura 49: Foco de la planificación en un enfoque ágil de proyectos. (Fuente: Elaboración propia)

La cantidad total de planificación en un proyecto ágil (combinando todas las zonas verdes) a menudo acaba siendo más del que se lleva a cabo en un proyecto tradicional. Por lo tanto, en los proyectos ágiles, acabamos haciendo más planificación, no menos, pero las actividades se distribuyen de diferente forma. No importa el esfuerzo, siempre hay una cantidad responsable de planificación inicial que debería ser realizada.

En la figura 50 se han ilustrado los riesgos de no realizar la suficiente planificación inicial.

Figura 50: Pérdidas por planes inadecuados. (Fuente: Elaboración propia)

Podemos ver en la parte izquierda de la figura anterior que cuando se invierte poco tiempo en planificar, el riesgo de equivocaciones y de retrasos debidos a tener que rehacer el trabajo es alto. Entonces, cuanto más tiempo y esfuerzo se invierta en planificar, más bajará el riesgo. Pero lo que algunas personas no ven es que hay un riesgo corolario de realizar demasiada planificación inicial. Este riesgo se representa en la figura 51.

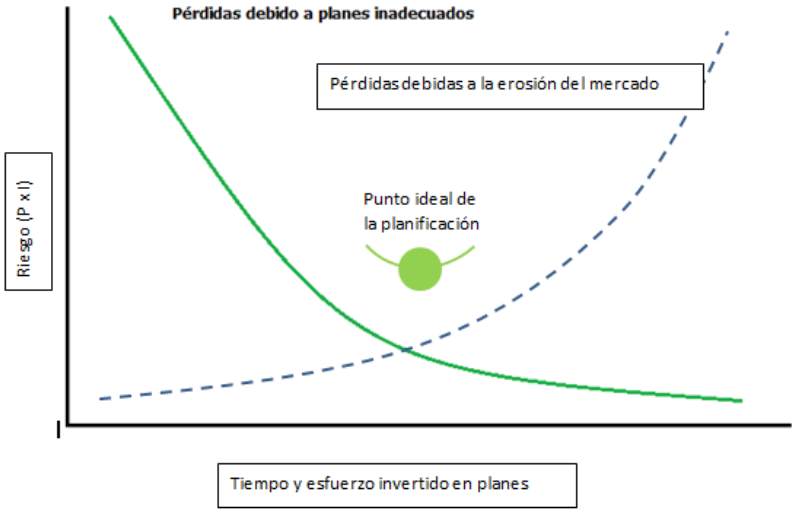


Figura 51: Riesgo corolario de realizar demasiada planificación inicial. (Fuente: Elaboración propia)

En la figura anterior vemos que la línea punteada representa los riesgos involucrados en realizar demasiada planificación inicial. El riesgo comienza bajo, pero cuanto más tiempo y esfuerzo se invierte en la planificación inicial, más se incremente el riesgo de crear planes muy detallados pero frágiles, así como el riesgo de retrasar la entrega del proyecto y la rentabilidad de la inversión ya que se gasta mucho tiempo planificando.

Por lo tanto debemos aspirar al punto óptimo (sweet spot), en el cual la suma de estos dos riesgos es más baja. Necesitamos realizar la suficiente planificación inicial para reducir la mayor parte del riesgo de tener que rehacer trabajos, mientras que evitamos sobreplanificar para minimizar el riesgo de retrasar la rentabilidad de la inversión.

Estas curvas son válidas para cualquier tipo de proyecto, pero debido a la naturaleza cambiante de los proyectos ágiles, el riego de romper la planificación ocurre antes que en proyectos más estables y predecibles. Por lo tanto, la cantidad de planificaciones inicial necesaria para los proyectos ágiles es típicamente inferior que en otros proyectos.

3. Los ajustes a medio plazo son la norma.

Cuando aspiramos a un objetivo estático, es apropiado aspirar cuidadosamente y entonces apuntar hacia él. Cuando aspiramos a un objetivo móvil que no sigue un camino fiable, necesitamos más que un enfoque tipo misil-guiado, realizando muchos ajustes a medio vuelo para asegurar que llega al objetivo.

Un misil-guiado se apunta en una dirección general y entonces, una vez que ha sido lanzado, un sofisticado sistema de sensores, propulsores, y sistemas retro alimentados, guían al misil al objetivo en movimiento. Esta analogía es apropiada para los proyectos ágiles, ya que a menudo tienen objetivos en movimiento. El producto que quiere el cliente puede transformarse por requisitos de última hora, los cuales se activan por cambios en el ambiente del negocio u ofertas de los competidores. Para permanecer en el blanco, los métodos ágiles utilizan sofisticados sistemas de detección y adaptación para obtener feedback y realizar los ajustes a la cartera y los planes tal y como se realiza el trabajo del proyecto.

Lo que se debe sacar en claro de estas analogías es que los métodos ágiles no rehúyen de la planificación. Aunque a un observador casual pueda parecerle que la planificación ha sido evitada mientras que el equipo va hacia adelante y desarrolla cosas, realmente hay una evolución de los procesos de planificación. Esta manera de planificar se adapta mejor a ambientes de alto cambio. La planificación continúa existiendo, primero en un plan de lanzamiento global, y luego en múltiples puntos a lo largo del proyecto para planificar las siguientes iteraciones y lanzamientos. Los proyectos ágiles también tienen muy en cuenta el feddback en sus procesos de planificación sobre la marcha. Por ejemplo:

- La repriorización de la cartera afecta a los planes de lanzamiento e iteración.
- El feedback que proviene de las iteraciones y demostraciones, genera cambios en los requisitos y nuevos requisitos.
- Las retrospectivas generan cambios en los procesos y técnicas.

2.5.3.1. Cartas ágiles

La carta de proyecto es uno de los primeros documentos producidos en un proyecto. Describe los objetivos del proyecto, el propósito, la composición, y el enfoque, y provee de la autorización del sponsor para proceder con el proyecto. Las cartas ágiles pueden ir desde hojas de trabajo muy simples con visiones poco expandidas de las declaraciones, a documentos extensamente detallados.

Las cartas ágiles asimilan que el ámbito puede cambiar y que inicialmente se pueden desconocer algunos detalles del proyecto. Por lo tanto, en lugar de tratar de especificar detalladamente el ámbito del proyecto, las cartas ágiles caracterizan los objetivos que se han visionado para el proyecto. Además describen los procesos y enfoques que el equipo debería utilizar para iterar hacia el producto final, así como el criterio de aceptación que se utilizará para verificar las entregas del producto.

Las cartas ágiles típicamente responden al conjunto de 5 preguntas siguiente:

- ¿De qué trata este proyecto? (Una descripción a alto nivel de la visión, misión, metas y objetivos del proyecto).
- ¿Por qué se lleva a cabo? (La lógica del negocio).
- ¿Cuándo comenzará y cuándo acabará?) (Las fechas de inicio y final objetivo).
- ¿Quién debe comprometerse? (Una lista de los participantes del proyecto y de los stakeholders).
- ¿Dónde se llevará a cabo? (Detalles de los lugares de trabajo, requisitos para su despliegue, etc.).
- ¿Cómo se llevará a cabo? (Una descripción del enfoque, el cual es particularmente importante para los proyectos ágiles si éstos son nuevos para la organización; los cambios respecto a los enfoques estándar, como el incremento de involucración de los clientes, necesitan ser explicados).

El proceso de requerir ayuda, alinea a los stakeholders con el proyecto. Una manera de ayudar a los stakeholders a explorar y cementar las bases del proyecto es que desarrollen conjuntamente los enunciados del proyecto. Los enunciados del proyecto son descripciones cortas de los objetivos, beneficios y discriminadores del proyecto. Estos enunciados describen rápidamente el proyecto o producto. A continuación puede verse un formato típico para estas tablas.

Para:	Clientes objetivo
Quien:	Necesidades (oportunidades o problemas)
Él:	Producto / nombre del servicio
Es un:	Categoría del producto
Que:	Beneficios principales / razón para comprar
Diferente a:	Alternativas principales de la competencia
Nosotros:	Diferenciación principal

Tabla 17: Ejemplos de cartas ágiles

Utilicemos como ejemplo el describir un nuevo curso para estudiantes:

Para:	Gestores de proyecto
Quien:	Quieren ser líderes de proyectos ágiles
Él:	“Aprender a llevar equipos ágiles”
Es un:	Curso de tres días
Que:	Lleva a los gestores de proyecto a través de un ciclo de vida de desarrollo ágil, incorporando casos reales de estudio y ejercicios.
Diferente a:	Cursos ágiles de organizaciones genéricas.
Nosotros:	Únicamente utilizamos instructores con experiencia en proyectos ágiles para asegurarnos que puedan resolver todas tus preguntas, y nuestros materiales suplementarios incluyen valiosas herramientas, casos de estudio, y hojas de trucos.

Tabla 18: Ejemplos de cartas ágiles

El formato del texto final de este enunciado seria:

Para gestores de proyecto que quieran convertirse en líderes de proyectos ágiles, el curso de tres días "Aprender a llevar equipos ágiles", lleva a los gestores de proyecto a través de un comprensivo ciclo de desarrollo de un proyecto ágil, incorporando casos reales de estudio y ejercicios prácticos. A diferencia de los cursos ágiles de las escuelas tradicionales, únicamente utilizamos profesores con experiencia práctica en proyectos ágiles para asegurar que puedan resolver todas tus preguntas, y nuestro material suplementario incluye valiosas herramientas, casos de estudio, hojas de trucos.

Otro ejercicio rápido que puede ayudar a alinear a los stakeholders alrededor del proyecto es crear un Tweet del proyecto. Este ejercicio requiere que los stakeholders describan el objetivo del proyecto en 140 palabras o menos. El intento de esta actividad es crear una descripción que lo abarque todo, pero sin evaluar la comprensión de los stakeholders del proyecto y de sus prioridades.

Ejercicios como crear los estatutos del proyecto o un Tweet de proyecto ayudan a desarrollar la carta. Entonces, una vez que se ha completado y aprobado la carta, ésta sirve como dispositivo para lanzar el proyecto, dando la autoridad de proceder al equipo, siguiendo el enfoque descrito o referenciado en la carta.

2.5.3.2. Caso de desarrollo de negocio

El caso de desarrollo de negocio para los proyectos ágiles no es tan diferente del utilizado en los proyectos tradicionales. Para la mayor parte, las diferencias radican en un nivel más bajo de documentación para los proyectos ágiles. Además, el caso de desarrollo de negocio para los proyectos ágiles puede incluir la discusión de los beneficios de las oportunidades tempranas, las cuales no son típicas de los proyectos tradicionales.

Hemos cubierto muchos de los conceptos y cálculos que alimentan el caso de proyecto de negocio en capítulos pasados, entrega basada en el valor. Los proyectos ágiles con negocios basados en el valor, y por lo tanto cualquier cálculo sobre el valor, como la rentabilidad de la inversión (ROI), la tasa interna de rentabilidad (TIR), y el valor neto actual (VAN), será capturado en el caso de negocio.

El caso de negocios describe porque la organización que hace de sponsor debería llevar a cabo el proyecto. Cuando hay múltiples proyectos compitiendo por un sponsor con fondos limitados, la organización debe evaluar los casos de negocio para determinar que proyector elegir.

Como se mencionó en la parte de la entrega basada en el valor, el valor del negocio de un proyecto normalmente se evalúa en términos monetarios. Para proyectos que son llevados a cabo con propósitos de seguridad o regulatorios y que no tiene un valor monetario fácil de determinar, podemos mirar las consecuencias de no llevar a cabo el proyecto (como el cierre del negocio, multas, demandas…), o podemos simplemente marcarlos como obligatorios y no perder más tiempo investigando su valor.

La cantidad de detalle en un caso de negocio ágil varía de un proyecto a otro. Para proyectos ágiles pequeños, puede no haber ningún documento exclusivo del caso de negocio. En su lugar, la discusión del caso de negocio y de los beneficios del negocio puede estar incluida en la carta del proyecto o en el documento de visión del proyecto o estatutos. Para proyectos ágiles grandes o aquellos que se llevan a cabo dentro de una organización que exige la creación de un documento para el caso de negocio, podríamos ver un documento incluyendo las siguientes secciones:

- **Visión de conjunto del proyecto:** Una corta descripción del trasfondo del proyecto y de sus objetivos.

- **Costes anticipados:** Cuanto estimamos que va a costar el proyecto, incluyendo fondos para contingencias razonables.

- **Beneficios anticipados:** Los beneficios, monetarios o de cualquier otro tipo, que estimamos va a originar el proyecto, como nuevos ingresos, ahorro de costes, un mejor servicio, mejora de la calidad, cumplimiento, condiciones de trabajo mejoradas, incremento de la satisfacción del cliente, etc.

- **Modelo de negocio e índices:** Cálculos del ROI, TIR, y VAN, como requiera la organización.

- **Hipótesis de ROI y riesgos asociados con el proyecto:** Las hipótesis que se hicieron al calcular los modelos e índices de negocio, como los ratios de inflación y los ratios de aprobación del cliente.

- **Riesgos de no llevar a cabo el proyecto:** Los riesgos (si hay alguno) que la organización afronta si elige no llevar a cabo el proyecto.

- **Análisis SWOT / PEST:** Los resultados relevantes de los análisis SWOT y PEST; Los análisis SWOT (que viene de fuerza (Strengths), debilidades (Weaknesses) oportunidades (Opportunities), y amenazas (Threats) y PEST (que viene de política (Political), economía (Economic), social (Social) y técnica (Technical), son herramientas gráficas para examinar los diferentes atributos del proyecto.

- **Recomendaciones:** Las recomendaciones del sponsor para el proyecto, como las Obligatorias (cumplimiento con las regulaciones) o Críticas (la prioridad más alta del sponsor), o las Altas, Medias, o Gustaría tenerlas.

2.5.3.3. Planificación de la iteración y la entrega

Los proyectos ágiles se dividen en entregas e iteraciones. Como hemos visto en puntos anteriores, una iteración es un periodo corto de desarrollo, de una duración típica de dos a cuatro semanas. Una entrega es un grupo de iteraciones que resultan en la finalización de un entregable valioso del proyecto. Por lo tanto, un proyecto tiene una o más entregas, y cada entrega contiene una o más iteraciones, como vemos en la figura 52. En esta figura, vemos un proyecto simple con dos entregas a producción. La primera entrega contiene 11 iteraciones, y la segunda 5.

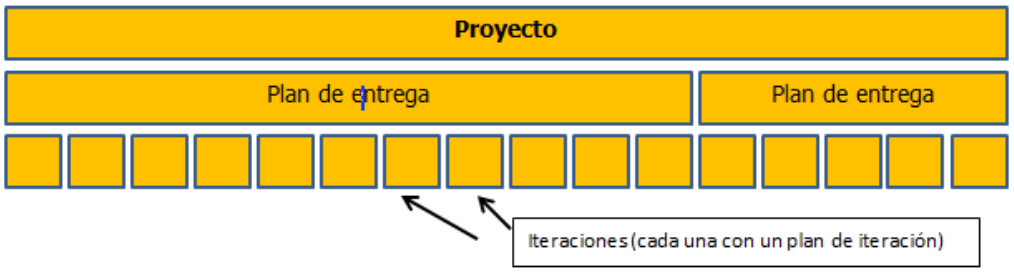


Figura 52: Proyecto. (Fuente: Elaboración propia)

Empezamos a planificar entregas e iteraciones al inicio del ciclo de vida del proyecto y revisamos el esfuerzo de planificación múltiples veces así como avanza el proyecto.

Planificación de la entrega.

Las entregas se planifican alrededor de la entrega de incrementos de funcionalidad útil y valiosa para el negocio o el cliente. Una entrega puede estar supeditada a una fecha ("Necesitamos algo para mostrarlo en la feria de muestras") o supeditada a una funcionalidad ("Una vez que podamos capturar y procesar los pedidos de los clientes, queremos ponernos en marcha; la gestión de los reportes y las renovaciones de cuenta pueden venir después"). Sea lo que sea que supedite la entrega, necesitamos determinar la funcionalidad que puede ser desarrollada y entregada para el lanzamiento planificado.

Por decirlo de otra forma, cuando planificamos una entrega preguntamos, "¿Que porción de la historia de usuario puede ser entregada en este lanzamiento?

La funcionalidad que representa todas las historias de usuario clasificadas como "must haves" (debemos tenerlo) y "Should have" (deberíamos tenerlo) han sido seleccionadas para la siguiente entrega. La pregunta entonces sería, "¿Cómo de probable es que seamos capaces de completar el trabajo para la fecha de entrega?". Para evaluar qué será capaz de completar el equipo y cuándo lo completará, confiamos inicialmente en las estimaciones del equipo para la primera entrega.

Entonces, después de que el equipo haya pasado por unas cuantas iteraciones, podemos comenzar a fijarnos en los indicadores de velocidad, como se muestra en la figura 53.

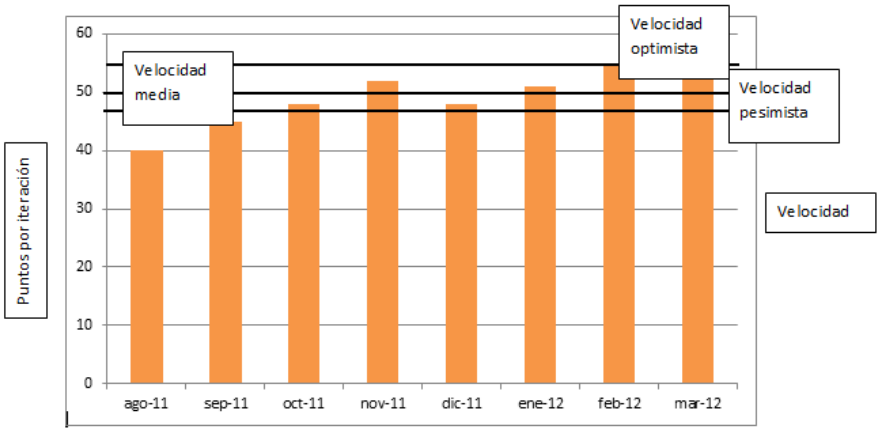


Figura 53: Indicadores de velocidad. (Fuente: Elaboración propia)

En el ejemplo de la figura anterior, la velocidad empezaba con 40 puntos completados en Agosto. En Noviembre, el proyecto rompió la marca de los 50 puntos, y entonces en Febrero, se alcanzó la velocidad de 55 puntos. Si queremos calcular el mejor caso de entrega, podemos utilizar la velocidad de 55 puntos para realizar una predicción optimista de cuantos puntos de historia pueden ser entregados en cada iteración, pero no parece muy probable que el equipo sea capaz de conseguir esta velocidad de forma consistente, dada la velocidad contabilizada en los otros meses del proyecto. Sería más seguro asumir un ratio de velocidad mensual más lento para la entrega. Basándonos en los datos del gráfico, podríamos utilizar un valor de 48 puntos de historia por mes para una estimación pesimista. Para la estimación más probable, calculamos la velocidad media, la cual en este caso es de 50 puntos de historia por mes.

Por lo tanto, si nos han dicho que los responsables del negocio están pidiendo realizar la entrega en dos iteraciones, necesitamos encontrar donde nos llevará esto en la cartera de historias de usuario. Empezamos calculando 2 meses x 50 puntos por mes = 100 de valor de funcionalidad. Sumamos entonces los puntos de historia de los elementos de la cartera desde la parte alta hasta la baja para encontrar donde están los 100 puntos en la cartera.

En este ejemplo, 100 puntos nos llevan a completar las funcionalidades para el “must have” y el “should have”, así como un par de historias “could have”. Por lo tanto, exceptuando circunstancias imprevistas, parece posible entregar todas las historias “must have” y “should have” para el lanzamiento. Estos cálculos ayudan a asegurar que no intentamos comprometer más funcionalidad de la que podemos desarrollar dentro de la entrega.

Mapas de historia.

Como se vio en la parte de entregas basadas en el valor, los mapas de historia pueden ser útiles formas para mostrar los planes de entrega. Un mapa de historia nos permite exponer y agrupar historias, primero por dependencias (el “backbone” y el “walking skeleton” representan los elementos que necesitamos tener en el proyecto) y luego por funcionalidad. Cuando utilizamos mapas de historia para mostrar planes de entrega, continuamos necesitando añadir los valores de puntos de historia dentro de cada entrega y comprobar si el plan es viable, dada la capacidad de desarrollo del equipo y la cantidad de tiempo designado para la entrega.

Planificación de las iteraciones.

La planificación de las iteraciones sigue la misma idea que la planificación de las entregas. En la planificación de las iteraciones, pedimos al equipo que seleccione historias de usuario las cuales el cliente ha señalado como elementos de alta prioridad y que pueden ser desarrollados, comprobados, y entregados dentro de la iteración. Volviendo al ejemplo anterior en el cual el equipo está promediando unos 50 puntos de historia por mes, o sobre 25 puntos de historia para cada dos semanas de iteración, utilizaremos 25 puntos de historia como nuestra meta de planificación para la iteración.

Sería fácil para el ScrumMaster o gestor de proyecto seleccionar los siguientes 25 puntos de funcionalidad válida de la parte alta de la cartera y decir al equipo que siga adelante con la iteración. No obstante, hacer esto podría minar las características del equipo de autoorganización y potenciación en cuya creación hemos trabajado tan duro. Por lo tanto, es importante para el equipo discutir los objetivos para la próxima iteración con el cliente o el responsable del producto. El equipo debería entonces seleccionar las historias de usuario las cuales se compromete a entregar como parte de la próxima iteración.

Debemos recordar que los miembros del equipo son los más cercanos a los detalles técnicos del proyecto y pueden tener acceso a información a lo cual no acceden el gestor del proyecto o el ScrumMaster. Por ejemplo, podrían tener programado el repetir algunos trabajos (reparar o mejorar trabajo previo) para esta iteración lo que reduciría su capacidad para afrontar nuevos trabajos. O a la inversa, el equipo puede estar confiado de poder acabar trabajo adicional en esta iteración basándose en importantes iniciativas de mejora que han llevado a cabo recientemente. Por lo tanto, mientras que el ScrumMaster o el gestor de proyecto deberían cuestionar cualquier proyección de la iteración que varíe de la media de velocidad del equipo, es importante dejar que el equipo planee sus propias iteraciones.

Con el enfoque Scrum de los proyectos ágiles, la planificación de la iteración implica un poco más de ceremonia que con otros enfoques, por lo que vamos a ver con más detalle este enfoque. El responsable del producto debe ir preparado a la reunión de planificación de la iteración con una cartera recién priorizada.

En la primera mitad de la reunión, el responsable del producto describe los elementos de la cartera que quiere desarrollar en la iteración, y los miembros del equipo seleccionan un conjunto de elementos que ellos creen son alcanzables. El responsable del producto tiene la última palabra en las prioridades de la iteración, y el equipo de desarrollo tiene la última palabra en la cantidad de trabajo que puede llevarse a cabo en la iteración.

En la segunda mitad de la reunión de planificación de la iteración, el equipo divide los elementos seleccionados de la cartera en sus partes constituyentes para formar la cartera de elementos de la iteración. Entonces discuten como se llevará a cabo el trabajo, realizando compromisos externos e internos para llevar a cabo el trabajo dentro de tiempo de iteración. En esta fase, no es aún importante para algunas personas tomar el control de las tareas. Estas asignaciones pueden llegar después de la reunión de planificación de la iteración y pueden cambiar a lo largo de ésta.

La nueva cartera de la iteración sirve como el plan del equipo y compromete a entregar las historias seleccionadas de la cartera dentro de la iteración. Para que el equipo esté seguro de entregar este trabajo, deben poder calcular su capacidad y ajustarla a cualquier ausencia planificada. Volvamos a nuestro equipo de Bob, Bill y Mary y veamos un ejemplo. Si Bob ha planeado tomarse una semana de vacaciones durante el periodo de dos semanas y asumimos que todos contribuyen igual a la iteración, entonces la capacidad del equipo será reducida 1/6 para esta iteración particular, ya que 2 semanas x 3 personas = 6 semanas. Ahora para calcular la capacidad reducida de la iteración, cogemos nuestra capacidad media de 25 puntos de historia y le restamos 1/6 de los 25 puntos. Por lo tanto 25-4 = aproximadamente 21 puntos.

Para contabilizar días fuera en lugar de semanas, deberíamos empezar calculando la capacidad reducida multiplicando el número de miembros del equipo por el número de días en la iteración, y entonces restar el número de días que el miembro del equipo no está disponible. Para este ejemplo, Bob está tomando 5 días, y hay 10 días en la iteración. Por lo tanto calculamos 3 x 10 = 30, y entonces 30 – 5 = 25 días disponible. Por lo tanto la nueva capacidad del equipo será de 25 puntos x (25/30) = aproximadamente 21 puntos.

Además para ajustar la capacidad durante la reunión de planificación de la iteración, para introducir las ausencias, los equipos deben determinar un regulador apropiado (basado en expectativas reales, más que en relleno arbitrario) para completar trabajo no anticipado, resolver problemas, y batallar con otros muchos problemas. Por lo tanto si el equipo normalmente tiene una media de 25 puntos, sería inteligente por su parte el comprometerse a un nivel reducido que les permita un colchón para completar historias de usuario que se vuelven más largas de lo anticipado o manejar distracciones que utilicen recursos para su reparación.

Responsabilidad y éticas profesionales.

Antes de continuar con el siguiente capítulo, veremos como el Código PMI de conducta ética y profesional, aplica a los conceptos de planificación adaptativa. Hay cuatro áreas principales en este código – Responsabilidad, Respeto, Justicia y Honestidad – las cuales veremos por turnos, viendo ejemplos de cómo se relacionan con los tópicos vistos en este capítulo.

RESPONSABILIDAD.

Tomar decisiones basándose en los mejores intereses para la compañía. Esta categoría incluye la responsabilidad de reportar pronto las caídas de velocidad y ayudar a los stakeholders a decidir qué hacer si parece que alguna de las funcionalidades necesarias no se completará a tiempo.

RESPETO.

Mantener una actitud de mutua cooperación. En la planificación, se debe mantener un cuidadoso equilibrio entre el responsable del producto, que quiere cuanta más funcionalidad posible, el equipo del proyecto, quien quiere gestionar el campo de aplicación de manera que sean capaces de entregar exitosamente las historias de la iteración a tiempo, y el ScrumMaster o gestor del proyecto, quien ayuda a coordinar las diferentes partes. Mantener una actitud de mutua cooperación es vital para mantener a estos stakeholders controlados y colaborando productivamente. Algunas veces puede ser necesario para un grupo comprometerse o realizar concesiones de acuerdo a las necesidades de otros grupos. Las reuniones de revisión de la iteración y las retrospectivas, son buenas oportunidades para evaluar con los stakeholders y asegurarse de que sienten que sus metas están siendo respectadas y para investigar las consecuencias de cualquier compromiso o concesión.

Negociar de buena fe. Como con otros aspectos de la planificación, negociar es un acto equilibrado entre las partes involucradas. Aunque los clientes o responsables del producto, el equipo de desarrollo, y los líderes de proyecto, quieren que el proyecto sea un éxito, a menudo tiene diferentes visiones de ese éxito. Durante las negociaciones sobre el ámbito del proyecto y su funcionalidad, debemos respetar y comprender las peticiones de los stakeholders.

Enfréntate al conflicto directamente. No podemos simplemente ignorar los conflictos que ocurren durante el proyecto. Por ejemplo, si se originan problemas y un conflicto a causa de las estimaciones durante una sesión de planificación póker, necesitamos asegurarnos de que el equipo resuelve los problemas antes de continuar adelante. Aparcar los debates sobre las discusiones para más tarde, raramente es una estrategia productiva a largo plazo. En lugar de eso, es mejor dejar que el equipo llegue a una decisión aceptable sobre los problemas.

No usar tu posición para influenciar a otros. Si eres un líder de equipo, ScrumMaster, o gestor de proyecto, no deberías utilizar el poder de tu posición para influenciar excesivamente en las estimaciones del equipo o decisiones. Los beneficios que un equipo capacitado ofrece al proyecto cuando el equipo realiza la estimación, planificación, y la toma de decisiones son más valiosos que ir por tu propio camino o balancear una estimación.

JUSTICIA.

No discriminar. Debemos ser cuidadosos de no discriminar a miembros del equipo durante la planificación y a lo largo del proyecto. Por ejemplo, incluso cuando el equipo completo o algunos miembros individuales repetidamente subestiman (o sobrestiman) pares del proyecto, no debemos utilizar esto contra ellos o ridiculizarlos. En lugar de eso, necesitamos ayudarlos a mejorar sus habilidades a la hora de estimar cuando discutimos las duraciones actuales y los costes en las retrospectivas.

HONESTIDAD.

Comprender la verdad. Cuando las estimaciones son o demasiado altas o demasiado bajas, necesitamos ayudar al equipo a investigar por qué varían. Debemos entender que, muchas veces, las variaciones de causas comunes son las culpables. En otras palabras, algunas cosas varían, y el nuevo trabajo es difícil de estimar. No obstante, cuando repetidamente aparece una tendencia, debemos explorar la varianza utilizando técnicas objetivas para mantener la investigación a parte de lo personal.

Ser sincero en todas las comunicaciones. Debemos discutir abiertamente las contingencias con el equipo y el cliente, en lugar de enmascarar las estimaciones y dar fechas de entrega falsas al equipo de desarrollo para crear una sensación de urgencia. Es mejor explicar honestamente los hechos del proyecto y el papel y necesidad de las provisiones para contingencias. Si se nos pide eliminar los planes de contingencias, tenemos la responsabilidad profesional de explicar los riesgos y comunicar el impacto de cualquier coste y planificar los sobrecostes que acurran en el proyecto.

EJERCICIOS

En el Anexo F.5, se encuentra un examen tipo sobre lo tratado en este punto.

2.6. DETECCIÓN Y RESOLUCIÓN DE PROBLEMAS

Este capítulo trata de la identificación y resolución de problemas. La efectividad con la que un equipo se enfrente a los problemas, tiene un impacto crítico en el éxito o el fracaso de un proyecto. Es muy fácil ignorar un problema y continuar con el proyecto, deseando que el problema se resuelva por si mismo. Pero los problemas eventualmente paran nuestro trabajo y a menudo nos hacen tener que rehacer el trabajo que ya teníamos hecho. Ser golpeado con la necesidad de diagnosticar un problema y luego tener que rehacer trabajo para corregirlo puede tener un efecto dramático en las planificaciones de nuestro proyecto.

Para ilustrar este concepto, veamos un ejemplo. Asumamos que nuestro proyecto consiste es conducir 10 millas en 10 horas.

En la milla 2, todo va bien. Hemos estado viajando 2 horas, y hemos avanzado 2 millas. Esto quiere decir que los resultados de nuestro plan y nuestro rendimiento actual son iguales.

De hecho, las cosas van exactamente de acuerdo al plan para las primeras 4 horas, en las cuales cubrimos 4 millas, pero entonces nos encontramos con un problema. Con este problema en nuestro camino, completar el proyecto de acuerdo al plan deja de parecer posible.

Cuando se encuentran problemas en los proyectos, lleva tiempo el diagnosticar qué clase de problemas son y determinar que se debe hacer con ellos. El tiempo utilizado en diagnosticar el problema es tiempo que no se gasta en la ejecución del proyecto, y por lo tanto se retrasa la planificación. En el ejemplo, es necesaria una hora para diagnosticar el problema.

La solución a un problema a menudo engloba deshacer el trabajo defectuoso. Tal vez pensamos que una dirección o un enfoque eran apropiados para el proyecto, pero hemos acabado viendo que no funcionan. Ahora necesitamos volver atrás y probar un nuevo enfoque. Esto es un golpe doble – no solo nos lleva tiempo el eliminar o deshacer el trabajo mal hecho sino que además nos retrasa en la cantidad de trabajo que debe realizarse en el proyecto. Por lo tanto perdemos tiempo mientras que perdemos lo que pensábamos que era trabajo ya realizado, y como resultado caemos incluso más atrás del progreso que teníamos planificado. En el ejemplo, se asume que serán necesarias dos horas para deshacer el trabajo mal realizado, lo que lleva dos pasos atrás en el ámbito completo del proyecto. Incluyendo la hora que hemos pasado diagnosticando el problema, el tiempo total perdido en el proyecto es ahora de 1 + 2+ 2 = 5 horas.

Dado este impacto no planeado, pronto hemos consumido todo nuestro tiempo planificado, incluso si no encontramos otros problemas en el proyecto. Y si nuestro presupuesto estaba basado en el esfuerzo, habremos utilizado todo nuestro presupuesto también.

En nuestro ejemplo, alcanzamos el final planificado del proyecto, cuando solo el 50 % del proyecto ha sido completado – lo cual no es muy bueno.

El secreto para minimizar los impactos de los problemas es encontrarlo lo antes posible, Detectar los problemas pronto reduce potencialmente el tener que rehacer trabajo. Deberíamos entonces diagnosticar y resolver los problemas lo más rápidamente posible para no consumir más tiempo no planificado del necesario. Esto puede sonar muy básico, pero muchos proyectos fallan al enfrentarse a los problemas, pareciendo no darse cuenta del doble impacto que tienen.

En esta sección, veremos las siguientes áreas:

Practica	Herramienta/técnica	Conocimiento / Habilidad (nivel)
Identificar problemas	> Ciclo de tiempo > Defectos escapados	> Normas de proyecto y calidad (Nivel 1) > Modos de fallo y alternativas (Nivel 3) > Análisis de varianza y tendencia (Nivel 3) > Límites de control (Nivel 3)
Resolver problemas	> Integración continua > Riesgo basado en picos > Verificación y validación frecuentes. > Desarrollo basado en ensayos / desarrollo del primer ensayo > Desarrollo basado en el ensayo de aceptación.	> Resolución de problemas (Nivel 1)

Tabla 19: Áreas a discutir

2.6.1. Identificar problemas

El concepto que veremos en esta sección está centrado en encontrar problemas y preocupaciones. El diagnosticar el ciclo de tiempo y las tendencias puede apuntar a potenciales problemas antes de que ocurran, mientras que otros métodos, como los defectos escapados y los modos de fallo ayudan a identificar problemas que ya han ocurrido.

Las reuniones diarias, son también mecanismos importantes para la identificación de problemas. No olvidemos que la tercera pregunta de las reuniones diarias es si los miembros del equipo están experimentando algún problema, bloqueo, o impedimento para su progreso.

La razón por la que preguntamos esto es para intentar sacar a la luz problemas lo más pronto posible, en lugar de esperar hasta que el equipo venga con esos problemas a nosotros, o el cliente se queje, o tengamos una retrospectiva. Cuando las preocupaciones aparecen en las reuniones diarias, necesitamos investigar estos problemas y determinar si hay un problema gestándose.

2.6.1.1. Tiempo de ciclo

El ciclo de tiempo es una medida de cuanto lleva el tener las cosas hechas. El ciclo de tiempo va desde que el equipo comienza a trabajar en una parte del proyecto, como una historia de usuario, hasta que el elemento ha sido finalizado, aceptado y puede proporcionar valor al negocio.

El ciclo de tiempo se encuentra estrechamente relacionado con el trabajo en progreso (WIP). Durante el ciclo, cuando se ha iniciado un elemento pero aún no ha sido completado, se considera como trabajo en progreso. Como hemos visto en capítulos anteriores, en la entrega basada en el valor, el WIP excesivo se asocia con un cierto número de problemas, incluyendo:

- El WIP representa dinero invertido que aún no tiene retorno de la inversión.
- El WIP esconde los cuellos de botella en los procesos, y enmascara los problemas de eficiencia.
- El WIP también representa riesgo en forma de trabajos repetidos, ya que pueden haber cambios en los elementos hasta que hayan sido aceptados. SI hay un largo inventario de WIP, podría haber mucho trabajo a repetir si se requiere un cambio.

Es a causa de este tipo de riesgos por lo cual los enfoques ágiles y Lean animan a limitar el WIP. Para poder hacer esto, debemos prestar mucha atención al ciclo de tiempo, ya que ciclos de tiempo largos llevan a incrementar el WIP. Por lo tanto los enfoques ágiles deliberadamente rompen el trabajo del proyecto en pequeñas tandas y enfatizan la importancia de finalizar los elementos y obtener confirmación de que son aceptables, lo antes posible. Por ejemplo, los proyectos que toman un enfoque Scrum tienen pequeñas historias de usuario, realizan el trabajo en iteraciones de dos semanas, y animan a obtener la aceptación del usuario y el propietario de las historias de usuario al final de cada iteración.

Utilizando este tipo de prácticas (que son comunes a todos los métodos ágiles), vemos una reducción en el ciclo de trabajo de los proyectos ágiles en comparación con lo enfoques no ágiles. Por ejemplo, en un proyecto de desarrollo de software en cascada, los pasos de análisis, diseño, codificación, y unidad de testeo para una pieza de funcionalidad, a menudo llevan meses e incluso años. Esto representa un ciclo de trabajo muy largo, el cual a su vez resulta en unos niveles de WIP incrementados con sus riesgos y problemas asociados. Con los métodos ágiles, el objetivo es minimizar el ciclo de tiempo.

Para ayudar a mantener la atención en minimizar los ciclos de tiempo, muchos equipos escriben “Fecha de inicio” y “Fecha de fin” en tarjetas Kanban o pizarras de tareas.

El tiempo transcurrido entre las fechas de inicio y final es el ciclo de tiempo del elemento de trabajo. El ciclo de tiempo del proyecto es la media de los ciclos de tiempo de estos elementos. Entonces tenemos que el rendimiento, el cual es la cantidad de resultados de un proceso (en otras palabras, la cantidad de trabajo que el equipo de proyecto es capaz de realizar). La relación de estas tres variables de rendimiento, WIP, y ciclo de tiempo está indicada en la siguiente fórmula:

Ciclo de tiempo = WIP / rendimiento.

Conocer el rendimiento del equipo nos permite predecir las capacidades futuras sin la necesidad específica de tener que conocer que se le ha pedido al equipo que haga. Conocer el ciclo de tiempo nos permite realizar compromisos más fiables con el cliente o la organización sobre cuánto tiempo nos llevará entregar el trabajo. El WIP mide cuanto trabajo tenemos en “depósito” y nos da una visión interna de los problemas, cuellos de botella en los procesos, y los riesgos relacionados con tener que repetir trabajos ya hechos.

Únicamente hemos visto los ciclos de tiempo para elementos de trabajo nuevos, pero el ciclo de tiempo también es útil cuando analizamos defectos y sus soluciones. El gráfico de coste de la figura 54, explica que cuanto más se tarda en solucionar un defecto, más caro es solucionarlo.

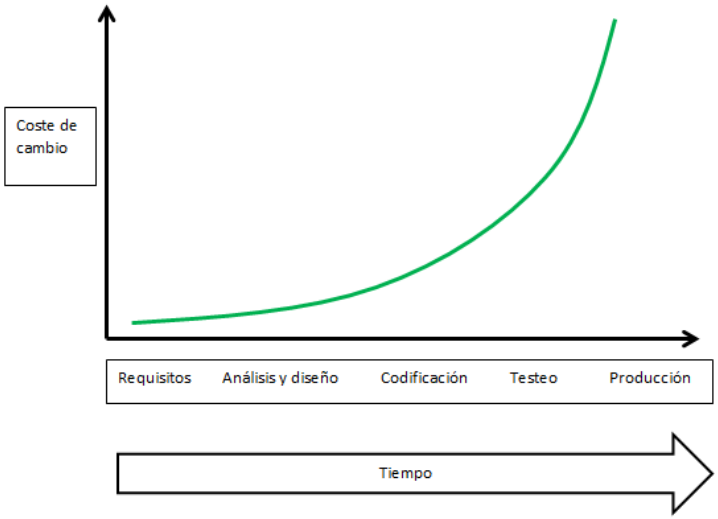


Figura 54: gráfico de coste. (Fuente: Elaboración propia)

Hay un número de razones para este incremento de coste, incluyendo:

- Mucho del trabajo que habremos hecho después del mal diseño, resultará en trabajo a deshacer.
- Cuanto más tarde se incluya en el ciclo de desarrollo, más stakeholders resultarán afectados por el defecto y más caro será repararlo.

Las técnicas ágiles de programación de pares, integración continua, y desarrollo basado en test (para proyectos de software), así como la participación activa de stakeholders a través de técnicas como las demostraciones y las revisiones del cliente, están todas enfocadas a encontrar defectos y cambiar las peticiones lo más rápido posible, antes de que los costes se eleven demasiado en la curva de coste.

Por lo tanto es ventajoso encontrar los cambios y defectos lo antes posible para minimizar el tener que rehacer trabajos y reducir los costes. Aquí es donde entra el ciclo de tiempo de defecto. El ciclo de tiempo de defecto es la cantidad de tiempo entre la introducción del defecto (cuando el defecto se incluye accidentalmente) y la solución del defecto (cuando el defecto es reparado).

La duración del ciclo de tiempo de defecto dicta como de lejos llegará el defecto en el gráfico de coste de cambio.

Para ayudar a minimizar el coste de las reparaciones, algunos equipos de proyecto rastrean activamente el ciclo de tiempo de defecto y fijan objetivos para una rápida resolución, como se indica en la figura 55:

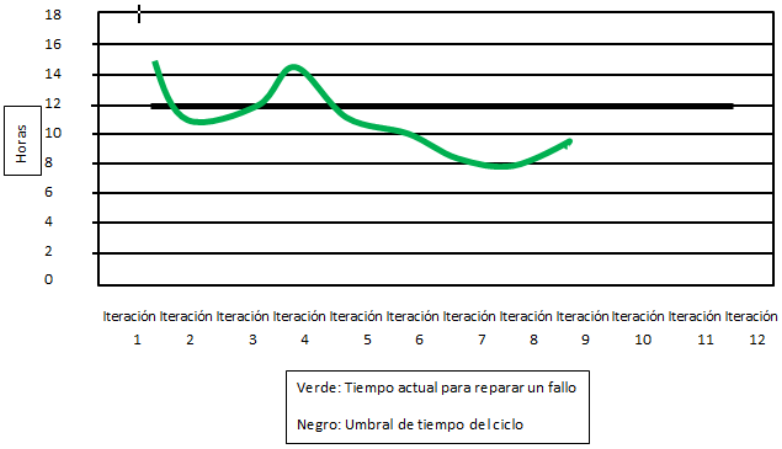


Figura: 55: Gráfico de tiempo de ciclo. (Fuente: Elaboración propia)

En este gráfico, podemos ver que el umbral del ciclo de tiempo es de 12 horas. Esto significa que, en media, el objetivo del equipo es tener reparados todos los defectos reportados en 1.5 días de trabajo. Por lo tanto si un defecto se encuentra el martes por la tarde, debería estar reparado al final del miércoles. Mirando el gráfico, podemos ver que para las primeras 5 iteraciones, la media del ciclo de tiempo está típicamente por encima de las 12 horas. En las siguientes iteraciones, no obstante, la media del ciclo de tiempo está por debajo de la media, bajando hasta las 8 horas.

Rastreando el ciclo de tiempo para crear nuevo trabajo y WIP, junto con el ciclo de tiempo, podemos minimizar tanto el potencial de tener que repetir trabajos como el coste de ello, cuando sea requerido en el proyecto.

2.6.1.2. Defectos escapados

Hemos hablado sobre investigar las preocupaciones que aparecen durante las reuniones diarias para descubrir los problemas pronto y la importancia de reparar los defectos encontrados lo antes posible, pero ¿qué pasa con los defectos que no se descubren durante los procesos de testeo y validación y llegan al cliente? Esto es lo que se llama Defectos escapados. Son los más costosos de reparar y están en lo más alto del gráfico de costes de cambios.

Desafortunadamente, por mucho que nos esforcemos en prevenirlos, un defecto ocasional puede colarse a través de todos los test y procesos de calidad en el producto final.

Cuando descubrimos un defecto escapado, obviamente es muy tarde para prevenirlo, pero podemos utilizar esta información para mejorar nuestros procesos yendo más allá y esperar reducir el número de defectos escapados en el futuro. El indicador de defectos escapados encontrados, contabiliza el número de defectos escapados encontrados en un periodo de tiempo (días, semanas o meses). En el dominio del desarrollo de software, por ejemplo, los proyectos acostumbran a rastrear los defectos escapados hasta la entrega en la que se originaron. Este rastreo ayuda a los equipos a evaluar y mejorar la efectividad de los procesos de aseguramiento de la calidad y control de calidad en el tiempo. Los resultados se muestran normalmente en formato gráfico, como vemos en la figura 56.

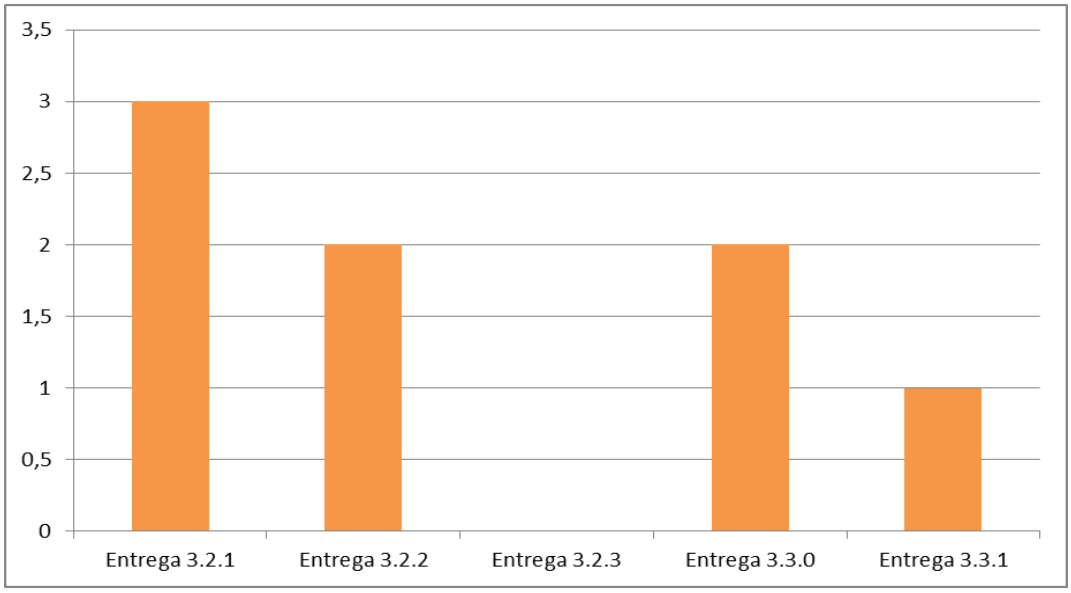


Figura 56: Número de defectos escapados encontrados por entrega. (Fuente: Elaboración propia)

Además de para rastrear el número de defectos escapados por entrega, estos gráficos pueden ser utilizados para mostrar las tendencias en el tiempo, como por ejemplo en meses, como vemos en la figura 57.

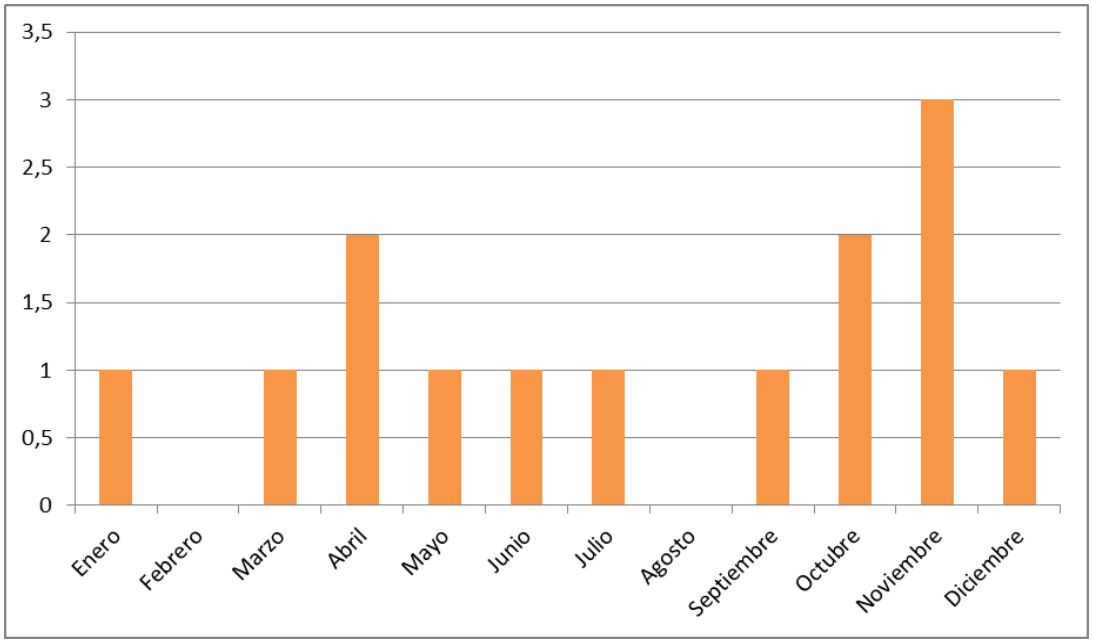


Figura 57: Numero de defectos escapados encontrados por mes. (Fuente: Elaboración propia)

2.6.1.3. Normas de proyecto y Calidad

La detección y resolución de problemas está cercanamente relacionada con la gestión de la calidad. Los procesos de prueba y otras prácticas que ponemos en práctica para encontrar defectos son parte de los esfuerzos en aseguramiento de la calidad y del control de calidad de nuestros proyectos.

La pregunta sería, ¿qué debe hacer el equipo para asegurar la calidad y el valor del producto? Estas normas pueden ser guardadas o referenciadas en la carta del proyecto, el equipo de desarrollo puede simplemente adoptar estas prácticas como normas de trabajo.

Las normas y prácticas de calidad pueden incluir cosas como:

- Medir la calidad del producto por ensayos superados y por la aceptación del cliente.
- Automatizar tantos ensayos como sea posible.
- Asegurarse de que la fase de ensayos se lleva a cabo como una parte más de cada iteración.
- Intentar solucionar como mínimo el 90 por ciento de los defectos encontrados dentro de la siguiente iteración.
- Incentivar a los representantes del control y el aseguramiento de la calidad para que trabajen con los desarrolladores y los representantes del negocio para entender el criterio de aceptación de cada característica.
- Solo clasificar los defectos como reparados cuando los representantes del negocio, no los desarrolladores, digan que están hechos.
- Asegurar que los probadores colaboran con los desarrolladores en los defectos encontrados y en los pasos para reproducir el defecto encontrado.

También necesitamos monitorizar y evaluar la calidad de los procesos de nuestro proyecto utilizando herramientas como los indicadores de defectos, análisis de varianza y de tendencia, y análisis de las causas de raíz, y tomar acciones para corregir cualquier problema que encontremos.

De manera breve, los indicadores de calidad y del proyecto se refieren a las guías de calidad y aceptación escritas o no escritas (tácitas) de las que el equipo se haga responsable. Estas guías pueden estar escritas como normas formales y monitorizadas a través de auditorías, o como es más típico para equipos pequeños, pueden estar desarrolladas a través de discusiones y reforzadas a través de revisiones y retrospectivas.

2.6.1.4. Modos de fallo y alternativas

Los conceptos que veremos ahora como parte de los modos de fallo y alternativas viene de Alistair Cockburn. Estas ideas tienen que ver con el lado humano del rendimiento y del proceso. Describen algunas verdades sobre las personas que explican porque, aunque sepamos qué es lo que deberíamos estar haciendo, a menudo nos comportamos de manera diferente.

Los cinco modos de fallo que Alistair describe en su libro, “Desarrollo ágil de software: El juego cooperativo”, son:

1. Cometer errores.
2. Preferir fallar de forma conservadora.
3. Inventar en lugar de investigar.
4. Ser criaturas de hábitos.
5. Ser inconsistentes.

Veamos rápidamente cada uno de estos modos de fallo para entender el problema.

1. **Cometer errores:** No es una sorpresa que las personas comenten errores. Ésta es una de las principales razones por las que se creó el desarrollo iterativo e incremental – para reconocer el hecho de que los fallos ocurren y para proveer de mecanismos para recuperarse y rápidamente superar este hecho.

2. **Preferir fallar de forma conservadora:** Cuando se enfrentan a la incertidumbre, las personas tienden a ir hacia lo conocido, incluso cuando son conscientes de que puede no ser el enfoque más óptimo a tomar. Como ejemplo de lo que se acaba de mencionar, en un ambiente ágil, si los gestores notan que el proyecto comienza a ir mal y encuentran problemas de manera frecuente, pueden inconsciente o conscientemente, volver a utilizar métodos de gestión más tradicionales (no ágiles) para llevar a cabo el proyecto. Como estamos hablando de personas, hay obviamente excepciones a esta regla. Algunas personas sienten que no tienen nada que perder con un nuevo enfoque. Hay maneras de avanzar en un proyecto para las dos maneras de pensar – volver al enfoque familiar o abrazar el nuevo enfoque.- pero cuando las personas con puntos de vista opuestos están en el mismo proyecto, pueden presentarse problemas.

3. **Inventar en lugar de investigar:** Esto describe la tendencia que tienen muchas personas (especialmente los ingenieros) de inventar nuevas formas de hacer las cosas, en lugar de buscar opciones disponibles y reutilizarlas. Esta tendencia puede ser un producto de los sistemas educacionales que recompensan el pensamiento individual y la experimentación científica, combinada con la satisfacción intelectual que obtenemos de resolver los problemas. No obstante, mientras que puede ser más divertido inventar en lugar de investigar, también suele ser más costoso, llevar más tiempo y ser más propenso a errores.

4. **Resistencia al cambio:** Las personas se resisten a los cambios, por lo que conseguir cambiar la forma en que se hacen las cosas será siempre difícil. A menudo, incluso cuando se sabe que hay enfoques mejores, no se adoptan ya que a algún nivel (consciente o inconsciente) el cambio es poco atractivo.

5. **Inconsistencia:** Finalmente, muchas personas son muy inconsistentes a la hora de seguir un proceso. Por lo tanto el desafío no es únicamente buscar mejores maneras de hacer las cosas, es conseguir que las personas acepten las nuevas maneras, cambien sus propios enfoques, y entonces apliquen los nuevos enfoques de forma consistente. Como Karl Wiegers resumió, “No estamos faltos de prácticas, estamos faltos en práctica”.

Hay también rasgos y comportamientos humanos que ayudan a contrarrestar estos fallos comunes. Estos modos de éxito pueden ser muy útiles en los proyectos. Incluyen:

1. **Ser bueno mirando alrededor:** Se refiere a la habilidad de las personas para observar, revisar, y darse cuenta cuando las cosas no van bien.

2. **Ser capaz de aprender:** Después de ver que está mal, encontramos maneras de repararlo y aumentamos nuestras habilidades y conocimientos a lo largo del camino.

3. **Ser maleables:** Esta es la habilidad de cambiar y aceptar las nuevas ideas y enfoques.

4. **Estar orgullosos del trabajo:** Somos capaces de dar un paso fuera de nuestras descripciones de trabajo para reportar o reparar un problema, ya que esto es lo correcto para el proyecto.

Basándose en estos modos de éxito, Cockburn sugiere algunas estrategias para superar los cinco modos de fallo:

1. **Luchando con disciplina y tolerancia:** Esta estrategia implica establecer una manera estándar de hacer las cosas y de animar a la gente a adoptarlas, mientras que también construye cierta capacidad para la tolerancia y el perdón dentro del enfoque. Entonces, si las personas difieren de la norma, centramos nuestra energía en restablecer esta práctica, con tal de que no necesite modificarse

2. **Empezar con algo tangible y concreto:** Resolvemos primero los problemas en nuestras mentes y después en la realidad. Para ayudar a las personas a superar su resistencia a un cambio o solución, podemos crear cosas que sean concretas y tangibles para representar la solución final. Por ejemplo, podemos crear maquetas de baja fidelidad que no solo expongan como fluirá la pantalla en un sistema informático, sino que además invite a tocar e interaccionar a los stakeholders para reordenar las pantallas. Esta interacción transforma un modelo abstracto en un flujo de trabajo tangible.

3. **Copiar y alterar:** Es a menudo más fácil modificar un diseño de trabajo para que encaje en nuestras necesidades que crear algo de la nada. “El síndrome de la página en blanco” es la parálisis que puede aparecer cuando tenemos que crear una nueva solución. Si en su lugar tomamos en enfoque de alterar una solución ya existente, tendremos un punto desde el que empezar.

4. **Observar y escuchar:** Aprendemos observando y escuchando a los demás. Algo tan simple como poner a un equipo de miembros junior con posibilidad de ver y escuchar a un equipo más experimentado se ha demostrado que mejora el rendimiento del equipo de miembros junior, incluso sin ninguna formación más.

5. **Apoyando la concentración y la comunicación:** Los trabajos de conocimiento requieren la dicotomía de ambos, tanto concentración como comunicación. En el desarrollo de software, por ejemplo, los programadores necesitan una cantidad suficiente de tiempo para entrar en el modo silencioso y productivo conocido como “flow”. Pueden gastar 20 minutos en entrar en este estado, solo por haber sido interrumpidos por un minuto o dos preguntas. No obstante, los intercambios de información y conversaciones son esenciales para los trabajadores de conocimiento y no deben ser evitadas. En lugar de eso, debemos animar este tipo de intercambios, ya que ayudan a tapar los huecos en la comprensión y dan solución a las preguntas.

La clave está en gestionar las comunicaciones con sabiduría. Como hemos visto en capítulos anteriores, la técnica XP recomienda el modelo conocido “cuevas y lugares comunes”. Las cuevas son áreas en silencio donde las personas pueden retirarse para realizar trabajo en silencio. Lugares comunes son áreas comunes de trabajo en las cuales se intercambian las ideas. El libro “Peopleware” de Tom Demarco y Anthony Lister, sugieren establecer un periodo de trabajo en silencio cada día por un tiempo especificado (p.e. 2 horas). Durante este periodo en silencio, las reuniones y llamadas de teléfono no están permitidas, para que las personas puedan centrarse y realizar el trabajo.

6. **Asignaciones del trabajo de acuerdo a la personalidad:** Debemos intentar emparejar las asignaciones de trabajo con el tipo de personalidad y vigilar los desajustes. Por ejemplo, las personas que prefieren cambiar código defectuoso en lugar de explicar por qué es defectuoso probablemente no sean los mejores profesores. Necesitamos jugar con los rangos de personalidad de las personas (orientado al detalle, sociales, analíticos, que den soporte, etc) cuando asignamos los roles.

7. **Talento:** Con trabajadores expertos, como los trabajadores de conocimientos, la diferencia entre la peor clase, la media y la mejor clase es muy grande. Necesitamos reconocer la gran diferencia que juega el talento y encontrar maneras de atraer y retener el mejor talento. Parece una frase obvia, ¿pero por qué muchas organizaciones la ignoran? Tal vez es porque es más fácil medir el impacto a corto plazo de un 5 por ciento de caída en los salarios de la parte de abajo que medir el desconocido impacto a largo plazo de perder a nuestra mejor gente.

8. **Recompensas que preserven la diversión:** Las estructuras de recompensa son una materia difícil; una vez las personas empiezan a esperar las recompensas, o si éstas son eliminadas, tiene un efecto muy negativo. Como ejemplo, recompensar a los niños con regalos por leer puede parecer una buena idea al principio para ayudarles a comenzar, pero esta táctica se nos puede volver en contra y hacer que el niño lea menos una vez que el sistema de recompensa se retire o que la recompensa ya no le parezca valiosa.

Por lo tanto necesitamos dar recompensas que sean provechosas a largo plazo, motiven la autoestima, como el orgullo por el trabajo, orgullo por cumplir y por contribuir. Estas recompensas tienen una atracción a largo plazo que no se desvanece en el tiempo.

9. **Combinando recompensas:** Las mejores estructuras de recompensa combinan elementos para realizar no solo un paquete convincente, sino verdaderamente solidario, de cuidado ambiente de trabajo en el que los objetivos personales y de la compañía están alineados. Un único sistema de recompensas no funcionará para todo el mundo, por lo que debemos construir diferentes tipos de sistemas.

10. **Feedback:** Un poco de feedback puede reemplazar mucho trabajo analítico. Los métodos ágiles incorporan el feedback a través de muchas de sus prácticas. Por ejemplo, La práctica XP tiene dos personas trabajando juntas, ofreciéndose feedback la una a la otra de manera inmediata. La integración continua provee de feedback rápido, y las revisiones de iteración proveen de feedback frecuente.

2.6.1.5. Análisis de varianza y tendencia

La varianza es la medida de como de apartadas están las cosas (o cuanto varían las cosas entre ellas). Por ejemplo, hay varianza en las estimaciones cuando múltiples personas estiman la misma cosa, y hay varianza entre las estimaciones y los resultados actuales, una vez que el trabajo ha sido realizado. La varianza es normal y debe ser esperada, pero necesitamos aprender cómo llevarla dentro de los límites aceptables. Y si la varianza fluctúa más allá de los límites aceptables, necesitamos saber cómo actuar.

Veamos un ejemplo de varianza que no está en el examen pero ayuda a ilustrar este concepto. Los expertos en software Lean Mary y Tom Poppendieck explican una historia sobre varianza en formación de pilotos. Los pilotos son evaluados en su rendimiento en el entrenamiento de vuelo, y los resultados son seguidos y graficados.

Cuando los pilotos rinden mal, se les grita y se les dice que mejoren. Cuando los pilotos lo hacen bien, son felicitados y alabados. Lo que se observa es que después de ser reprendidos, los pilotos que rendían mal mejoran en los siguientes vuelos, mientras que los pilotos que más rendían generalmente lo hacen peor en los siguientes vuelos.

Una conclusión que se puede sacar de estos resultados es que gritarles a las personas es una manera muy efectiva de mejorar su rendimiento y que el griterío debe aumentar, mientras que el alabar y felicitar no es efectivo y debe detenerse. Esto podría ser una conclusión incorrecta, no obstante. Lo que realmente se está observando es la variación real. Algunas cosas van bien, mientras que otras van mal. Dentro de cada conjunto de datos, los bajos extremos probablemente serán seguidos por un valor más alto, igual que los extremos más altos probablemente irán seguidos de un valor más bajo. Por lo tanto lo que debemos aprender de este escenario es que los procesos simplemente varían, y debemos aceptar un cierto grado de varianza en un proceso real.

El experto en calidad W. Edwards clasifica la varianza en variación de causas comunes y variación de causas especiales. La varianza de causas comunes se refiere a la media diaria de diferencias a la hora de realizar el trabajo, y la varianza de causas especiales se refiere a los grandes grados de varianza debidos a factores especiales o nuevos.

Por ejemplo, si se nos da el trabajo de clavar clavos en madera durante todo el día, algunos clavos entrarán rectos, mientras que otros lo harán con cierto ángulo, ésta es una causa común de varianza. No obstante, si alguien apaga las luces, la varianza probablemente será mucho mayor, ya que el ambiente ha cambiado (causas especiales de variación). Deming va más allá al decir que hay dos errores clásicos que los managers hacen:

Error 1: Reaccionar a un resultado como si fuese resultado de una causa especial, cuando en realidad proviene de causas comunes de variación.

Error 2: Tratar un resultado como si fuese resultado de causas comunes de variación, cuando realmente proviene de causas especiales.

Un ejemplo de error 1 podría ser investigar porque historias que se estimaron durarían cuatro días de trabajo realmente llevaron cinco días para completarse. Un ejemplo de error 2 podría ser permitir que el proyecto de clavar clavos continúe con las luces apagadas después de haber detectado un problema potencial.

El punto a comprender aquí es que debemos aceptar las causas de variación común en los proyectos, mientras que debemos estar vigilantes para intervenir en causas especiales de variación. En otras palabras, deberíamos evitar la microgestión y en su lugar trabajar para eliminar los cuellos de botella y los impedimentos. Por ejemplo, preguntar a los desarrolladores por qué no han codificado cinco de las características esta semana cuando la pasada si lo hicieron es errar en aceptar las causas de variación común; este tipo de cosas varían únicamente un poco y no son perfectamente lineales o predecibles.

Por lo tanto, en los proyectos ágiles en particular, centrar mucho esfuerzo en seguir la conformidad con un plan rígido es probablemente no la mejor manera de utilizar el tiempo del proyecto. En su lugar, debemos observar los indicadores externos y a la reunión diaria en la que el equipo reporta los problemas, impedimentos o bloqueos a su trabajo para ver si existen problemas especiales que necesiten ser resueltos.

Medidas Anticipadas contra Demoradas.

El análisis de tendencias es un aspecto particularmente importante para los gestores de proyecto, ya que proporciona información de problemas futuros, antes de que ocurran. Aunque medidas como la cantidad de presupuesto consumido, por ejemplo, continúan siendo importantes, estas medidas son indicadores demorados; en otras palabras, dan una vista de algo que ya ha ocurrido. Mientras que los indicadores Demorados que proveen de una visión perfecta del pasado pueden ser muy interesantes para los contables, los indicadores Anticipados que proveen visiones del futuro, aun siendo imperfectas, son muy interesantes para los gestores de proyecto. Esto es así porque los indicadores Anticipados proveen información sobre lo que está ocurriendo o sobre lo que va a comenzar a ocurrir en el proyecto. La indicación anticipada de un problema potencial es más útil que los indicadores demorados, ya que ayudan al equipo a adaptarse y replanificar apropiadamente.

Otro punto a tener en cuenta es que la identificación de tendencias es más importante que los valores de datos actuales.

2.6.1.6. Límites de control

Los límites de control en un contexto ágil tienen una interpretación débil, que incluye niveles de tolerancia y señales de alerta. Los límites de control nos ayudan a diagnosticar problemas antes de que ocurran, y nos proveen de guías para operar dentro de ellos. Algunas de las recomendaciones o reglas de oro ágiles, como los equipos de 12 o menos miembros, pueden interpretarse como límites de control.

Una manera en la que podemos utilizar estos límites es para monitorizar la velocidad para medir como de probable es que podamos completar un trabajo en la fecha acordada. Por ejemplo, si nos quedan 600 puntos de funcionalidad en la cartera y 10 meses antes de que comiencen los preparativos del despliegue, deberíamos establecer nuestros límites de control en 60 puntos por mes, la cual es la velocidad necesaria para alcanzar la meta.

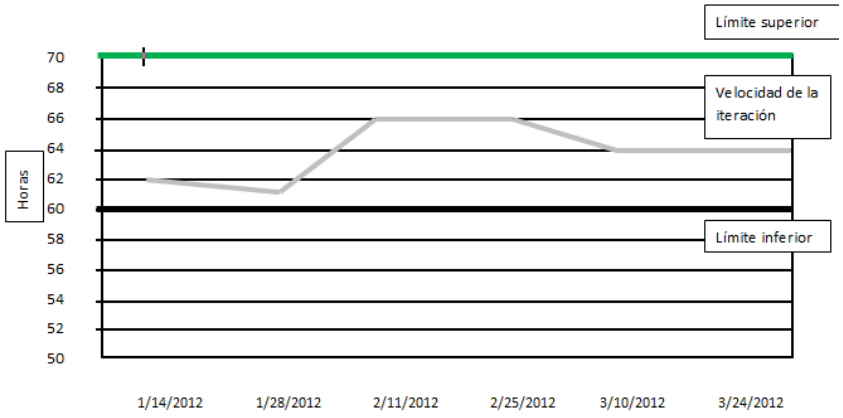


Figura 58: Límites de control de velocidad. (Fuente: Elaboración propia)

En la figura 58, tenemos un límite de control inferior en 60 puntos por iteración. Si la velocidad cae por debajo de este límite, existirá la posibilidad de que la funcionalidad acordada no esté desarrollada a tiempo. Obviamente, una pequeña caída por debajo de los 60 puntos no es tan mala como un periodo sostenido de caídas; lo que más nos debería preocupar es la media, que en estos gráficos es fácil de interpretar.

Las tablas Kanban y de tareas que limitan el WIP son también una forma de límites de control en los proyectos ágiles. Estas herramientas previenen de introducir demasiado trabajo en una actividad, y ayudan al equipo a controlar la cantidad de trabajo en progreso.

2.6.2. Resolviendo problemas

En la primera parte de este capítulo hemos visto cómo identificar los problemas. Ahora veremos cómo resolver esos problemas una vez han sido identificados. Veremos las técnicas ágiles designadas para resolver rápidamente los problemas antes de que se hagan más grandes.

Las técnicas de resolución de problemas utilizadas en los métodos ágiles, animan a abordar los problemas antes de que éste se eleve demasiado en la curva de gasto de cambio.

2.6.2.1. Integración continua

La integración continua es un proceso de desarrollo de software. Utilizando esta técnica, el equipo integra frecuentemente un código nuevo y modificado en el repositorio del código del proyecto. Esta práctica ayuda a minimizar los problemas de integración que resultan de múltiples personas realizando cambios incompatibles al mismo código base. Típicamente, cuanto más frecuentemente se realizan estos compromisos al código, más pequeña es la cantidad de código que necesita ser cambiada para permitir una nueva construcción (o versión) del software que compile correctamente. La integración continua es un ejemplo de maneras en las que los proyectos ágiles se enfrentan a los problemas lo antes posible. Aunque estemos hablando solo en términos de software, el concepto general se puede aplicar a cualquier otro trabajo de trabajadores de conocimiento en el que múltiples personas estén desarrollando piezas separadas del proyecto y uniendo esas piezas para crear un producto funcional y valioso.

A continuación vemos los componentes de un sistema de integración continua:

- **Sistema de control del código fuente:** Éste es el software que realiza el control de versión de todos los ficheros que se representan en la compilación, o versión del producto, que está siendo desarrollada. Una vez que el equipo ha completado algunos cambios al código fuente, deben comprobar la nueva versión de código dentro del sistema de control de código fuente. Un sistema de control del código fuente utilizado comúnmente es Apache Subversion.

- **Herramientas de construcción:** El código fuente necesitará ser compilado antes de que los tests puedan ser ejecutados. Muchos ambientes integrados de desarrollo (IDEs) como Microsoft Visual Studio o Eclipse pueden servir como herramienta de construcción para compilar el código.

- **Herramientas de test:** Como parte del proceso de construcción, se ejecutan unidades de test para asegurar que la funcionalidad básica opera según lo planeado. Las herramientas de unidad de test, como NUnit (para Microsoft.Net) y JUnit (para Java), ejecutan los tests pequeños escritos en estas herramientas para comprobar el código contra cambios no anticipados en su comportamiento.

- **Programador o desencadenador:** Las construcciones deben ser lanzadas en una programación (p.e. cada hora) o cada vez que se detecte un cambio en el código fuente (p.e. desencadenado por el cambio). El programador o desencadenador, es el componente que inicia el proceso de integración.

- **Notificaciones:** Si una construcción falla, el equipo necesita ser notificado para que puedan corregir la construcción lo antes posible. Las notificaciones se envían generalmente por e-mail, pero en muchas de las herramientas actuales pueden también realizarlas por mensajería instantánea y con notificaciones de Twitter.

Para los propósitos del examen, se debe comprender que la integración continua utiliza herramientas automatizadas para comenzar el proceso de integración cuando el código se comprueba o en intervalos temporizados. Además para comprobar que el código nuevo o cambiado compila correctamente, existen unidades automatizadas de test para asegurar que el sistema sigue rindiendo como se esperaba después de integrar el nuevo código.

La integración continua es importante para los proyectos ágiles ya que provee los siguientes beneficios:

- El equipo recibe una advertencia rápida de código roto (inválido), conflictivo, o incompatible.

- Los problemas de integración se reparan tal y como ocurren, con el enfoque de fecha de lanzamiento. Esto desplaza cualquier cambio a la parte baja de la curva de coste de cambio y evita el trabajo de última hora antes del lanzamiento.

- El equipo recibe feddback inmediato del impacto global del código que están escribiendo.

- Esta técnica asegura la comprobación frecuente del código y alerta al equipo de problemas más pronto que tarde.

- Si se encuentra un problema, el código puede restablecerse hasta el último estado conocido libre de fallos para su comprobación, demostración, u otros propósitos de lanzamiento.

Las desventajas o costes de utilizar la integración continua son:

- El tiempo de preparación necesario para establecer un servidor de construcciones y configurar el software de integración continua: esto se realiza típicamente como una iteración de actividad 0 (antes de que dé comienzo el trabajo de desarrollo en el proyecto).

- El coste de adquisición de una máquina para que actúe como servidor, ya que normalmente se trata de una máquina dedicada.

- El tiempo necesario para construir un conjunto de tests automatizado y comprensible cuando se comprueba el código.

A pesar de los costes, la integración continua está establecida como una buena práctica estándar para el desarrollo de proyectos de software, y muchos equipos multi-persona utilizan este enfoque. Los beneficios de esta tecnología sobre las viejas prácticas son significativos; con la integración continua, muchas más pruebas se llevan a cabo por día, y pasa menos tiempo antes de que un problema sea identificado.

2.6.2.2. Puntas basadas en riesgo

Una punta basada en riesgo es una prueba corta de concepto que el equipo lleva a cabo para investigar un problema. Como ejemplo, imaginemos que estamos construyendo una unidad de producción de contenedores basados en algas para capturar emisiones de dióxido de carbono. Esta unidad de producción pide iluminación subacuática utilizando partes baratas disponibles. En este escenario, debemos llevar a cabo una punta para comprobar las opciones de protección contra el agua de unos tubos fluorescentes. Testeando varias opciones, realizando experimentos a baja escala, podemos reducir los riesgos del proyecto asociados a un componente clave del éxito del proyecto (desarrollar iluminación subacuática que esté bien de precio) pronto y al mínimo coste.

Este enfoque de realizar experimentos cortos para investigar las porciones de riesgo del proyecto se encuentra en el corazón de las puntas basadas en riesgo. Si el ejercicio de concepto de protección es exitoso, podemos eliminar el riesgo, y el perfil de riesgo general del proyecto se verá reducido.

Si un ejercicio de concepto de protección no es exitoso, no obstante, podemos probar diferentes enfoques. Si ninguno de los enfoques que intentemos es exitoso, alcanzaremos una condición conocida como “fallo rápido”. Aunque suene mal, el fallo rápido puede de alguna manera, ser un buen resultado; en lugar de continuar en un proyecto que podría fallar, los fondos y recursos restantes pueden ser redirigidos a otros proyectos que esperen recursos.

Las compañías de investigación y desarrollo con un modelo de negocio que se basa en el desarrollo de nuevos productos, reconocen que un cierto número de sus proyectos pueden no tener éxito. Generalmente operan con la filosofía de que si algo es fácil, otras compañías ya lo habrían hecho, y en cambio ellas buscan progresar en nuevos enfoques. En este ambiente, los fallos rápidos reducen mucho la cantidad de los costes en proyectos que no son viables y permiten a la compañía intentar más enfoques durante cada año, incrementando las posibilidades de encontrar un nuevo producto revolucionario.

Las puntas basadas en riesgo se utilizan a menudo en los proyectos de software para testear tecnologías nuevas o poco conocidas al inicio del proyecto antes de avanzar en exceso con el desarrollo. Por ejemplo, imaginemos que el análisis del equipo de una característica de prioridad media lleva a la pregunta de “¿Podemos interactuar con el viejo sistema desde un dispositivo móvil?” El equipo puede investigar este tema a través de una punta basada en riesgo antes de continuar demasiado lejos en el camino de analizar, diseñar, y desarrollar la característica.

2.6.2.3. Verificación y validación frecuentes

El tema de la verificación y validación frecuentes se practica a muchos niveles en los proyectos ágiles. Es un efectivo antídoto, tanto para el hecho de que cometer errores es parte de ser humanos como para la discrepancia en las expectativas que surge cuando escuchamos la descripción de alguien de un objetivo final e interpretamos su descripción de manera diferente. Con la validación y la verificación frecuentes, nos estamos asegurando que las cosas están funcionando y progresando como deberían, y estamos atentos a cualquier discrepancia en las expectativas. Haciendo esto, habilitamos que el proyecto converja rápidamente en la solución real, incluso cuando esta solución cambia respecto a los requisitos establecidos inicialmente.

La siguiente figura es una variación de un gráfico común en XP que se utiliza típicamente para mostrar la planificación frecuente y los lazos de feedback. No obstante, este grafico también es válido para ilustrar como la verificación y validación frecuente que se lleva a cabo en los proyectos ágiles.

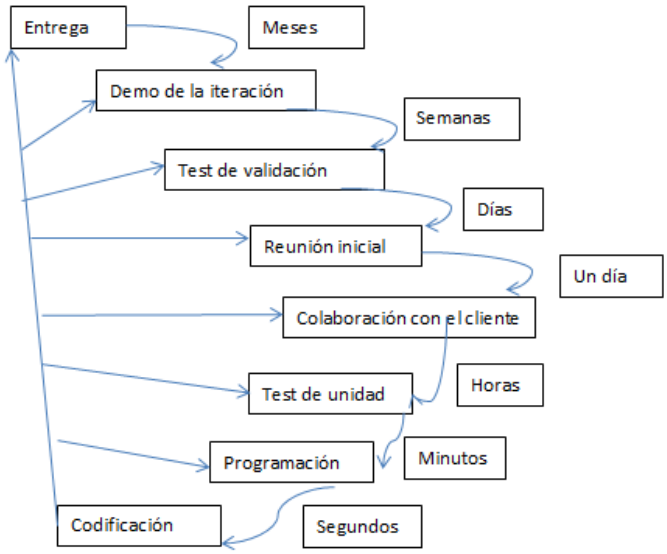


Figura 59: Verificación y validación frecuente en proyectos ágiles. (Fuente: Elaboración propia)

En la figura 59, podemos ver como la práctica de software de la programación por pareja ofrece ciclos de verificación y validación muy cortos (unos pocos segundos). Si en una de las parejas una de las personas detecta un error o malentendido de la intención deseada, el problema se resuelve inmediatamente, ya que los dos desarrolladores discuten el código que están escribiendo.

En el siguiente nivel, vemos que las unidades de test se llevan a cabo cada pocos minutos por los desarrolladores para verificar que el código está alcanzando el resultado deseado y para validar si la funcionalidad existente continua trabajando. La colaboración del cliente (el siguiente nivel) debe ocurrir también frecuentemente. Esta colaboración provee de feedback de si los diseños son correctos y los objetivos se están cumpliendo. A través de las reuniones diarias, podemos validar quien está trabajando en que, y cuando las cosas estarán acabadas y preparadas para ser integradas. Entonces, cada par de semanas, realizamos una demostración del producto y una revisión de la iteración para verificar que estamos construyendo el producto correcto y para validar la corrección de la solución hasta el momento. Por lo tanto, tenemos ciclos dentro de ciclos todos ellos interesados en la verificación y validación frecuentes en el proyecto.

A causa de que los métodos ágiles aceptan que construir productos nuevos y a menudo intangibles puede conllevar desafíos en la comunicación, existen múltiples niveles deliberados de actividades de comprobación y confirmación para asegurarse que encontramos los problemas lo antes posible. Los proyectos ágiles de software automatizan muchos de estos tests para que sean ejecutados más frecuentemente con costes más bajos. De nuevo, la idea detrás de la verificación y validación frecuente está centrada en encontrar los problemas lo antes posibles y mantenerlos lo más abajo posible de la curva de coste de cambio.

2.6.2.4. Desarrollo basado en el test (TDD) / Desarrollo basado en el primer test (TFD)

El desarrollo basado en el test (TDD) y el desarrollo basado en el primer test (TFD) son ambas técnicas derivadas de la industria del desarrollo de software. TDD y TFD fomentan el aportar una mentalidad diferente al desarrollo y emplean ciclos cortos de test y ciclos de feedback. Existen diferencias entre TDD y TFD, pero para los propósitos del examen, simplemente es necesario comprender a alto nivel, los comportamientos que estas técnicas fomentan y porque son beneficiosas. Por esta razón, examinaremos estas técnicas al mismo tiempo, bajo la cabecera de Desarrollo basado en test, o TDD.

La filosofía detrás del TDD es que los test deben ser escritos antes de que se escriba el código. En otras palabras, los desarrolladores deberían pensar primero en cómo se debería testear la funcionalidad y entonces escribir los tests en un lenguaje de unidad de testeo (como NUnit o JUnit) antes de que realmente se empiece a desarrollar el código. Inicialmente los tests pueden fallar, ya que los desarrolladores no habrán escrito aún el código para proporcionar la funcionalidad requerida.

Por lo tanto, con TDD, los desarrolladores comienzan un ciclo de escribir código y hacer los test hasta que el código los supere todos. Entonces, si es necesario, limpian el diseño para hacerlo más fácil de entender y mantener sin cambios en el comportamiento del código. Este último proceso se llama “Refactorización”.

Rojo, verde, refactorización / Rojo, verde, limpio.

Como hemos visto antes, los desarrolladores normalmente utilizan las herramientas de integración continua para ejecutar las construcciones. Estas herramientas notifican a los desarrolladores si el código pasa o falla comunicando un estatus verde o rojo.

El proceso de escribir un código que inicialmente falla, añadiendo código hasta que pase el test, y entonces refactorizando el código se conoce como “Rojo, Verde, refactorización” o a veces “Rojo, Verde, Limpio”.

A continuación se exponen algunos de los beneficios principales de asociar estrechamente el código a los test de esta manera:

- Al centrarnos primero en los tests, debemos pensar como la funcionalidad puede ser utilizada por el cliente. Esto nos hace pensar en preocuparnos con el resultado antes de las implementaciones, lo cual según los estudios, lleva a mejores diseños y niveles más altos de satisfacción del cliente.

- Escribiendo los tests antes del código, asegura que tenemos como mínimo un test. Una mejor cobertura de test del código mejora los sistemas de calidad y permite al equipo de desarrollo y al cliente tener más confianza en el código.

- El testeo temprano y frecuente nos ayuda a capturar los defectos pronto en el ciclo de desarrollo, lo cual previene que los defectos se extiendan y sean costosos de resolver. Eliminando los defectos pronto en el ciclo de desarrollo mientras el código continua fresco en la memoria de los desarrolladores, en lugar de más tarde, también reduce la cantidad de tiempo gastado encontrando la causa de los defectos.

- El enfoque de escribir sistemas en unidades pequeñas de test lleva a un sistema más modular, flexible y ampliable.

También hay desventajas o costes en este enfoque:

- Las unidades de test son usualmente escritas por el mismo desarrollador que implementará el código; por lo tanto, probablemente veamos las mismas malinterpretaciones de los requisitos en los test y en el código.

- Algunos tipos de funcionalidad, como las interfaces de usuario, son difíciles o muy costosas en tiempo de testear a través de las unidades de test.

- Los tests, por ellos mismos, también precisan de un mantenimiento, y tal y como el proyecto crece y cambia, la carga para sostener los scripts de los tests también lo hace.

- Cuando las personas ven un número alto de tests pasados, pueden tener una falsa sensación de seguridad sobre la calidad del código.

2.6.2.5. Aceptación del desarrollo basado en test (ATDD)

La técnica de aceptación del desarrollo basado en test (ATDD) mueve la atención de los tests del código a los requisitos del negocio. Igual que con el TDD, los tests se crean antes de comenzar el trabajo con el código, y estos tests representan como se espera que se comporte la funcionalidad en un nivel de aceptación de test.

Típicamente captamos estas pruebas cuando cogemos la historia de usuario de la cartera y discutimos su comportamiento deseado con los representantes del negocio. Los test de aceptación pueden ser captados en un marco de test funcionales, como el “FIT” o el “FitNesse”.

El proceso global va a través de las cuatro fases de discusión, destilación, desarrollo y demostración.

Veamos estas cuatro fases con más detalle:

1. **Discutir** los requisitos: Durante la reunión de planificación, planteamos al propietario del proyecto o al cliente preguntas que están diseñadas para obtener el criterio de aceptación.

2. **Destilar** los test en un formato de entorno amigable: En esta siguiente fase, obtenemos los tests listos para ser introducidos en nuestra herramienta de aceptación de test. Normalmente esto implica el estructurar los tests en un formato de tabla.

3. **Desarrollar** el código y enganchar los tests: Durante el desarrollo, los tests se enganchan al código y los tests de aceptación se llevan a cabo. Inicialmente, los tests fallan ya que no pueden encontrar el código necesario. Una vez que el código se escribe y los tests se enganchan a él, los tests validan el código. Pueden volver a fallar, por lo que el proceso de escribir código y testearlo continua hasta que el código para los tests.

4. **Demostración:** El equipo realiza tests exploratorios utilizando los scripts de tests de aceptación automatizados, y hace una demostración del software.

Cuando se combinan las tareas de definir el criterio de aceptación y discutir los requisitos, nos vemos forzados a concretar el acuerdo sobre el comportamiento exacto que debe mostrar el software. De alguna manera, este enfoque refuerza la discusión de la “definición de realizado” a un nivel muy granulado para cada requisito.

Para todas estas técnicas – TFD, TDD y ATDD - el punto principal que es necesario comprender para el examen es que es una buena práctica para el equipo de desarrollo el comprometerse en pensar sobre como el sistema (o producto, si expandimos este concepto al desarrollo de proyectos que no sean de software) debe ser testeado antes de comenzar a desarrollar el producto. Otro punto importante a tener en cuenta para el examen es que crear los tests antes de desarrollar el producto significa que nos encontramos con las pruebas para la mayoría del trabajo de desarrollo, lo cual a su vez significa que el producto será mejor probado.

2.6.2.6. Resolviendo problemas

La última actividad que discutiremos en esta parte es la resolución de problemas. Aunque debemos seguir unos pasos para encontrar indicadores de los proyectos lo antes posible y entonces tomar acciones para evitar que esos problemas ocurran, es inevitable que encontremos problemas en nuestros proyectos. Por lo tanto, necesitamos maneras de resolverlos.

Los métodos ágiles animan a comprometer al equipo a identificar, diagnosticar y resolver los problemas. Por lo tanto, más que ser el dominio del gestor del proyecto o de investigadores externos el buscar y corregir los problemas, es más una actividad del equipo completo. El siguiente capítulo, Mejora Continua, discutirá el proceso de retrospectiva, pero ahora la resolución de problemas.

Hay tres pasos involucrados en la resolución de problemas basada en el equipo.

Paso 1: Reunir datos.

Las actividades de reunir datos ayudan a crear una visión compartida del problema. Sin datos, el equipo simplemente especula sobre qué cambios y mejoras deberían realizarse. Al reunir datos, estamos recolectando e integrando las piezas de un puzle a resolver.

Éstas son algunas técnicas de ayuda basadas en el equipo que pueden ser utilizadas para reunir datos, incluyendo:

- **Línea de tiempo:** Esta técnica ayuda a los participantes a pensar sobre el proyecto desde una perspectiva basada en el tiempo.
- **Triple de monedas de cinco:** Con esta técnica, las cinco mejores ideas son elaboradas por cinco grupos, cinco veces.
- **Puntos de código de color:** Se trata de una técnica de agrupación por afinidad basada en grupos.

- **Loco, triste, alegre:** Esta técnica explora los elementos emotivos del proyecto.
- **Localizar los puntos fuertes:** Con esta técnica los participantes buscan que va bien.
- **Histograma de satisfacción:** Con esta técnica, los participantes revisan sus sentimientos a lo largo del proyecto.
- **Radar de equipo:** Se trata de una herramienta de evaluación multidisciplinaria.
- **De igual a igual:** Se trata de una herramienta de diagnóstico de fortalezas.

Veamos las dos primeras técnicas con más detalle.

Línea de tiempo. El equipo puede utilizar la técnica de la línea de tiempo para diagnosticar el origen y progresión de un único problema o de un número de problemas. Para empezar, el facilitador dibuja una línea de tiempo para el periodo de revisión, el cual puede ser una iteración de la línea de tiempo del problema. El facilitador pide entonces a los miembros del equipo que recuerden eventos buenos, problemáticos y significativos que ocurrieron durante la línea de tiempo.

Trabajando individualmente al inicio, los miembros del equipo escriben eventos en notas adhesivas de colores. El color de las notas indica como categorizan el evento los miembros del equipo (como bueno, problemático, o significativo). Cuando todo el mundo tiene un conjunto de notas, son invitados a colocarlas en la línea de tiempo y revisar lo que otros han puesto.

Pidiendo a los miembros del equipo que recuerden la línea de tiempo, obtenemos una mejor visión de los eventos y de lo que contribuyen a los problemas. Como resultado, podemos obtener pistas sobre cómo evitar problemas similares en el futuro. La representación espacial de causa y efecto que esta técnica provee y la oportunidad de ver las interpretaciones de otras personas de los eventos nos ayuda a recordar detalles adicionales y construir los resultados de otros.

Una vez que las notas han sido colocadas, el equipo discute la línea de tiempo de izquierda a derecha, añadiendo nuevas cartas si es necesario. También registran debajo de la línea de tiempo sus sensaciones sobre los eventos y dibujan una línea de tendencia para indicar las sensaciones del equipo. Expresar su respuesta emocional puede ayudar a hacer salir a la superficie información adicional que puede ser útil para generar la visión (paso 2 del proceso de resolución de problemas basado en equipo).

Triple de monedas de cinco. Se trata de un ejercicio de obtención de datos en el que los participantes destinan cinco minutos a obtener datos de al menos cinco ideas relacionadas con un tema en concreto. El equipo se divide en grupos de cinco (o si el grupo entero es de menos de siete, el ejercicio se hace en un grupo), y los grupos llevan a cabo cinco rondas de expansión. (Esta técnica obtiene su nombre de una competición de tiro en la que se dispara a cinco objetivos desde cinco yardas en 5 segundos).

Este ejercicio pide a la gente que trabaje en solitario al principio y piense en al menos cinco problemas que ocurren durante una iteración. Los miembros del equipo registran entonces los problemas en notas adhesivas o tarjetas. Después de los primeros cinco minutos, todo el mundo pasa sus cartas a la persona de su derecha. Pasan cinco minutos elaborando ideas sobre las cartas que acaban de recibir. Este proceso se repite cinco veces por lo que todo el mundo puede contribuir con sus ideas y tiene una oportunidad de pensar y añadir cosas a las ideas de los demás. El objetivo de esta técnica es crear un ambiente en el cual los participantes tengan tiempo tanto de realizar una reflexión personal como de expandir las ideas de los demás.

Paso 2: Generar ideas

Ahora veremos el segundo paso en el proceso de resolución de problemas – la generación de ideas. Este paso implica ejercicios de colaboración que están enfocados a dirigir y analizar los datos obtenidos en los pasos previos y darles sentido. Las actividades realizadas en este paso ayudan al equipo a interpretar y comprender las implicaciones de los problemas antes de empezar a resolverlos.

Las actividades que ayudan al equipo a generar ideas incluyen:

- La tormenta de ideas.
- Los cinco porqués.
- Fishbone.
- Priorizar por puntos.
- Identificar temas.

Veamos las tres primeras técnicas con más detalle.

TORMENTA DE IDEAS (BRAINSTORMING). El ejercicio de tormenta de ideas pretende generar un gran número de ideas que son entonces filtradas en una lista selecta de ideas e introducidas en el proceso. El alto volumen de ideas generadas durante este ejercicio ayuda a contrarrestar el problema común de que las mejores ideas raramente se ofrecen las primeras. Los equipos pueden tomar diferentes enfoques para esta técnica, incluyendo:

- **Total libertad**: Todo el mundo expone ideas.
- **Ronda por turnos**: La gente pasa una ficha alrededor del grupo. Cuando un miembro del equipo recibe la ficha, tiene que sugerir una idea y pasar la ficha a la siguiente persona.
- **Tiempo en silencio**: Todo el mundo tiene cinco minutos para trabajar individualmente y pensar en ideas, y después se reúnen todos.

Una vez que las ideas han sido generadas, se filtran basándose en el criterio creado por el equipo. Por ejemplo, si el equipo está pensando ideas sobre problemas relacionados con la disponibilidad de los representantes del negocio, el equipo puede decidir añadir filtros para los elementos que son capaces de influenciar y elementos que nos son capaces de influenciar. Entonces revisan cada idea teniendo en cuenta estos filtros y deciden si incluyen o excluyen la idea para el siguiente paso en el proceso de resolución de problemas, decidiendo que hay que hacer.

CINCO PORQUÉS. El objetivo de este ejercicio es descubrir las relaciones causa – efectos subyacentes a un problema particular y llegar a la raíz que causa el problema. La técnica se origina en Toyota y es utilizada rutinariamente en las técnicas Lean.

Cuando utilizamos el ejercicio de los cinco porqués en un equipo, las personas trabajan en parejas o en pequeños grupos. Dentro de sus grupos, preguntan “¿Por qué?” cinco veces para moverse más allá de las respuestas automáticas y habituales y llegar así a la raíz de los problemas. Aquí un ejemplo:

Pregunta 1: ¿Por qué obtuvimos este fallo del sistema en la demostración de la iteración?

Respuesta 1: Intentamos acceder a los datos de ventas para una tienda que no tiene ventas.

Pregunta 2: ¿Por qué el acceder a una tienda sin ventas da problemas?

Respuesta 2: La rutina de búsqueda devuelve un valor nulo que no puede ser soportado por el sistema.

Pregunta 3: ¿Por qué no capturamos los valores nulos y mostramos un mensaje de error más claro?

Respuesta 3: Podemos capturarlos ahora que sabemos de ellos, pero era la primera vez que los habíamos visto en las ventas.

Pregunta 4: ¿Por qué no se codifican todos los resultados para soportar los nulos?

Respuesta 4: No lo sé; nunca ha sido una prioridad.

Pregunta 5: ¿Por qué no es una prioridad, parece un punto débil del sistema?

Respuesta 5: Estoy de acuerdo. Deberíamos añadirlo en el módulo de checklist.

FISHBONE. Un diagrama Fishbone (raspa de pescado) es una herramienta visual que a menudo acompaña al ejercicio de los cinco porqués. Es una manera de mostrar la causa principal del análisis de problemas. Con esta técnica, el equipo identifica los factores que están causando o afectando la situación del problema y buscan las causas más probables.

El proceso comienza con el dibujo de un diagrama Fishbone vacío y escribiendo el problema en la "cabeza". El siguiente paso es identificar las categorías de los factores que contribuyen, los cuales son también escritos en el diagrama. Las categorías pueden estar relacionadas con las preguntas del ejercicio de los cinco porqués, o pueden estar relacionadas a un conjunto de categorías normalmente utilizadas, como:

- Gente, procedimientos, políticas, lugares
- Sistemas, suministradores, habilidades, alrededores.

Otra opción para el equipo es elegir cualquier categoría que tenga sentido para el problema. El facilitador del ejercicio pregunta a los miembros del equipo "¿Cuáles son [rellenar con el nombre de la categoría] los factores que contribuyen al problema?" Las respuestas se rellenan entonces como los "huesos" en las categorías.

Paso 3: Decidir qué hacer.

El último paso en el proceso de resolución de problemas es decidir qué hacer con el problema. Las técnicas basadas en el equipo utilizadas en este paso incluyen experimentos para validar los enfoques y los objetivos mesurables para registrar el progreso y la resolución de problemas. Ayudan a construir un conjunto claro de acciones para resolver el problema.

Las técnicas normalmente utilizadas en este proceso incluye:

- Temas cortos.
- Objetivos SMART.
- Juego de planificación retrospectiva.
- Círculo de preguntas.

Veamos las dos primeras técnicas en más detalle.

TEMAS CORTOS. Esta actividad ayuda al equipo a ponerse de acuerdo en las acciones para la resolución de problemas. Se presenta al equipo folios o pizarras con las categorías escritas en ellas, y entonces el equipo se pone de acuerdo sobre con que ideas proseguir. Las categorías pueden incluir:

- Que va bien, Hazlo diferente la siguiente vez.
- Guardar, Borrar, Añadir.
- Empezar a hacer, Para de hacer, Hacer más de, Hacer menos de.

OBJETIVOS SMARTS. Esta actividad ayuda al equipo a crear metas que son Especificas (Specific), Medibles (Measurables), Alcanzables (Attainable), Relevantes (Relevant) y Oportuno (Timely) (SMART). Los objetivos con estas características son más probables de ser conseguidos exitosamente. Al inicio del ejercicio de objetivos SMART, el facilitador lista las características SMART en una hoja o pizarra en blanco y entonces se compara un objetivo no SMART como por ejemplo “Necesitamos realizar más pruebas”, con una versión SMART del mismo, como por ejemplo “Cada módulo debe tener y pasar una unidad de test, un test funcional, y un test de sistema antes de terminar la iteración”.

Una vez que todo el mundo entiende las características de los objetivos SMART, se forman grupos más pequeños para transformar las acciones de resolución de problemas en objetivos SMART. El equipo revisa entonces cada objetivo. Discuten si los objetivos son realmente objetivos SMART y hacen modificaciones donde sea necesario. El proceso de asegurar que los objetivos tengan las características SMART confirma también que la gente comprende que puede pasar y ayudarles a formar modelos mentales de que debe hacerse para completar los objetivos y resolver los problemas.

2.6.3. ¿Por qué centrarse en comprometer al equipo?

Antes de entrar en la parte de ética y responsabilidad de esta parte, veamos un concepto importante. Cuando intentan resolver los problemas del proyecto, las empresas demasiado a menudo pasan por alto la mejor fuente de soluciones – el equipo del proyecto. El equipo normalmente tiene las mejores soluciones prácticas incluso cuando no parezcan las mejores en la teoría.

Veamos los diferentes beneficios de comprometer al equipo:

1. **Al preguntar a un equipo una solución, heredamos consenso para la propuesta.**
2. **Comprometer al equipo nos da acceso a un amplio conocimiento de los hechos.**
3. **Las soluciones son prácticas.**
4. **Cuando se realizan consultas, las personas trabajan duro para generar buenas ideas.**
5. **Pedir ayuda muestra confianza no debilidad.**
6. **Las ideas de otros modelan el comportamiento deseado.**

Más allá de los beneficios que obtenemos de comprometer al equipo en la resolución de problemas, hay otros pocos puntos a considerar sobre la utilización de este enfoque, que incluyen precauciones a tener en cuenta. Veamos esto:

USO Y PRECAUCIONES.

A las personas no les gusta que les traten como a máquinas. En lugar de eso, si les pedimos que piensen, nos sorprenderán con soluciones innovadoras y prácticas que tienen un gran respaldo de las personas que necesitan implementar las soluciones. El equipo tiene el conocimiento de primera mano del problema y, lo que es más importante, que tipos de acciones son prácticas para resolverlo.

Naturalmente, comprometer a los equipos no es un remedio que lo cure todo. Hay algunos puntos que necesitamos tener en mente:

- **Resolución de problemas reales**: Deberíamos utilizar al equipo para resolver problemas reales únicamente, no para tomar decisiones como que marca de tóner comprar para la impresora. Si preguntamos al equipo cualquier decisión, hasta las más pequeñas, el trabajo no avanzará. Comprometer al equipo para resolver los problemas, es una herramienta a utilizar cuando te encuentras atascado en el problema a solucionar.

- **Cohesión de equipo pobre**: Si el equipo está fragmentado y contiene grupos que se oponen, el sentimiento de que el otro grupo está resolviendo sus problemas, puede minar el proceso. Necesitamos mantener a los miembros del equipo alineados los unos con los otros para lograr que el proceso de resolución de procesos sea más efectivo.

- **Cambios de equipo y proyecto**: Si una parte importante del equipo cambia durante un largo periodo de tiempo, necesitamos evaluar al equipo para asegurarnos de que siguen en línea con el enfoque de resolución de problemas. Tener que llevar a cabo las ideas de otros es tan malo como no ser consultado, por lo tanto necesitamos evaluar si las personas continúan estando de acuerdo con esta política.

- **Seguimiento a través de**: Una vez que has preguntado por las soluciones, asegúrate de que las aplicas. Es muy desmoralizante que te pidan trabajar en una solución y luego ver que esa solución no se aplica.

2.6.4. Ética y responsabilidad profesional

Antes de pasar al resto de capítulos veremos como el código de conducta ética y profesional se aplica a los conceptos de resolución y detección de problemas. Hay cuarto áreas principales de este código – responsabilidad, respeto, justicia y honestidad – que veremos por turno., buscando ejemplos de cómo se relacionan con los conceptos vistos anteriormente.

Responsabilidad.

Tomar decisiones basadas en los intereses de la compañía. Debemos atacar a los problemas pronto; retrasar su resolución no es lo más interesante para la compañía, y fuerza a cambiar cada paso de la curva de costes de cambios. Si cedemos a la tentación de "dejarlo aparcado a ver si se soluciona solo", no estaremos actuando responsablemente. En su lugar, debemos tratar los problemas tal y como aparecen. Las primeras soluciones son por lo normal más rápidas y mejores.

Proteger la información propietaria: Si investigando un problema, encontramos que los errores tienen que ver con proveedores externos, necesitamos ser cuidadosos de que comunicamos la información solo a las partes interesadas, y que estamos respetando los acuerdos relacionados con la información propietaria y la confidencialidad.

Reportar comportamiento no ético y violaciones. Si los problemas de los que se quejan los miembros del equipo tienen que ver con groserías, faltas de respeto, acoso a los miembros de la plantilla, debemos comunicar estos problemas al departamento de recursos humanos y a cualquier otro grupo que sea apropiado. No es suficiente con intentar solucionarlo dentro del proyecto.

Respeto.

Mantener una actitud de mutua cooperación. La resolución de problemas debería centrarse en encontrar y arreglar los problemas, no en señalar y culpar. El enfoque de identificar y solucionar los problemas no es posible si el equipo está más pendiente de culpar a otros.

Respetar las diferencias culturales. Deberíamos reconocer que las personas de otras culturas pueden sentirse menos cómodas reportando o discutiendo problemas en la presencia de sus responsables. Por lo tanto, deberíamos proveer de mecanismos para permitir una contribución segura y anónima.

Hacer frente al conflicto directamente. Los problemas relacionados con conflictos en el equipo deben ser tratados inmediatamente. La respuesta de "guárdalos para la retrospectiva" no es apropiada.

No uses tu posición para influenciar a otros. Necesitamos facilitar no dictar; deberíamos dejar a los miembros del equipo solucionar los problemas del proyecto. Haciendo esto, no solo obtendremos mejores soluciones sino que además serán mejor aceptadas internamente.

Justicia.

Buscar y revelar conflictos de interés. Tal vez hayamos encontrado el problema y seamos nosotros mismos. Si el equipo tiene problemas con el gestor del proyecto, el ScrumMaster, o el líder del equipo, deberíamos ofrecer la asistencia de una tercera parte y estar preparados para escuchar las quejas.

Honestidad.

Comprender la verdad. Cuando diagnosticamos problemas, deberíamos intentar comprender porque la gente actúa de la manera en la que lo hace. Que motiva su comportamiento. A veces las personas se comportan de formas inesperadas porque su ambiente de reglas, recompensas, y reprimendas ha sido trastocado.

Ser veraz en todas las comunicaciones. A las personas no les gusta escuchar malas noticias ahora, pero les gusta aún menos escuchar peores noticias más tarde. Puede ser difícil al principio reportar malas noticias o problemas, pero el reportar lo antes posible es lo mejor que se puede hacer. Reportando los problemas anticipadamente, tenemos una mejor oportunidad de repararlos.

EJERCICIOS

En el Anexo F.6, se encuentra un examen tipo sobre lo tratado en este punto.

2.7. MEJORA CONTINUA

En este último capítulo hablaremos de la mejora continua en los proyectos ágiles. Las técnicas y herramientas y conocimientos y habilidades a conocer para el examen son las siguientes:

Practica	Herramienta/técnica	Conocimiento / Habilidad (nivel)
Prácticas de mejora continua	> Retrospectivas > Procesos de adaptación	> Compartir el conocimiento (Nivel 1) > Principios de los sistemas de pensamiento (Nivel 3) > Análisis de procesos (Nivel 2) > Aplicación de las nuevas prácticas ágiles (Nivel 3) > Código de Ética PMI (Nivel 2) > Proceso de mejora continua (Nivel 2) > Auto ayuda (Nivel 2)

Tabla 19: Conocimientos y habilidades para el examen PMI-ACP

2.7.1. Prácticas de mejora continua

La mayoría de proyectos tradicionales capturan la mayoría de sus lecciones aprendidas al final del proyecto. La intención detrás de capturar estas lecciones es permitir a la empresa aplicarlas a futuros proyectos que tengan un campo parecido, o a proyectos con dinámicas de equipo similares.

Este enfoque no es válido. Necesitamos aplicar los beneficios de lo que aprendemos sobre la marcha en nuestro proyecto actual, y lo antes posible.

La aplicación inmediata de las lecciones aprendidas es especialmente crítica para los proyectos con cambios rápidos y para proyectos con niveles altos de incertidumbre y riesgo, en los que mantener el rumbo puede ser fatal para el proyecto si las circunstancias cambian.

Por lo tanto, los proyectos ágiles programan las actividades de mejora continua en el plan como parte de la metodología. Este enfoque ágil a las lecciones aprendidas es deliberado y frecuente, y ayuda a asegurar que el equipo considera de forma regular la adaptación y la mejora hasta el punto en que se vuelve habitual y parte de su forma de trabajar.

2.7.1.1. Retrospectivas

Las retrospectivas, comunes a todos los métodos ágiles, son el primer evento de aprendizaje, reflexión y reajuste de los proyectos ágiles. Una retrospectiva es una reunión especial que se lleva a cabo después de cada iteración, en la cual los miembros del equipo se reúnen para inspeccionar y mejorar sus métodos y trabajo de equipo.

Como las retrospectivas ocurren durante el proyecto, las lecciones y mejoras que resultan de ellas son aplicables y pertinentes al trabajo por llegar; después de todo, el trabajo por llegar tendrá el mismo campo de negocio, técnico y dinámicas de equipo que la iteración que se está evaluando. En otras palabras, la retrospectiva ofrece valor inmediato al proyecto actual, en lugar de únicamente documentar buenos consejos y deseos para un proyecto futuro con dominios o dinámicas similares.

Las retrospectivas ofrecen un número de beneficios para los equipos, incluyendo los siguientes tipos de mejoras:

- **Productividad mejorada:** Aplicando las lecciones aprendidas y reduciendo los trabajos repetidos, el equipo puede obtener más trabajo productivo realizado.
- **Mejora de las capacidades:** Las retrospectivas proveen de un foro para difundir el conocimiento no usual, y así como aumenta el número de personas con este conocimiento, aumenta el número de personas que lo pueden utilizar.
- **Mejora de la calidad:** Podemos mejorar la calidad de nuestros proyectos encontrando las circunstancias que llevan a defectos y eliminándolas.
- **Mejora de la capacidad:** Las retrospectivas se centran en encontrar mejoras en la eficiencia de los procesos, que puedan mejorar la capacidad del equipo para trabajar.

El proceso de retrospectiva atraviesa los siguientes cinco pasos (ver figura 60):

1. Sentar las bases.
2. Reúne datos.
3. Genera ideas.
4. Decide que hacer.
5. Cierra la retrospectiva.

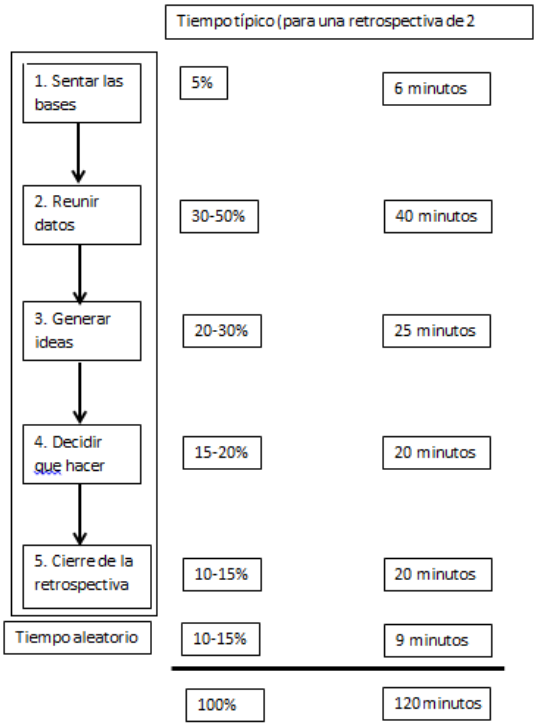


Figura 60: Proceso de retrospectiva. (Fuente: Elaboración propia)

Estos pasos se llevan a cabo en un ciclo continuo sincronizado con las iteraciones.

Veamos los pasos de la retrospectiva con más detalle:

Paso 1: Sentar las bases

Al inicio de la retrospectiva, necesitamos sentar las bases para ayudar a las personas a centrarse en la tarea para reflejar como van las cosas. También preparamos a los participantes para los siguientes pasos en la retrospectiva de obtención de datos y generación de ideas. Al establecer las bases, nuestro objetivo es crear la atmosfera en la que las personas se sientan cómodas hablando sobre cosas que pueden no haber ido muy bien en el proyecto.

Una técnica utilizada a menudo para que la gente participe es hacerles hablar pronto en el proceso de retrospectiva. La teoría detrás de este enfoque es que si no hablan pronto después del inicio de la retrospectiva, se puede entender que es correcto no hablar. Sentir que el silencio es aceptable es contraproducente, ya que el punto principal de la retrospectiva es que las personas hablen sobre cómo van las cosas y sobre qué queremos hacer en el futuro. Por lo tanto, podemos hacer que las personas comiencen pidiéndoles que se presenten la primera vez que una sesión de se lleva a cabo. También podemos pedir a los participantes que remarquen que esperan obtener de la retrospectiva, o decir una o dos palabras que describan que sienten en la iteración o sobre el progreso. El objetivo es lograr que la gente se acostumbre a hablar en el grupo y animarles a contribuir durante toda la reunión.

La siguiente parte de establecer las bases es remarcar el ámbito de la retrospectiva y las materias a discutir. También necesitamos establecer algunos valores del equipo y acuerdos sobre cómo llevar a cabo la retrospectiva. Son acuerdos del equipo sobre qué es aceptable y qué no lo es.

El elemento final de establecer las bases es que las personas estén con el ánimo adecuado para contribuir con información e ideas. Los participantes pueden sentirse inquietos sobre sacar a la luz problemas por miedo de conflicto. Pueden sentir que el criticar el proceso refleje una mala imagen sobre ellos o su gestión. No obstante, el objetivo real de la retrospectiva es simplemente encontrar vías de mejora. El hecho que hayamos cometido errores antes será una pequeña consecuencia si podemos continuar y no repetir estos errores de nuevo. Hacer que la gente entienda y crea en este concepto es complicado.

Las actividades que ayudan a establecer las bases incluyen:

- Recepción.
- Poner y quitar la atención.
- ESVP
- Acuerdos de trabajo.

Veamos estas actividades con detalle.

Recepción: Utilizamos este ejercicio para ayudar a las personas a poner a parte sus preocupaciones y centrarse en las retrospectiva. En un formato de ronda de turnos, pedimos a la gente que resuma en una o dos palabras que esperan obtener de la retrospectiva, que tienen principalmente en mente, o que opinan de la retrospectiva.

Poner y quitar la atención. Utilizamos esta actividad para establecer una mentalidad de comunicación productiva.

En este ejercicio, remitimos a los participantes a la pizarra y les pedimos que discutan que términos tienen significado para ellos, invitándoles a proveer ejemplos. Después, les preguntamos si están dispuestos a permanecer en la comuna de la izquierda Focus On. Si no están de acuerdo, debemos explicar los problemas de permanecer en la derecha e intentar obtener consenso.

ESVP. Éste es un ejercicio de grupo en el que los participantes, de manera anónima, se asocian a ellos mismos con una de las siguientes identidades y registran su elección en un papel:

- **Exploradores**: Los exploradores están ansiosos por descubrir nuevas ideas, y quieren aprender todo lo que puedan.
- **Compradores**: Los compradores buscan entre toda la información disponible y compran cualquier nueva idea útil.
- **Turistas**: Los turistas no están interesados en el trabajo de la retrospectiva, pero están felices de encontrarse fuera de su lugar habitual de trabajo.
- **Prisioneros**: Las personas que se clasifican a sí mismas como prisioneras sienten que están siendo forzadas a atender a la retrospectiva y estarían haciendo cualquier otra cosa.

Los resultados anónimos son recogidos y contados por el grupo de manera que podamos medir el nivel de energía y compromiso de los participantes.

Después de que los resultados hayan sido contados, podemos destruir los papeles para que nadie esté preocupado de que se pueda conocer su respuesta. Entonces preguntamos a los participantes cómo se sienten respecto a los marcadores y qué significan estos resultados para la retrospectiva.

Acuerdos de trabajo. Para esta actividad, formamos pequeños grupos con los participantes y les damos diferentes temas en los que trabajar. Pedimos a los pequeños grupos que definan y expliquen los acuerdos de trabajo que quieren que se establezcan en la retrospectiva. Entonces, con el grupo completo, destinamos tiempo para clarificar y refinar las ideas, desarrollando una única lista en la que trabajar.

Paso 2: Reunir datos.

En la fase de reunir datos, creamos una imagen compartida de que sucede durante la iteración. Sin una visión común de lo que ocurre, el equipo solo especulará en que cambios o mejoras realizar y puede en realidad estar llevando a cabo problemas diferentes sin darse cuenta.

Como vimos en el capítulo anterior, existen actividades basadas en el equipo que pueden ser utilizadas para reunir datos.

Cuando acabamos con este paso del proceso, deberíamos tener una vista entendible de observaciones, hechos y hallazgos, los cuales tienen un conocimiento compartido por el equipo.

Paso 3: Generar ideas.

Esta fase da al equipo tiempo para evaluar los datos que fueron reunidos en los pasos previos y encuentran ideas útiles a partir de ellos. El objetivo de generar ideas es ayudar a los miembros del equipo a comprender las implicaciones de sus hallazgos y discusiones.

Paso 4: Decidir qué hacer.

Las actividades involucradas en el paso de “Decidir qué hacer” hacen pasar al equipo de pensar en la iteración que acaban de completar a pensar en la siguiente iteración, incluyendo que deberían cambiar y si su comportamiento debe ser diferente.

En este paso, el equipo identifica los elementos de acción de alta prioridad, crea planes detallados para los experimentos, y establece objetivos mesurables para conseguir los resultados deseables.

Paso 5: Cierre de la retrospectiva.

El paso final es cerrar la retrospectiva. Proveemos a los miembros del equipo de oportunidades para reflejar que ha pasado durante la retrospectiva y para expresar valoración los unos de los otros. Las actividades que resumen que decide mantener el equipo y que cambiar nos ayudan a hacer un mejor uso de nuestro tiempo y a reforzar el valor de la retrospectiva.

Hay varias actividades basadas en el equipo que pueden ser útiles en esta fase final:

- Plus/Delta.
- Ayudado, obstaculizado, Hipótesis.
- Retorno del tiempo invertido.
- Valoraciones.

Las veremos en más detalle:

Plus/Delta. En este ejercicio, capturamos y validamos las ideas del equipo las cuales deberíamos hacer más (cosas que están yendo bien) y que deberíamos cambiar (cosas que no están yendo bien) en una pizarra o tarjeta.

Ayudado, obstaculizado, Hipótesis. Este ejercicio ayuda a generar feedback en el proceso de retrospectiva y produce ideas de mejora. Para llevar a cabo esta sesión, preparamos primero tres cartas, una titulada “Ayudado”, otra titulada “Obstaculizado” y la última con el título de “Hipótesis”. Explicamos al equipo que estamos buscando mejorar el proceso de retrospectiva y que queremos feedback sobre lo que ellos creen ayudará, lo que creen que será un obstáculo y cualquier idea de mejora (hipótesis) para las siguientes retrospectivas. Los participantes escriben estas ideas en notas adhesivas y colocan las notas en la pizarra adecuada.

Retorno del tiempo invertido. Éste es otro ejercicio para generar feedback en el proceso de retrospectiva. Este ejercicio se centra en si las personas piensan que su tiempo se ha gastado correctamente.

Valoraciones. Durante este ejercicio, los miembros del equipo tienen una oportunidad de expresar su valoración a los otros miembros del equipo por su ayuda, contribuciones, esfuerzos para resolver los problemas, etc.

2.7.1.2. Compartir el conocimiento

Compartir el conocimiento es un componente clave de los métodos ágiles. No debe de parecernos sorprendente ya que los métodos ágiles están diseñados para ambientes de trabajo del conocimiento (como recordatorio, un proyecto de trabajadores del conocimiento se caracteriza por expertos en una materia colaborando para crear o mejorar un producto o servicio). Visto que el conocimiento es la base de los proyectos ágiles, veremos cómo compartirlo.

La difusión del conocimiento ocurre en muchos niveles, de manera obvia y no tan obvia. Las demostraciones de un producto son un ejemplo de un método obvio. El propósito principal de estas demostraciones es no mostrar el producto, ya que el equipo sabe muy bien lo que funciona y lo que no; en su lugar, las demos se llevan a cabo porque son formas ceremoniales de compartir los conocimientos a través de los siguientes tipos de diálogo:

Del equipo al cliente: Aquí se incluye qué se cree que se ha pedido y qué se ha sido capaz de crear, de manera que el cliente pueda valorar si es correcto y está enfocado a lo que esperaba.

Del cliente al equipo: Me gusta esta parte, creo que está bien, pero esta otra parte no está bien. Y esto me recuerda que necesitamos algo para poder hacer X.

Un ejemplo de una manera menos obvia de compartir información es la co-ubicación del equipo. Esta práctica no se lleva a cabo para ahorrar espacio o para poder ejercer un control más fácil; se lleva a cabo para aprovechar la difusión del conocimiento tácito (no escrito) que ocurre en los ambientes cara a cara y a través de comunicación osmótica.

Casi todas las prácticas ágiles están estructuradas para compartir el máximo de conocimiento. En los proyectos de software, por ejemplo, los desarrolladores son grupos de expertos en tecnología, por lo que, ¿por qué no se realizan las reuniones diarias vía e-mail o a través de otra forma de tecnología? La respuesta es simple. La causa es que el objetivo real de la reunión diaria es compartir información dentro del equipo, no solo crear una lista de trabajos realizados, trabajos planeados o de cualquier problema que haya aparecido. Este mismo concepto se aplica a muchas prácticas ágiles; cuando examinamos como están construidas a través del punto de vista de maximizar la transferencia de conocimiento, todas las peculiaridades de éstos cobran sentido. Por ejemplo, miremos las prácticas XP. Hasta cierto punto, la mayoría de estas prácticas tienen un componente de transferencia de conocimiento.

Las retrospectivas son también vehículos de transmisión del conocimiento. Otra forma en la que los métodos ágiles enfatizan la difusión del conocimiento es utilizando herramientas simples como cartas en un muro para planificar y programar un proyecto. Estas actividades simples incluyen a más miembros del equipo y requieren menos habilidades, formación y herramientas que las necesarias para programas informáticos como Primavera o Microsoft Project. Las herramientas simples también ayudan a crear una comprensión común del plan.

Por lo tanto, si entendemos que compartir conocimiento es algo bueno, ¿por qué a veces es tan complicado hacerlo? Normalmente se recompensa más por lo que se sabe que por lo que se comparte. Este sistema de recompensa, normalmente, desmotiva el compartir conocimiento. Para promocionar la idea de compartir conocimiento, la cultura de empresa debería fomentar y premiar el descubrir, innovar y compartir información.

“EVALUAR UN NIVEL POR ENCIMA” PARA FOMENTAR EL COMPORTAMIENTO DESEADO.

Tiendes a perder aquello que no puedes medir, como por ejemplo la difusión del conocimiento, las ideas, la colaboración, la creatividad. Este hecho es un dilema y un problema potencial, pero afortunadamente las personas han encontrado métodos para medir y recompensar la difusión de conocimiento para no perder estos beneficios. La siguiente historia ilustra este concepto:

Nucor Steel ha sido una de las compañías de acero más exitosas en los Estados Unidos desde 1980 hasta la actualidad. Donde otras compañías se han retirado ante la feroz competencia extranjera y relaciones laborales pobre, Nucor ha seguido competitiva y rentable y ha mantenido unas buenas relaciones laborables. Ellos atribuyen buena parte de este éxito a su esquema "paga por rendimiento", el cual se basa en la productividad del equipo.

En Nucor, al contrario de en otras compañías que utilizan esquemas de bonificación tradicionales basados en la productividad, si eres un jefe de una planta de acero, no se te paga en función de lo bien que lo haga tu planta sino en función de como de bien lo hagan todas las plantas. En el siguiente nivel inferior, los jefes de departamento no obtienen un bono basándose en el rendimiento de su departamento sino en el rendimiento de todos los departamentos. Esta forma de actuar se extiende por toda la organización, incluyendo a los líderes de equipo quienes obtienen beneficios en función de la productividad de todos los equipos, no solo del suyo, y los individuos, quienes obtienen beneficios en base al rendimiento de sus equipos, en lugar de al suyo propio.

Esta forma de actuar da el resultado de un conocimiento compartido. Por ejemplo, si un gestor de planta crea una mejora para un proceso o descubre una manera de ahorrar dinero, estará incentivado para compartir este conocimiento con otras plantas. De la misma manera, los equipos que coordinan procesos son recompensados por compartir sus ideas con los otros equipos, y los miembros de un equipo que ayudan a sus compañeros de equipo, son también recompensados.

La forma de actuar descrita en este ejemplo se llama "Evaluar un nivel por encima", lo cual se refiere a medir algo a un nivel superior al del ámbito de control normal (p.e., el nivel de equipo, en lugar del nivel individual) para fomentar la cooperación y la difusión de conocimiento. Esto ocurre en los proyectos ágiles, aunque no se haga mucha propaganda de ello. Mary Poppendieck, una experta en métodos Lean, aconseja "En lugar de asegurarse de que la gente es evaluada en su ámbito de control, es más efectivo evaluar a las personas un nivel por encima de su ámbito de control. Ésta es la mejor manera de fomentar el trabajo de equipo, la colaboración, y la optimización global en lugar de local".

Hemos visto ejemplos de evaluaciones un nivel por encima en la manera en como controlamos la velocidad. Podemos fácilmente trazar la velocidad a miembros individuales de un equipo y determinar quién es el más productivo, pero esta forma de actuar probablemente fomentará comportamientos negativos y falta de cooperación. Por lo tanto, en su lugar medimos la velocidad a nivel de equipo; como resultado los miembros del equipo estarán motivados a ayudarse los unos a los otros. El mismo concepto aplica a la difusión del conocimiento. Para que la difusión del conocimiento ocurra, necesitamos basar nuestro seguimiento y recompensas en los logros del equipo, por lo tanto no se obtendrán beneficios de retener información o de ser el mejor en hacer X.

Éste es el motivo por el que muchas de las técnicas ágiles, como la propiedad de código común o la programación por pares, promocionan la difusión de conocimiento; queremos el conocimiento repartido por todo el equipo, no residiendo en una única persona. El uso prolífico en los métodos ágiles de gráficos, de emisores de información, personas, y de diagramas funcionales apoyan el concepto de compartir información, de la misma forma.

Por lo tanto, la mentalidad y la forma de actuar es “Déjanos que te mostremos lo que sabemos, y entonces podrás decirnos si estamos en lo cierto o si nos equivocamos”. Es una buena manera de sacar a la luz y resolver los malentendidos, pero puede ser una manera de actuar extraña para las empresas que son más reservadas y cautas en sus comunicaciones. Tampoco se trata de jactarse de todo, no obstante. Se trata de gestionar los riesgos, confirmar que la manera en la que estamos haciendo las cosas es apropiada y aporta valor, y estamos centrados en transferir conocimiento. Éstos son los componentes clave cuando llevamos a cabo proyectos de trabajadores de conocimiento, que transforman información, en lugar de hormigón o acero.

2.7.1.3. Proceso de adaptación

Cuando aprendemos procesos en proyectos ágiles, modificamos las metodologías para adaptarlas al ambiente del proyecto que estamos utilizando. Algunas metodologías son fácilmente adaptables, como el Kanban.

En contraste, métodos como el Scrum son menos adaptables. El método Scrum realiza recomendaciones de cómo transformar la empresa en un ambiente ágil que soporte mejor la manera de trabajar del método Scrum, ya que cambiar la metodología puede dañar o corromper el propio método con prácticas no standard para ese ambiente.

Claramente hay beneficios y riesgos en los dos extremos. Cuando consideramos la pregunta de si el proceso de adaptación es apropiado, debemos reconocer primero que todos los proyectos son diferentes. Resuelven diferentes problemas, encaran diferentes desafíos, utilizan personas diferentes, y operan en diferentes empresas con diferentes culturas y normas. La naturaleza única de los proyectos es el por qué iluminados cómo Jim Highsmith y Alistair Cockburn recomiendan una manera de actuar de proceso por proyecto - en otras palabras, crear procesos que sean específicos para el trabajo que tenemos entre manos.

Otros toman otra visión diferente. Ron Jefferies pretende mantener una adherencia a los valores y prácticas ágiles. Su preocupación es que las metodologías pueden hacer de la manera de trabajar ágil algo irreconocible y entonces, cuando los proyectos fallen, se le echará la culpa a la metodología ágil.

Por lo tanto tenemos dos visiones diferentes, ambas con puntos de vista válidos a considerar. ¿Dónde nos lleva esto? El riesgo de fallar es grande cuando gente sin experiencia en métodos ágiles modifica los métodos.

El equilibrio y las interrelaciones de las prácticas ágiles es complicada.

Como hemos visto en capítulos anteriores, eliminar o aumentar elementos sin entender las relaciones entre ellos, puede llevar a problemas.

A veces es tentador evitar una práctica porque es difícil o no es aceptada. Esto puede ser una señal de que la práctica no ha encajado bien en la empresa, pero a su vez está resaltando un problema que debe ser resuelto. Por ejemplo, si el equipo de desarrollo de software no está cumpliendo con los tiempos de una iteración de dos semanas porque necesita tres días para compilar y liberar un entorno de prueba, no se debe automáticamente cambiar el periodo de la iteración a tres o cuatro semanas, sino que se debe investigar porque el proceso de compilar y liberar necesita de tanto tiempo, y mejorarlo.

Por lo tanto, para el examen, se ha de comprender que el proceso de adaptación puede ser efectivo y productivo, pero que se debe estar alerta de los riegos que implica esta práctica. A continuación veremos las maneras claves de mitigar estos riesgos:

1. **Intenta utilizar el método ágil tal cual, antes de intentar cambiarlo.** Los métodos son creados basándose en el conocimiento colectivo de muchos profesionales, por lo tanto no te apresures en cambiarlos.

2. **Examina cuidadosamente la motivación para abandonar, modificar o añadir una práctica.** ¿Es el cambio una salida para evitar problemas más fundamentales, o verdaderamente será valioso?

2.7.1.4. Principios de sistemas del pensamiento

Antes de empezar a cambiar el proceso ágil, necesitamos comprender más sobre el ambiente o sistema, en el cual se utiliza. Los proyectos de trabajadores del conocimiento se caracterizan por el nivel de complejidad alrededor de los requisitos, así como por la complejidad de la tecnología utilizada en ellos. La combinación de estas complejidades nos da el paisaje del proyecto ilustrado en la figura 61.

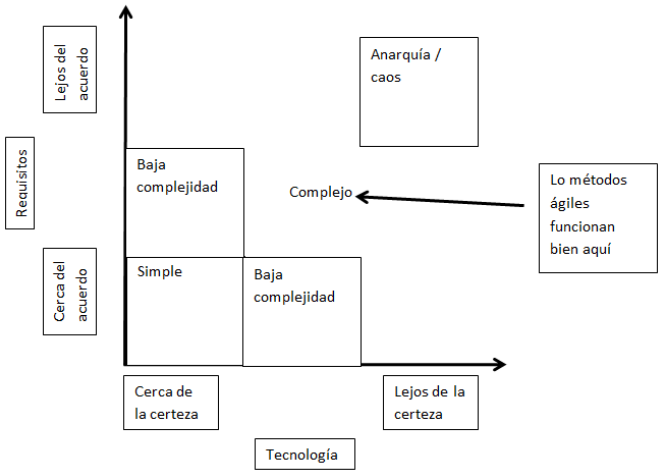


Figura 61: Niveles diferentes de complejidad en los proyectos. (Fuente: Elaboración propia)

Vemos los rangos de proyectos desde simple (abajo a la izquierda) hasta los categorizados como anárquicos o caóticos (arriba a la derecha). La manera de trabajar ágil funciona bien para proyectos situados en la mitad. Estos proyectos son complejos y tienen incertidumbre alrededor de los requisitos y el enfoque técnico. Los métodos ágiles pueden naturalmente utilizarse en proyectos simples, en los cuales la empresa y el equipo obtendrán los beneficios del incremento de colaboración, comunicación, y visibilidad, pero estos proyectos funcionarán bien con métodos más tradicionales.

En contraste, los proyectos complejos empiezan a ponerse difíciles cuando se mezclan con métodos tradicionales.

No obstante, los métodos ágiles no son la panacea. Si no hay un acuerdo claro sobre que estamos construyendo o qué manera de trabajar o herramientas estamos utilizando para construirlo, nuestro proyecto será un caos. En este caso, ni el método ágil ni otra forma de trabajar podrá asegurar el éxito.

2.7.1.5. Análisis de procesos

El proceso de análisis está estrechamente relacionado con el proceso de adaptación y los principios de los sistemas de pensamiento. Cuando llevamos a cabo el proceso de análisis, estamos revisando y diagnosticando problemas con los métodos ágiles. Este análisis puede llevar a la decisión de adaptar el proceso.

Alistair Cockburn provee la siguiente lista de antipatrones de metodología (o cosas negativas sobre las metodologías) a tener en cuenta:

1- **Un tamaño para todos los proyectos:** No es posible crear la metodología óptima para todos los tipos de proyectos, todas las tecnologías y todos los tamaños de equipos. Por lo tanto, tened cuidado con las afirmaciones de que un tamaño encaja en todas las metodologías de trabajo.

2- **Intolerante:** Una metodología es como una camisa de fuerza en la que hay un conjunto de convenios y políticas a los cuales las personas se adhieren y utilizan. El tamaño y forma de la camisa de fuerza debería ser elegida por el equipo y no debería ser más estrecha de lo necesario, para que las personas tengan un poco de maniobrabilidad en sus acciones.

3- **Pesada:** Es una creencia común pero errónea el que la metodología más pesada es en la práctica, la más segura. No obstante, añadir peso a una metodología no es probable que mejore la oportunidad del equipo de entregar el proyecto exitosamente. En su lugar, desvía el tiempo del equipo del objetivo real del proyecto.

4- **Adornado:** Todos los métodos tienden a adornarse. Añadimos cosas que creemos “deben” ser realizadas, pero debemos ver esto como una señal de añadidos potencialmente caros y causantes de errores.

5- **No probado:** Muchas metodologías no están probadas. Son propuestas creadas de la nada que son “debes” en acciones en toda regla. Por ejemplo, cuan frecuentemente has escuchado frases como. “Bien, ¿realmente parece que puede funcionar?” En lugar de crear metodologías teóricas muy complejas, es mejor reutilizar, ajustar, modificar y crear justamente lo que necesites. Observa que funciona realmente en los proyectos y utilízalo, no algo que no ha sido probado y que pensamos que “debería funcionar”.

6- **Utilizado una vez:** Una metodología que es utilizada una vez es un poco mejor que una que no se ha utilizado nunca, pero sigue no siendo la receta para el éxito. La realidad es que diferentes proyectos necesitan diferentes formas de trabajar, y porque una forma de trabajar haya funcionado bien en determinadas circunstancias, no garantiza que funcione bien en otras.

Si lo que acabamos de ver eran los puntos a tener en cuenta con una metodología, ¿cuáles son los signos de éxito que indican que estamos haciendo las cosas bien? A causa de que muchos proyectos fallan al crear productos de alta calidad satisfactorios para los stakeholders y son entregados a tiempo mientras mantienen a los jefes y al equipo contentos, estaremos haciéndolo bien con nuestra manera de actuar si cumplimos el siguiente criterio:

1. **El producto se lanza:** El producto sale de puertas para afuera.

2. **La dirección se mantiene intacta:** Nadie fue despedido por lo que estaba haciendo (o no haciendo).

3. **Las personas en el proyecto quieren trabajar de la misma forma la próxima vez:** Encuentran la manera de trabajar efectiva y agradable.

Cockburn también lista siete principios que son atributos recurrentes de metodologías que tienden a cumplir este criterio de éxito:

1. **La comunicación interactiva y cara a cara es el canal más rápido y barato para intercambiar información:** Haz de las comunicaciones cara a cara la manera de comunicar por defecto, y organiza el lugar de trabajo y los eventos del proyecto o ceremonias para influenciar esta manera de hacer.

2. **El exceso de peso en la metodología es costoso.** La documentación escrita es lenta de producir y resta tiempo para completar el proyecto, por lo tanto, minimiza esta documentación hasta el mínimo nivel suficiente.

3. **Los equipos grandes necesitan metodologías más pesadas:** Así como crece el equipo, se hace más difícil mantener la comunicación por osmosis y el conocimiento tácito, por lo tanto de debe incluir más conocimiento en la documentación.

4. **Grandes actos son apropiados para proyectos con gran criticidad:** Así como la penalización por errores aumenta en un proyecto, deben aumentar los actos y el cuidado tomado en desarrollar el producto. Por ejemplo, fallar al desarrollar un videojuego malgasta el tiempo de ocio del usuario y puede resultar en unas ventas pobre del juego. Fallar al desarrollar un procesador de textos malgasta tiempo de negocio. Así como la criticidad de un producto aumenta a través de los niveles de afectar los recursos esenciales hasta afectar a la vida de las personas (p.e. equipo médico para salvar vidas), el rigor asociado con el producto debe también incrementarse.

5. **Incrementar el feedback y la comunicación reduce la necesidad de entregables intermedios:** Se puede escribir largamente para demostrar que se comprenden los requisitos exigidos por el cliente (entrega intermedia que lleva mucho más tiempo del necesario para desarrollar el producto), o podemos crear algo que muestre que entendemos los requisitos (y que es una parte real de la creación del producto y desplaza el desarrollo del trabajo a lo largo del proyecto).

6. **Disciplina, habilidades, comprender el proceso en contra, formalidad y documentación:** Jim Highsmith nos previene, "No confundas documentación con conocimiento", ya que el conocimiento puede ser tácito (no escrito). Deberíamos también reconocer que "Los procesos no son disciplina". La disciplina es elegir hacer algo de una cierta manera, mientras que el proceso es únicamente seguir unas instrucciones. Y finalmente, "No confundas formalidad con habilidad". En términos de formalidad y habilidad, uno no se refiere al otro. Podemos estar buscando gente inteligente que pueda realizar trabajo de investigación y aplicar sus habilidades y conocimientos, en lugar de aquellos que siguen procesos y formalidad para crear documentación.

7. **La eficiencia es prescindible en las actividades sin cuellos de botella:** Siguiendo los conceptos de la Teoría de las Restricciones, a menos que estemos mejorando la restricción en nuestro sistema, nuestras mejoras no darán beneficios para los resultados globales del proyecto. Por ejemplo, mejorando nuestras actividades de obtención de requisitos o análisis no beneficiará al proyecto si el cuello de botella para el proceso está en la codificación o el testeo. A menos que los procesos de codificación o testeo se mejoren, el proyecto como un todo, no verá ningún incremento en la eficiencia. Por lo tanto, necesitamos ser cuidadosos con las restricciones y realizar nuestras mejoras.

2.7.1.6. Aplicar las nuevas Prácticas Ágiles

Hay veces en nuestros proyectos que necesitamos buscar prácticas nuevas o emergentes para aumentar nuestros propios procesos. En esos momentos, debemos tener en mente las recomendaciones de utilizar formas de trabajar que hayan sido probadas y utilizadas en proyectos en los cuales el producto fue enviado, la cadena de mando quedará intacta, y los equipos quieran utilizar esa manera de trabajar de nuevo. Por lo tanto, si hay un problema en el proyecto, como características que lleven mucho tiempo de desarrollo desde que son elegidas de la cartera hasta que son testeadas, podríamos querer buscar otra forma de hacer las cosas que nos ayude a solucionar el problema.

Seleccionando las prácticas ágiles existentes para aumentar nuestros procesos, en lugar de intentar inventar una por nuestra cuenta, obtendremos los beneficios de una manera de trabajar ya probada, y que ya ha sido depurada de errores. Una vez visto esto, también seguimos la idea de investigar en lugar de inventar, la cual, generalmente, ahorra tiempo. No obstante, cuando elegimos una nueva práctica, debemos verificar antes que el ambiente de trabajo para el que fue diseñada, no es muy diferente del nuestro, ya que estas prácticas son específicas para cada situación.

Volviendo al ejemplo de racionalizar las funciones a través de los procesos de desarrollo, podemos emplear técnicas como Kanban para ayudarnos a solucionar el problema. Limitando el WIP en el proyecto y controlando el ciclo de tiempo, podemos obtener un rendimiento más rápido y un ciclo de tiempo más predecible.

Cuando se piense en aplicar prácticas nuevas, se debe tener en cuenta lo siguiente:

1. **¿Es una solución natural?** ¿Puede el problema solucionarse atendiendo a la causa de raíz? ¿La solución es realmente adoptar una nueva manera de trabajar, o es dejar de hacer lo que nos está llevando al problema?

2. **Investiga tú mismo:** Comprueba la validez de las reclamaciones, busca casos de estudio, y reúnete con gente del proyecto, si es posible. ¿La nueva manera de trabajar tendrá realmente los beneficios que están siendo reclamados?

3. **Prueba pequeñas dosis:** Prueba la nueva manera de trabajar solo en una o dos iteraciones al principio. Comprueba si funciona antes de comprometer al equipo completo o a la empresa. Si la práctica funciona, perfecto; si no lo hace, el daño será menor.

4. **Revisa los efectos colaterales:** Una vez que la práctica se ha implantado durante un tiempo, échale un buen vistazo durante la retrospectiva. ¿Los beneficios superan la molestia de hacerla? ¿Debemos continuar, parar o intentar algo más?

Las siguientes prácticas son conocidas y suelen ser adoptadas por los equipos:

1. **Desarrollo basado en el comportamiento:** Se trata de una manera de testear de dentro a afuera, para múltiples stakeholders.

2. **Puesta en marcha Lean:** Es una manera de trabajar de bajo coste, que emplea revisiones regulares y puntos de pivote.

3. **Opciones reales:** Ésta es una manera de trabajar basada en el cálculo de la teoría de la decisión para evaluar decisiones de tiempos y opciones.

2.7.1.7. Código de conducta ética y profesional PMI

Se verán a continuación algunos aspectos del código.

Primero de todo, hay que entender que el código no es solo para los miembros PMI, es aplicable también para "Los no miembros que solicitan la apertura de un proceso de certificación PMI" Esto quiere decir que aunque no seas un miembro del PMI, al solicitar realizar el examen PMI-ACP, debes aceptar el ser llevado por el código. Veamos qué es lo que estamos aceptando.

El código de conducta ética y profesional PMI es un documento de resalta cuatro áreas de comportamiento profesional por las que debemos regirnos. Cada área del código tiene un conjunto de normas a las que debemos aspirar y un conjunto de normas obligatorias. Las normas a las que debemos aspirar describen los ideales PMI que los profesionales deberían esforzarse por conseguir o aspirar a ellos. Las normas obligatorias son comportamientos que el PMI espera que todos los profesionales sigan.

El otro aspecto del código a tener en cuenta son los procesos de conformidad disciplinares y éticos. Si alguien cree que has violado las normas obligatorias, se supone que debe delatarte, y si tú encuentras que otro miembro del PMI ha violado las normas, se supone que debes delatarle. El comité de ética PMI investigará cualquier informe.

PMI ofrece la versión completa del Código en su web (www.pmi.org). Debería ser leída para familiarizarse con ella. No obstante, el código no tiene un peso muy importante en el examen. El objetivo principal de incluir el código en el examen es remarcar que el código existe.

2.7.1.8. Procesos continuos de mejora

La mejora continua es el proceso constate de mejorar la manera de trabajar en el proyecto y el producto. Este proceso, nunca se completa; en su lugar, continua a lo largo del proyecto al igual que lo hace la comunicación con los stakeholders. La mejora continua es algo que siempre hacemos.

El ciclo de vida ágil, mostrado en la siguiente figura 62, emplea un ciclo continuo de Planificación, Desarrollo, Evaluación, y Aprendizaje.

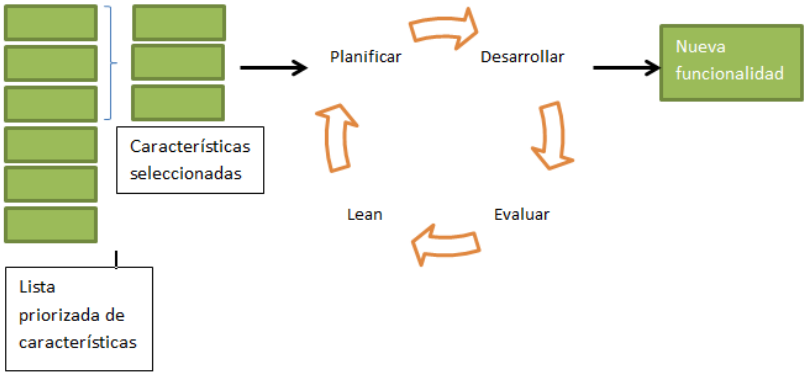


Figura 62: Ciclo continuo de Planificación, Desarrollo, Evaluación, y Aprendizaje. (Fuente: Elaboración propia)

Este ciclo es muy similar al ciclo de Deming "Planificar, Realizar, Chequear, Actuar" para la resolución de problemas y la mejora continua, que vemos en la figura 63:

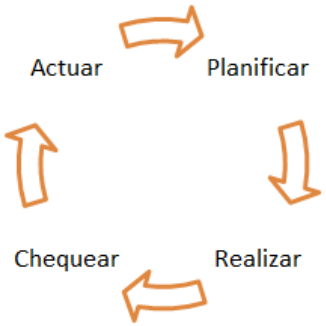


Figura 63: Planificación, Ejecución, Validación, Representar. (Fuente: Elaboración propia)

La mejora continua tiene capas como una cebolla. En los proyectos de desarrollo de software, por ejemplo, la mejora continua tiene lugar en el nivel de código con la programación por pares. Una persona escribe el código, y la otra lo revisa, critica, y sugiere mejoras. Las reuniones diarias discuten el trabajo hecho y cualquier problema o impedimento en el proyecto. Estos problemas se convierten en ítems "a realizar" para el ScrumMaster o el gestor del proyecto, el cual es responsable de eliminar los impedimentos y mejorar el proceso. A un nivel más amplio, los ciclos de demostración, revisión, y retrospectiva que se llevan a cabo cada dos semanas o mensualmente examinan el proceso y buscan mejoras que llevar a cabo en la siguiente iteración.

Mejora continua del producto.

Tal y como las prácticas y procesos del equipo son refinadas de forma iterativa y continua, también lo hace el producto desarrollado. El desarrollo iterativo e incremental es una formad de mejora continua, con feedback del cliente que nos lleva hasta la solución final.

Cuando construimos pequeños incrementos y obtenemos feedback, el producto evoluciona hacia los requisitos reales del negocio. Y a veces, los requisitos reales del negocio pueden ser un poco diferentes de los requisitos iniciales, tal y como el proceso de creación ilumina mejores opciones.

Por este ciclo de desarrollo en pequeños incrementos, revisiones, discutiendo como mejorar, y entonces haciendo más desarrollo y quizás mejorando algunas cosas, el producto o servicio es construido incrementalmente a través de un proceso de mejora continua.

2.7.1.9. Autoevaluación

Las evaluaciones de cómo mejorar no están reservadas únicamente para el proceso y el producto; también son aplicables a los equipos de personas del proyecto. Por lo tanto es una práctica estándar para el equipo reflejar como de bien lo están haciendo y buscar cosas que puedan mejorar.

James Shore ofrece un modelo de autoevaluación de preguntas y puntuaciones, centrado en las prácticas XP. Los equipos pueden utilizar este modelo para medir su rendimiento. El gráfico de puntuación y preguntas mide como rinden los equipos dentro de las siguientes categorías:

- Pensando.
- Colaborando.
- Liberando.
- Planificando.
- Desarrollando.

El cuestionario se completa respondiendo a las preguntas dentro de cada categoría y puntuando las respuestas entre 0 y 100.

La siguiente tabla muestra unas pocas preguntas de la categoría de Planificación:

Preguntas de planificación	Si	No	Práctica XP
¿Todos los miembros del equipo han comprendido que están construyendo, por qué lo están construyendo, y qué consideran los stakeholders un éxito?	25	0	Visión
¿Tiene el equipo un plan para conseguir el éxito?	4	0	Planificación de lanzamiento
¿El equipo busca de forma regular nueva información y la utiliza para mejorar su plan de éxito?	3	0	Planificación de lanzamiento
¿El plan del equipo incorpora a las personas expertas del negocio así como a los programadores, y todos los involucrados creen que el plan es alcanzable?	4	0	El juego de la planificación

Tabla 20: Preguntas de la categoría de Planificación

Una vez que todas las preguntas han sido realizadas y puntuadas, los resultados son plasmados en un diagrama de tela de araña.
Idealmente, el equipo se involucra entonces en analizar el diagrama para identificar qué áreas podrían utilizar la mejora. Los tipos de acciones de mejora que pueden ser realizadas, incluyen formación o añadir más sesiones de trabajo con los representantes de la empresa.

Jean Tabaka también ofrece un modelo para evaluar los atributos de los equipos de alto rendimiento. El modelo de Tabaka investiga las siguientes áreas:

1. **Autoorganización:** ¿Está el equipo autoorganizado en lugar de funcionar a base de órdenes?

2. **Capacitado para tomar decisiones:** ¿Está el equipo capacitado para discutir, evaluar, y tomar decisiones, en lugar de estar supeditado a una autoridad exterior?

3. **Cree en la visión y el éxito:** ¿Los miembros del equipo comprenden la visión del proyecto y los objetivos, y creen verdaderamente que, como equipo, pueden resolver cualquier problema para conseguir estos objetivos?

4. **Equipo comprometido:** ¿Los miembros del equipo, están comprometidos para tener éxito como equipo, en lugar de hacerlo individualmente a cualquier coste?

5. **Creen los unos en los otros:** ¿El equipo tiene la confianza de trabajar de forma continua en mejorar sus habilidades para actuar sin miedo o enfado?

6. **Toma de decisiones participativa:** ¿El equipo está comprometido en la toma de decisiones participativas, en lugar de inclinarse hacia una toma de decisiones autoritaria o ceder a las decisiones de otros?

7. **Llevados por el consenso:** ¿Las decisiones del equipo se rigen por el consenso, en lugar de por lo que diga el líder? ¿Los miembros del equipo comparten sus opiniones con libertad y participan en la decisión final?

8. **Desacuerdo constructivo:** ¿El equipo es capaz de negociar a través de una variedad de alternativas y tomar la que ofrezca el mejor resultado?

Para el examen no es necesario recordar detalles de estos modelos de autoevaluación. Simplemente es necesario entender que son, cuando se usan, y el propósito detrás de ellos. Estas herramientas encajan en el ciclo de adaptación; no solo debemos mejorar nuestros procesos y productos sino también nuestros equipos.

EJERCICIOS

En el Anexo F.7, se encuentra un examen tipo sobre lo tratado en este punto.

3. CASO PRÁCTICO

3.1. OBJETIVOS DEL CASO PRÁCTICO

Una vez vista la teoría sobre los métodos ágiles, el siguiente paso es intentar plasmar en un caso práctico todo lo que hemos visto, de manera que podamos observar que beneficios se obtienen de la aplicación estos métodos.

En concreto se aplicará la metodología ágil a una organización de trabajo ya existente, pero que no lo utiliza, de manera que se puedan comprobar sus beneficios.

En primer lugar se realizará un estudio del funcionamiento actual de la empresa caso del estudio, viendo en que partes se podrían implantar los métodos ágiles.

Veremos que técnicas de los métodos ágiles pueden ayudar a mejorar aún más la eficiencia del trabajo en general y conseguir aumentar la productividad y la calidad del trabajo realizado.

A continuación se expondrá una amplia visión de la empresa, describiendo su historia y centrándonos en describir la actividad objeto del estudio, de manera que podamos establecer una amplia comprensión.

3.2. VISIÓN GENERAL DE LA EMPRESA CASO DE ESTUDIO

La empresa objeto de estudio es el Laboratori General d'Assaigs i Investigacions (LGAI Technological Center S.A., en la actualidad llamada APPLUS+Laboratories).

Se trata de una empresa que tiene como misión el soporte tecnológico a la industria mediante la realización de ensayos de laboratorio, calibración de equipos de medida, certificación, formación técnica e investigación y desarrollo (I+D).

Actualmente el LGAI cuenta con departamentos dedicados a los ensayos de compatibilidad electromagnética, seguridad eléctrica, metrología, acústica, construcción, resistencia al fuego, vibraciones, química, tecnologías del juego, agroalimentario, mecánica y seguridad de la información, así como a la certificación de sistemas.

Como podemos comprobar se trata de una empresa con una variedad de sectores muy amplia, implicada en muchos escenarios tecnológicos diferentes, pero comunicados entre sí.

3.2.1. Historia del Laboratori General d'Assaigs i Investigacions

Los precedentes del Laboratorio General d'Assaigs i Investigacions (LGAI) se remontan al año 1907, cuando el Ayuntamiento y la Diputación de Barcelona crearon el Laboratorio de Investigaciones y Ensayos, situado en unas dependencias de la Escuela Industrial de Barcelona, con el objetivo de prestar un servicio a la industria y al comercio mediante los ensayos y el análisis de productos alimentarios y químicos; materiales de construcción y carbones y aceites industriales.

El año 1921 se unificaron todos los laboratorios que había y se situaron en el edificio de la Escuela Industrial de la calle Urgell de Barcelona, bajo el nombre de Laboratori General d'Assaigs i Condicionament (figura 64). El 1922 se convirtió en un laboratorio general de carácter oficial, dependiendo de la Mancomunidad y organizado en cinco laboratorios: el del Instituto de Electricidad y Mecánica Aplicadas; el del Instituto de Química Aplicada; el de los Servicios Técnicos de Agricultura; el de la Escuela de Adoberia; y el de la Escuela Textil. Durante la República, el Laboratorio pasó a depender de la Generalitat de Catalunya, adscrito a los Departamentos de Cultura (1931-1934 y 1934-1936) y de Economía y Agricultura (1934 y 1936-1939).

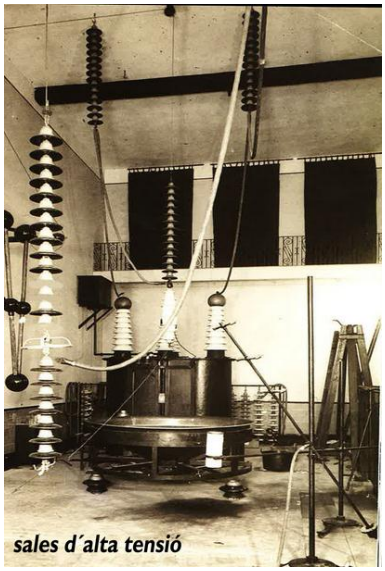


Figura 64: Salas de alta tensión (Fuente: web http://www.applus.com/es/aboutUs/ourHistory)

El año 1935 pasó a denominarse Laboratori General d'Assaigs i Investigacions. Con la finalización de la guerra civil, la Diputación de Barcelona volvió a asumir la titularidad del Laboratorio. A partir de 1944, a raíz de la creación del Instituto de Investigaciones Técnicas, el Laboratorio se dedicó exclusivamente al servicio público de ensayos y análisis, mientras que la investigación técnica pasaba a ser competencia del nuevo Instituto.

El año 1984 fue transferido a la Generalitat de Catalunya, por el Decreto 44/1984, de 15 de febrero, quedando adscrito al Departamento de Industria y Energía, mientras que por la Ley 23/1984 de 28 de noviembre, se creaba de nuevo el Laboratorio como entidad de derecho público con personalidad jurídica propia. Desde el año 1987 las instalaciones del LGAI están situadas en Bellaterra, junto al campus de la Universidad Autónoma de Barcelona.

Historia moderna de la compañía en los últimos 10 años, tras su privatización.

El año 2002 se inició el proceso de privatización del Laboratorio con la constitución de la sociedad LGAI Technological Center, S.A. que desde 2003 pertenece al grupo de certificación Applus+ (www.applus.com).

En figura 65 podemos ver el esquema general del grupo Applus+, y como queda ubicado en la actualidad el LGAI, llamado actualmente Applus+ Laboratories de manera comercial, aunque su nombre legal continúa siendo LGAI Technological Center S.A.

Figura 65: Esquema general del grupo APPLUS+ (Fuente: http: // www.applus.com /es /aboutUs /ourHistory)

Evolución del grupo Applus+ en los últimos años.

En el año 2004, el grupo Applus+, logra facturar 213 millones de euros con una cifra de negocio del 30% fuera de España. Applus+ compra SOLUZIONA Calidad y Medio Ambiente (Norcontrol y Novotec) y se convierte en la mayor compañía de inspección industrial y medioambiental y asistencia técnica de España.

En el año 2005 se adquiere la empresa pública danesa de inspección de vehículos, Applus+ Bilsyn, que hoy cuenta con 125 centros de inspección. Applus+ se convierte en líder mundial en inspección de vehículos.

En el año 2006, Applus+ factura 540 millones de euros y logra presencia internacional en más de 40 países. Adquiere el grupo holandés Röntgen Technische Dienst (RTD), líder en ensayos no destructivos e inspección para la industria petrolera y del gas en Europa. Applus+ se posiciona como líder en servicios de ensayos para la industria petrolera y del gas. Adquiere K1, el segundo operador de inspección de vehículos en Finlandia. K1 realiza el 20% de inspecciones en el país.

En 2007 “The Carlyle Group” se convierte en el principal accionista de Applus+, con más del 70% de participación en el grupo. A través de distintas adquisiciones en Finlandia de operadores de ITV, Applus+ alcanza una red de 45 estaciones y se confirma como segundo operador del país.

En el año 2008 Applus+ logra facturar 837 millones de euros además de un fuerte proceso de internacionalización. Adquiere MBI, líder en ensayos no destructivos del Reino Unido. Adquiere JanX Integrity Group, la empresa líder en el mercado americano de ensayos no destructivos.

En el año 2009, logra la concesión del 100% de las ITV en Irlanda (National Car Testing). El NCT realiza en torno a 1,4 millones de inspecciones de vehículos al año en 46 estaciones por todo el país.

En el año 2011, Applus+ logra 1206 millones de euros de facturación en más de 65 países.

Adquiere la empresa malaya Velosi, multinacional de servicios de inspección, certificación, ensayos y servicios de selección de personal para la industria petrolera y del gas, lo que consolida su posición de liderazgo global en el sector.

Adquiere la empresa alemana BKW, que ofrece servicios de ensayos y caracterización de materiales para la industria aeronáutica europea.

Desde el 2002 y gracias a un fuerte proceso de internacionalización, logra obtener el 70% de la cifra de negocio fuera de España repartida por los 5 continentes.

Refuerza su posición en Australasia con la adquisición de la compañía australiana John Davidson and Associates (JDA), multinacional de referencia en servicios de inspección en la región de Asia-Pacífico.

3.2.2. Descripción de la actividad de la empresa objeto del caso de estudio

Nos centraremos concretamente en el estudio del departamento de Seguridad Eléctrica de LGAI (Applus+Laboratories).

El departamento de Seguridad Eléctrica se encuentra integrado en el centro E&E (Electrical and Electronics). El centro de E&E incluye otros laboratorios además del de Seguridad Eléctrica, como son el de Vibraciones, Acústica, Climáticos, Compatibilidad Electromagnética.

A continuación, veremos una descripción de las actividades que se llevan a cabo en dicho centro, de manera que podamos entender, primero, las características de la actividad que se lleva a cabo, y en la que implementaremos la metodología ágil.

Definición de Seguridad Eléctrica

En primer lugar, se proporciona una definición de que se entiende como seguridad eléctrica y su importancia en la industria actual.

La seguridad eléctrica es el campo de la ingeniería eléctrica responsable del buen uso de la energía eléctrica y el mantenimiento de sistemas eléctricos y electrónicos para que sean seguros para las personas.

Todo el material eléctrico comercializado (entendiendo por «material eléctrico» cualquier clase de material eléctrico destinado a utilizarse con una tensión nominal comprendida entre 50 y 1000 V en corriente alterna y entre 75 y 1500 V en corriente continua) debe cumplir con la Directiva de Baja Tensión (2006/95/CE; figura 66).

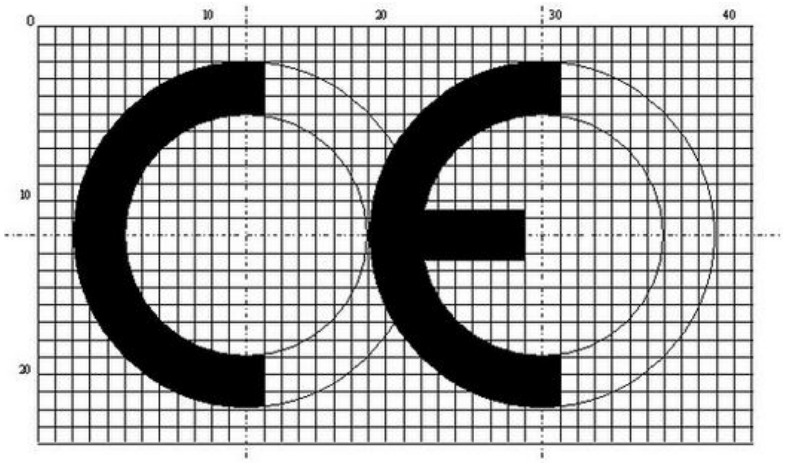


Figura 66: Símbolo marcado CE (Fuente: Web)

En dicha directiva se dan unos requisitos esenciales que se deben cumplir de manera que el producto comercializado sea seguro y no implique riesgos para la persona que lo está utilizando, figura 67.

Figura 67: Distintos productos cubiertos por la Directiva de Baja Tensión. (Fuente: Web)

En la directiva se establece que una forma de obtener presunción de conformidad con los requisitos esenciales incluidos en ésta, es dar cumplimiento a una norma de seguridad eléctrica armonizada aplicable al material eléctrico que se quiere comercializar.

Para la gran mayoría de productos eléctricos existe una norma de seguridad eléctrica aplicable armonizada que da por lo tanto presunción de conformidad a los requisitos esenciales recogidos. Hay que destacar que no es en absoluto obligatorio el cumplir con una norma de Seguridad Eléctrica, lo que es obligatorio es cumplir con la Directiva de Baja Tensión.

Por lo tanto, muchos fabricantes utilizan las normas de Seguridad Eléctrica para determinar si su producto cumple o no con los requisitos esenciales de la Directiva de Baja Tensión.

Aquí es donde entra en escena el departamento de Seguridad Eléctrica de LGAI, el cual ofrece sus servicios y capacidad para ensayar productos de acuerdo a las normativas de Seguridad Eléctrica correspondientes.

Las normas de seguridad eléctrica cubren aspectos tales como:

- ✓ Protección contra el choque eléctrico.
- ✓ Efectos del calentamiento de los equipos durante el uso normal.
- ✓ Riesgos físicos.
- ✓ Efectos de condiciones anormales durante el uso del equipo.
- ✓ Protección contra el agua, el polvo y la entrada de cuerpos sólidos.
- ✓ Marcado y documentación de seguridad.
- ✓ Verificación de aislamientos.

Más adelante se verá en detalle el proceso de un proyecto de seguridad eléctrica y se detallará cada uno de estos aspectos.

Dando una visión general, podríamos decir que durante un proyecto de seguridad eléctrica, hay una parte del tiempo que se destinada a:

- ✓ Revisión documental del marcado y de las instrucciones.

En todas las normas de seguridad eléctrica se incluye un capítulo en el que se dan los requisitos sobre el marcado y las instrucciones de uso que deben acompañar a todo producto comercializado. Una de las tareas es, por lo tanto, revisar que todas las indicaciones que exige la norma aplicada se encuentran marcadas sobre el producto, y que en las instrucciones de uso, instalación y mantenimiento se incluyen todas las instrucciones de seguridad indicadas por la norma y necesarias para que el uso del producto por parte del usuario final sea seguro y se minimicen los posibles riesgos.

Dentro del marcado necesario que debe aparecer sobre el producto, se incluyen por ejemplo, las características de la alimentación del producto (tensión, frecuencia, potencia), la marca identificativa del producto, el nombre de modelo o referencia que lo identifica.
Las instrucciones de seguridad a incluir en los manuales, dependen del tipo de producto, pero en general se deben dar indicaciones para el uso del producto de manera segura, y advertir de los peligros que puedan llegar a ocurrir durante su utilización.

- ✓ Revisión documental de certificados de componentes.

Otro capítulo habitual en las normas de seguridad eléctrica, es el de componentes. Dentro de este capítulo se dan los requisitos que deben cumplir los componentes que forman el producto evaluado. Por ejemplo, si el producto evaluado es una lámpara de mesa, aplicaremos la norma de producto para requisitos de seguridad eléctrica (UNE-EN 60598-1 junto a su parte 2 particular UNE-EN 60598-2-4), de forma global al producto, pero dentro del capítulo de componentes, se exigirá que los componentes que forman la luminaria (cable de alimentación, clavija, portalámparas, interruptor…) cumplan con sus respectivas normas de producto de forma individual. La norma da la opción de que se pueda demostrar el cumplimiento de esos componentes con sus respectivas normas, a través de certificados/informes de ensayo emitidos por laboratorios acreditados e independientes (es decir, no es necesario que durante los ensayos globales a la luminaria se deban ensayar todos sus componentes además, pueden haberse ensayado con anterioridad). En este caso, por lo tanto, la tarea consistirá en revisar los certificados / informes de ensayo proporcionados por el fabricante de la luminaria y comprobar que las normas aplicadas en ellos son correctas y que las versiones de esas normas son vigentes.

- ✓ Realizar inspecciones visuales de la construcción del equipo.

Otro de los capítulos habituales en las normas de seguridad eléctrica es el de construcción. En este capítulo se indican los requisitos constructivos que debe satisfacer el producto ensayado. Por ejemplo, se indica que los cables no deben dañarse en el interior del equipo debido a aristas cortantes, que las partes metálicas de los equipos deben conectarse a tierra si pueden quedar en tensión por un defecto de aislamiento. También estarían aquí incluidas todas las verificaciones realizadas para comprobar que no hay un peligro de choque eléctrico para el usuario del producto, con la ayuda de calibres de ensayo que simulan el dedo de una persona. Durante los ensayos se deberá comprobar que la muestra ensayada cumple con todo lo indicado por la norma.

- ✓ Realización de pequeños ensayos de corta duración.

Hay una serie de ensayos sencillos que se realizan para comprobar distintos requisitos de la norma, y que aparecen detallados en los distintos capítulos. Estos ensayos no implican una preparación o tiempo de ensayo considerable (habitualmente entre 15 minutos a 1 hora). Por ejemplo, en aparatos que utilizan una clavija para su conexión a la red eléctrica y contienen condensadores, se debe comprobar que si el usuario desconecta el aparato y toca los pines de la clavija, la descarga de esos condensadores no pueda provocarle un daño, para lo cual hay que medir la tensión entre los pines de la clavija un segundo después de la desconexión de la red. Otra prueba habitual es comprobar que la estabilidad del equipo es adecuada y no es susceptible de volcar durante el uso normal del mismo, para lo cual se coloca el equipo en un plano inclinado en todas las posiciones en las que será utilizado.

- ✓ Realización de ensayos largos y más complejos.

Las normas comprenden una serie de ensayos de una complejidad mayor de lo que hemos visto en los puntos anteriores, mediante los cuales verificamos una serie de requisitos, como que las temperaturas de las partes externas del equipo no superen ciertos límites, de manera que el usuario del producto no pueda quemarse al utilizarlo, que la potencia consumida por el equipo no sea excesiva y sea acorde a lo indicado en la placa de características del mismo, que los diferentes materiales plásticos que forman el equipo no sean inflamables y no puedan provocar un incendio a causa de un mal funcionamiento de un componentes interno del equipo, que si el equipo se utiliza de manera anormal o se produce un fallo de alguno de sus componentes, esto no derive en un peligro para el usuario del mismo, entre otros. La realización de estos ensayos requiere de un tiempo de preparación y de ensayo superior al que veíamos en el punto anterior (entre más de una hora y días en ocasiones).

3.3. DESCRIPCIÓN DE LA ORGANIZACIÓN ACTUAL EN EL CENTRO DE SEGURIDAD ELÉCTRICA

En esta parte del caso práctico describiremos la organización actual del centro de Seguridad Eléctrica y como se estructuran los proyectos.

3.3.1. Descripción de la organización de trabajo

El trabajo en un laboratorio de ensayos de Seguridad Eléctrica, se organiza en forma de proyectos. En cada proyecto se realizan los ensayos y verificaciones necesarias para dar conformidad a una determinada norma sobre un producto determinado.

Para este caso práctico se describirá la parte de la organización que se encarga del desarrollo técnico del proyecto, aunque también existe toda una parte de la organización de carácter comercial, encargada de hacer llegar los proyectos para su ejecución.

En la figura 68, se puede observar el organigrama del centro, con cada una de las capas que lo forman:

Figura 68: Organigrama actual de la parte productiva. (Fuente: Elaboración propia)

Como podemos observar, se trata de un organigrama vertical, es decir cada bloque o puesto de trabajo depende directamente del bloque anterior.

A continuación, se definirán las funciones actuales de cada uno de los puestos, de manera que se tenga una idea clara de esas funciones y de las responsabilidades de cada uno de ellos.

3.3.2. Funciones de los diferentes puestos

3.3.2.1. Área comercial.

Dentro del área comercial englobaríamos a todos las personas que se encargan de gestionar las peticiones de oferta de los clientes. Son los encargados de contactar con el potencial cliente y realizar la oferta económica en función de sus necesidades. Realizan también el seguimiento de la oferta una vez enviada al cliente y negocian el precio ofertado en caso de que sea conveniente el hacerlo. Son también responsables de desarrollar la línea de negocio en colaboración con el product manager, buscando nuevos clientes y productos.

3.3.2.2. Director de operaciones

El director de operaciones se encarga de gestionar todo lo necesario para que los proyectos puedan salir adelante. Gestiona los equipos, materiales y personal asociados al laboratorio, y soluciona cualquier problema que pueda surgir y que impida el normal desarrollo del proyecto. Además está entre sus cometidos el buscar la mejora del rendimiento de los proyectos mediante nuevos métodos de trabajo y optimizaciones en el equipo de trabajo.

Es el responsable de todas las personas debajo de él en el organigrama.

3.3.2.3. Product manager

El product manager es el responsable de una determinada línea de producto (en este caso la línea seria el laboratorio de ensayos de seguridad eléctrica). Se encarga del desarrollo comercial de la misma y de la gestión económica. Rinde cuentas al manager general del estado de la facturación y producción global de la línea de negocio.

3.3.2.4. Test manager

El test manager es el responsable técnico de los trabajos realizados y se encarga de controlar la facturación y producción de los proyectos. Debe ser un experto en los trabajos a realizar ya que se encarga también de solucionar dudas sobre cómo realizar los ensayos o interpretar la normativa.

Tiene trato directo con los clientes para temas técnicos y comerciales.

3.3.2.5. Técnicos de Laboratorio.

Los técnicos de laboratorio son quienes realizan el trabajo técnico y quienes poseen el conocimiento de cómo realizar las tareas.

Planifican la secuencia de ensayo en función de la normativa a ensayar.

Tienen trato directo con los clientes para cualquier tema técnico.

3.3.3. Flujo de trabajo

A continuación se describirá el flujo de trabajo actual para un proyecto de seguridad eléctrica, desde que entra la petición de oferta hasta que se entrega el informe de resultados al cliente:

1. En primer lugar entra la petición de oferta del cliente a través del área comercial, bien sea el propio cliente el que realiza el contacto y la petición o por una acción comercial realizada desde este departamento. Como se ha comentado anteriormente, no es en absoluto obligatorio cumplir con una determinada norma, pero si que lo es cumplir con la Directiva de Baja Tensión, siendo la mejor manera de hacerlo cumplir con la norma de producto aplicable. Por lo tanto, es frecuente que sean los propios clientes quienes contacten directamente con el área comercial para poder ensayar sus productos.

2. La oferta es analizada y en caso de ser necesaria ayuda para su realización, el Product manager encarga la tarea al Test Manager, el cual puede a su vez requerir la ayuda de alguno de los Técnicos de Laboratorio expertos en la normativa demandada por el cliente. Los aspectos analizados para la confección de la oferta son:

- El tipo de producto que se pide ensayar; de manera que sea posible seleccionar la norma adecuada.
- Particularidades del producto concreto a ser ensayado; no es igual ensayar un cargador para teléfono móvil que un rack de telecomunicaciones compuesto de varios armarios con decenas de circuitos electrónicos en su interior, aunque a los dos les aplica la misma norma de seguridad eléctrica (UNE-EN 60950-1). Por lo tanto es necesario hacer constar toda una serie de condicionantes en la oferta para ajustar el precio final al esfuerzo y tiempo reales necesarios para ensayar el producto.
- Equipos e instalaciones necesarios para los ensayos; una vez analizado el tipo de producto y sus particularidades, es necesario comprobar que se dispone de todo lo necesario, a nivel de equipos, para realizar los ensayos, o si será necesario subcontratar algo, de manera que podamos reflejar esto en el precio final de la oferta.
- Número de horas persona necesarias para llevar a cabo el proyecto completo; en el caso de productos estándar, el número de horas se encuentra definido para cada norma, pero si el producto a ensayar tiene alguna particularidad especial, es necesario recalcular este número de horas, teniendo en cuenta todas estas particularidades.

Para analizar todos estos aspectos, como se puede ver, es necesario conocer las normas y sus requisitos, por lo que es normal que el área comercial se apoye en los expertos del área de producción (responsable técnico y técnicos de laboratorio), para la confección de las ofertas.

3. Una vez la oferta es aceptada por el cliente, el product manager entrega la petición de trabajo al Test Manager, el cual asigna a uno de los Técnicos de Laboratorio para la realización de los trabajos contenidos en ella. Esta asignación se realiza en función de la carga de trabajo de los técnicos, del plazo de ejecución que se haya pactado con el cliente, pero también se busca realizar esta asignación en función del nivel de conocimiento del técnico en la norma a aplicar. Si es posible asignar a un técnico con un mayor nivel de experiencia en la norma concreta a aplicar, se ganará en rapidez a la hora de ejecutar los trabajos.

4. El técnico de laboratorio planifica los trabajos contenidos en la oferta y se pone en contacto con el cliente para obtener toda la información necesaria para la realización de éstos. Cada técnico es responsable de organizar su tiempo, de manera que se pueda llegar a cumplir con el plazo ofrecido al cliente. Si a la hora de realizar esta planificación descubre que nos es posible cumplir con ese plazo, debe avisar de inmediato para que se tomen las medidas oportunas (apoyos puntuales de otros técnicos para mejorar la velocidad del proyecto, por ejemplo). El técnico es la cara visible para el cliente una vez superada la fase comercial y es con él con quien se realizarán todas las comunicaciones a partir de ese momento, a excepción de que sea necesario tratar algún tema económico debido al desarrollo del proyecto.

5. Una vez finalizados los trabajos contenidos en la oferta, el técnico de laboratorio genera un informe de ensayos con los resultados obtenidos. En el informe final se recogen los resultados para cada una de las pruebas y su veredicto respecto a los requisitos de la norma (conforme o no conforme). Los informes generados pueden ser favorables o no favorables, en función de si el producto ensayado cumple o no con los requisitos de la norma aplicada. En el caso de que durante los ensayos surjan no conformidades, estas se comunican al cliente, quien tiene la oportunidad de solucionarlas antes de la emisión del informe final, de manera que sea posible emitir un informe conforme a la normativa. Solo se emiten informes no conformes en el caso de que el cliente no pueda solucionar las no conformidades encontradas debido a su gravedad, o que se pacte con el cliente emitir el informe con esas no conformidades.

6. Una vez generado el informe por el técnico, se designa un revisor el cual verifica que la información contenida en el informe de ensayos es correcta. Este revisor, puede ser el propio test manager, o un técnico de laboratorio con la cualificación adecuada en la norma ensayada. Durante esta revisión se generarán diversas versiones del informe final, hasta que se llegue a la versión definitiva que cumpla con los requisitos de calidad exigidos.

7. Una vez revisado el informe, el test manager (que a su vez es el responsable técnico) aprueba el informe y lo firma. Para esta aprobación, el responsable técnico vuelve a realizar una revisión del informe, con lo que cualquier punto no encontrado en la revisión anterior es aquí detectado. Si todo es correcto el informe queda listo para su emisión.

8. El siguiente paso es enviar el informe al cliente junto con la factura de los trabajos realizados. Este envío se realiza desde el área comercial, cerrando así el círculo del proyecto.

En la figura 69 puede verse un resumen de lo que acabamos de describir.

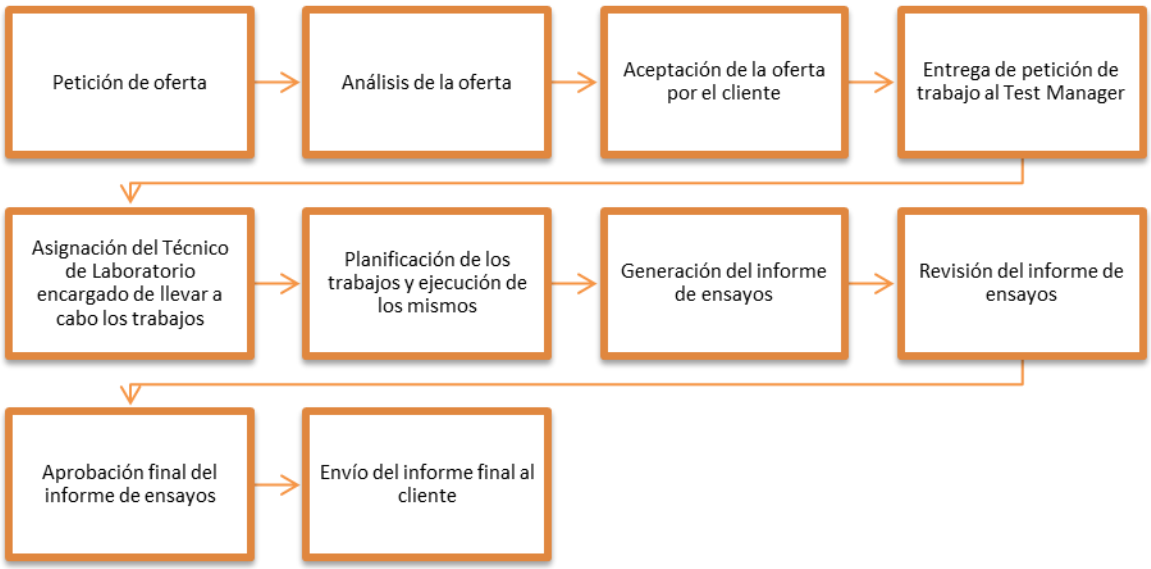


Figura 69: Flujo de trabajo actual. (Fuente: Elaboración propia)

3.3.4. Tareas adyacentes al flujo de trabajo

Durante todo el proceso contenido en el flujo de trabajo, aparecen una serie de tareas adyacentes, necesarias para el desarrollo del proyecto. Se detallan estas tareas a continuación.

1. Para la realización de la oferta, es necesario mantener un contacto constante con el cliente para obtener toda la información necesaria y plasmar en la oferta exactamente lo que necesita el cliente, con lo cual el comercial, y el test manager y el técnico de laboratorio si ha sido necesaria su ayuda, deben destinar parte de su tiempo en ello.

2. Una vez enviada la oferta, es necesario realizar un seguimiento de la reacción del cliente y conocer su opinión de cara a mejorar las posibilidades de aceptación. Esta tarea debe realizar el comercial, y debe ser capaz de ver si el cliente considera que el precio es elevado, de manera que se pueda gestionar un descuento o ajuste, en caso de que se considere necesario.

3. Una vez aceptada, la oferta debe introducirse en el sistema y posicionarse en la cola de tareas pendientes. Esta tarea es realizada por el product manager, quien avisa al test manager de la nueva entrada.

4 Si el test manager no ha participado en la confección de la oferta, debe estudiarla para comprender los trabajos a realizar. Este punto es importante, y puede requerir de bastante tiempo, en función de la complejidad de la oferta.

5. Una vez comprendida la oferta, el test manager asigna al técnico de laboratorio quien debe destinar un tiempo también a estudiar la oferta de cara a comprender el trabajo a realizar (lo cual se realiza con la ayuda del test manager).

6. Durante los ensayos, el técnico se encarga de todo el contacto con el cliente en lo relativo a los ensayos a realizar:

- Pide toda la información necesaria.
- Comunica los incumplimientos hallados durante los ensayos.
- Asesora al cliente en cómo resolver esos incumplimientos.
- Es responsable de utilizar los equipos adecuados y comprobar que se encuentren dentro de los periodos de calibración establecidos.

7. El Test Manager debe cuidar de que los equipos de ensayo estén calibrados y en un estado adecuado para su uso. En caso de encontrar alguno con cualquier problema debe avisar a los técnicos para restringir su uso. También debe validar los certificados de calibración antes de que el equipo sea declarado apto para su uso

3.4. IMPLANTACIÓN DE METODOLOGÍA ÁGIL

Una vez que hemos descrito la actividad de la empresa estudiada, la organización actual y hemos visto en que consiste la actividad que en ella se realiza, veremos cómo podemos utilizar las técnicas descritas en las metodologías ágiles y qué ventajas podríamos obtener de su utilización.

Como hemos visto, la metodología ágil se centra en las personas que poseen el conocimiento de cómo realizar el trabajo y busca por encima de todo el facilitar su trabajo.

En primer lugar, identificaremos a los personas del esquema anterior que tienen el conocimiento de cómo llevar a cabo los proyectos.

3.4.1. Trabajadores con el conocimiento para realizar los trabajos

Como hemos visto, en la parte final del organigrama, se sitúan los técnicos de laboratorio, que son los que en último lugar realizan la parte técnica del trabajo, llevan a cabo los ensayos y verificaciones, y son los expertos en este campo.

Por lo tanto son los principales trabajadores con el conocimiento para realizar los trabajos.

Por encima de ellos, se encuentra el test manager, quien también posee el conocimiento de cómo realizar los trabajos, aunque no los realice. Es el responsable técnico y por lo tanto el responsable de que todo se realice de la manera adecuada, y siguiendo los estándares de calidad necesarios.

Está por lo tanto, también incluido en el grupo de trabajadores con el conocimiento.

Por lo tanto, en nuestro esquema, los trabajadores poseedores del conocimiento son los técnicos de laboratorio y el test manager, como vemos en la figura 70.

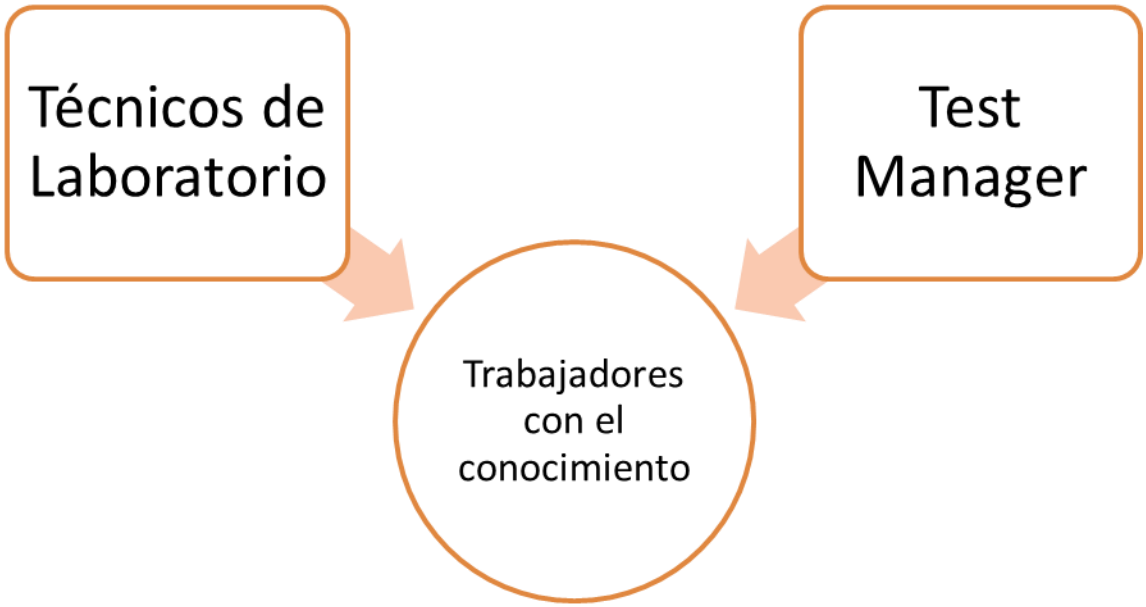


Figura 70: Trabajadores con el conocimiento. (Fuente: Elaboración propia)

3.4.2. Liderazgo de servicio

Una vez que se ha detallado qué trabajadores van a realizar las tareas y trabajos contenidos en la oferta económica, se han de identificar los líderes que gestionarán a estos trabajadores. En este caso concreto, el líder inmediato y que mayor influencia tendrá sobre los técnicos es el test manager. Por lo tanto, quedará identificado como el líder ágil del proyecto.

Según lo indicado por la metodología ágil, este líder (el test manager), deberá adoptar el rol de "Líder servidor" que vimos en la descripción de los métodos ágiles.

El líder servidor, tendrá las siguientes funciones:

1. Proteger al equipo de las interrupciones.
2. Eliminar los impedimentos para progresar.
3. Mantener la visión del proyecto en mente de todos los trabajadores.
4. Procurar que el equipo disponga de todo lo necesario.

Por otro lado, el director de operaciones y el product manager, deben ser líderes servidores del test manager y de los técnicos de laboratorio en menor medida.

Estos líderes deberán delegar en los trabajadores que poseen el conocimiento, de manera que ellos mismos puedan desarrollar su trabajo, auto organizándose, tal y como los métodos ágiles promueven.

De esta manera podrán desarrollar sus funciones sin necesidad de tener que preocuparse de dirigir de manera constante el día a día de los trabajadores. Esto ayudará a que la calidad del resto de sus tareas aumente, al poder dedicarles más tiempo. Por ejemplo, una de las tareas del test manager es comprobar que los ensayos se realizan de acuerdo a los estándares de calidad establecidos, al disponer de más tiempo, por no tener que gestionar a los técnicos, podrá dedicarse a esta tarea de forma más exhaustiva.

En la figura 71 se puede observar un organigrama que simplifica el anterior, teniendo en cuenta el concepto de dirección de servicio o liderazgo de servicio que se acaba de mostrar:

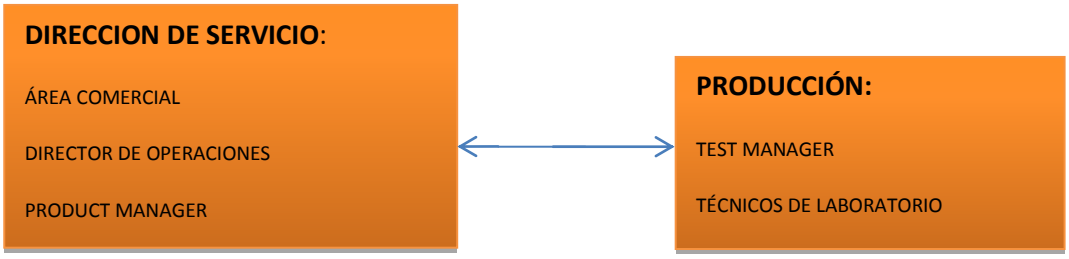


Figura 71: Organigrama basado en métodos ágiles. (Fuente: Elaboración propia)

Se trata de un organigrama más plano en el cual se le da la misma importancia a la parte productiva y directiva. Con las flechas de doble sentido, se quiere enfatizar que la comunicación en las dos direcciones debe ser importante y que deben trabajar juntos.

Una vez que hemos reubicado a cada rol en el nuevo organigrama se verá que técnicas de las estudiadas en los métodos ágiles pueden ayudar a mejorar el sistema de trabajo original.

3.4.3. División del proyecto en iteraciones

3.4.3.1 Partes del proyecto y su importancia

Como hemos visto, los métodos ágiles dividen el trabajo del proyecto en iteraciones. En cada iteración, se realiza una parte determinada del trabajo intentando que sea la que aporta más valor, para de esta manera intentar asegurar que como mínimo completaremos las partes más valiosas gracias a la priorización de las mismas.

En un proyecto de seguridad eléctrica, todas las partes del mismo son importantes. Por lo tanto, la decisión sobre cómo dividir el trabajo en distintas iteraciones y priorizarlas por valor aportado requiere un análisis un poco más delicado.

En este caso concreto, una buena opción es identificar que partes del proyecto implicarían un mayor número de cambios a posteriori en el caso de descubrir una parte del equipo analizado que no cumpla con los requisitos de la norma ensayada. Es decir, si se descubre al final de los ensayos que la placa de circuito impreso principal del equipo no está diseñada de acuerdo a los requisitos exigidos por la norma, el fabricante se verá obligado a rediseñarla para lograr el cumplimiento. Este cambio, puede hacer que sea necesario repetir varios capítulos de la norma que ya han sido realizados, ya que el resultado de los ensayos contenidos en ellos se verá afectado por los cambios en la placa de circuito impreso. Si en cambio realizamos el análisis de la placa de circuito impreso al inicio de los ensayos y descubrimos que debe ser modificada, podremos parar los ensayos en ese punto y no tendremos que repetir nada, ya que no habremos realizado ninguna prueba.

Por lo tanto, como norma, programaremos este tipo de ensayos como prioritarios, de manera que podamos descubrir cuanto antes posibles incumplimientos de la normativa que lleven a cambios severos en el diseño del equipo.

Por lo general, las normas de seguridad eléctrica se dividen en los siguientes ensayos o verificaciones:

1. Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…).
2. Protección contra el choque eléctrico.
3. Medida de potencia y corriente.
4. Ensayo de calentamiento normal.

5. Ensayo de medida de corrientes de fuga y rigidez dieléctrica.
6. Resistencia a la humedad.
7. Funcionamiento anormal.
8. Estabilidad y riesgos físicos
9. Resistencia mecánica.
10. Construcción.
11. Componentes.
12. Medida de distancias en el aire y líneas de fuga.
13. Resistencia al calor y al fuego.

A continuación se expondrá una breve descripción de cada uno de estos puntos:

1. Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…).

Como se vio anteriormente, uno de los capítulos habituales de las normas de seguridad eléctrica es el capítulo de marcado y documentación que ha de acompañar al equipo. Este capítulo regulará que información debe estar marcada sobre el producto (características eléctricas, nombre o marca del fabricante, modelo de referencia,…) y que información de seguridad debe ser incluida en las instrucciones que acompañan al producto (como instalar, utilizar y mantener de forma segura el producto).

2. Protección contra el choque eléctrico.

En el capítulo de choque eléctrico se describen las verificaciones visuales y ensayos que deberá satisfacer el producto, para asegurar que si se usa de acuerdo a las instrucciones del fabricante, no hay riesgo de que el usuario sufra un choque eléctrico. Entre otras pruebas, se aplicaran calibres de ensayo que simulan un dedo humano a la superficie exterior del producto y se intentará hacerlos contactar con partes a tensión peligrosa. Una no conformidad en este apartado, puede hacer que sea necesario rediseñar parte de la envolvente del producto, por ejemplo.

3. Medida de potencia y corriente.

La potencia y corriente consumida por el producto cuando se encuentra operando en las condiciones normales de funcionamiento más desfavorables no debe diferir de la indicada por su marcado más allá del rango indicado por la norma. Una no conformidad en este punto, puede hacer que sea necesario modificar el marcado del equipo (solución más sencilla) o, si esto no es posible por la razón que sea, sea necesario variar la potencia o corriente consumida por el producto.

4. Ensayo de calentamiento normal.

En este capítulo evaluamos las temperaturas externas e internas del producto durante su funcionamiento normal (ver equipo utilizado en la figura 72).

La norma indica que todos los componentes incluidos en el producto deben trabajar dentro de sus especificaciones correspondientes (es decir, que si la temperatura ambiente de funcionamiento máxima de la fuente de alimentación interna es de 40ºC, se debe verificar que el ambiente en el cual está trabajando en el producto final no excede de esos 40ºC). Las medidas se realizan con un registrador de temperaturas, como el de la figura 70.

La norma también da límites para las temperaturas de las superficies externas accesibles a los usuarios del producto, de manera que estos no puedan llegar a dañarse con ellas.

Si durante este ensayo descubrimos componentes trabajando fuera de especificaciones o partes externas del producto con temperaturas excesivas, el fabricante tendrá que cambiar componentes, o modificar la ventilación del equipo, o la distribución de las fuentes de calor internas para separarlas de las partes externas… Todo esto llevará tener que repetir partes de la norma.

Figura 72: Registrador de temperaturas utilizado en la prueba de calentamiento normal. (Fuente: Web)

5. Ensayo de medida de corrientes de fuga y rigidez dieléctrica.

En este ensayo se miden dos parámetros relacionados con la protección contra el choque eléctrico. En primer lugar se mide la corriente de fuga existente durante el funcionamiento normal del equipo. Esta corriente es la corriente que se escapa a través de los aislamientos, filtros antiparasitarios, etc, presentes en el equipo, y que atravesará a cualquier persona que toque la superficie del equipo y no se encuentre bien aislada de tierra (por ejemplo, toque la superficie del equipo con una mano, mientras con la otra toca una estantería metálica cercana, o el mismo suelo). La norma limita esta corriente a unos valores considerados seguros que no se deben exceder.

En segundo lugar, se realiza una prueba de rigidez dieléctrica a alta tensión para verificar que los aislamientos y distancias en el aire que se ha utilizado para proteger al usuario del choque eléctrico, se encuentran bien dimensionados. Por ejemplo, se aplicaran 1.500 voltios entre los dos polos de la alimentación del equipo y las partes metálicas de la envolvente que se encuentren conectadas a la tierra de protección. Durante la aplicación de esta tensión, no debe producirse conducción de corriente entre las partes separadas.

En caso de descubrir una no conformidad en alguno de estos ensayos, puede ser necesario rediseñar el aislamiento en cuestión, pero no tiene por qué ser necesario rediseñar partes muy importantes del equipo.

6. Resistencia a la humedad.

En esta prueba se coloca el producto ensayado en ambiente saturado de humedad (93%), a temperatura de laboratorio (25ºC), durante 48 horas. Seguidamente se lleva a cabo el ensayo de rigidez dieléctrica y el de corrientes de fuga descrito anteriormente. Una no conformidad en este ensayo, tendrá los mismos efectos que en el caso anterior.

7. Funcionamiento anormal.

Durante los ensayos de funcionamiento anormal, comprobamos que el equipo continúa siendo seguro incluso en el caso de que alguna de sus partes no funcione correctamente o el usuario haga un uso incorrecto de él.

Ejemplos de funcionamiento anormal que se someterá a ensayo son por ejemplo, el bloqueo de motores, el fallo de componentes de los circuitos electrónicos en modo cortocircuito o circuito abierto, cubrir oberturas de ventilación en las envolventes, variaciones en la tensión de alimentación que provoquen potencias por encima de lo normal, simular roturas de juntas de estanqueidad en equipos conectados a la red de agua, sobrecarga de conectores externos (como USB’s por ejemplo), etc.

El requisito para este tipo de pruebas, por lo general, es que el equipo no emita llamas, material fundido, gases venenosos o inflamables, que su envolvente se deforme llegando a ser accesibles las partes activas peligrosas, etc. Una no conformidad en este tipo de ensayo puede llevar a tener que rediseñar las protecciones contra todos estos efectos implementadas en el diseño del equipo.

8. Estabilidad y riesgos físicos

Durante estos ensayos, se comprueba que el producto sea estable cuando está colocado sobre la superficie de uso, y no pueda volcar. Para ellos se coloca el equipo en un plano inclinado 10º respecto a la horizontal. También se comprueba que el equipo no pueda dañar físicamente al usuario por presentar aristas cortantes u otras superficies capaces de provocar un daño. Una no conformidad en este punto puede llevar a tener que acabar de pulir la superficie externa del producto o a tener que rediseñar la distribución del peso del equipo.

9. Resistencia mecánica.

En las pruebas de resistencia mecánica, verificamos que el producto es capaz de soportar un manejo el rudo esperado durante su uso normal, sin que quede dañado comprometiendo el cumplimiento con los requisitos de la norma (por ejemplo la protección contra el choque eléctrico). Para ello se somete el producto a impactos con una fuerza determinada por la norma, a caídas desde la superficie normal de trabajo hasta el suelo, si por ejemplo el equipo está destinado a utilizarse sobre una mesa, etc. Una no conformidad en alguna de estas pruebas, puede llevar al rediseño de las protecciones mecánicas externas del producto.

10. Construcción.

En este capítulo se indican los requisitos constructivos que debe satisfacer el producto ensayado. Por ejemplo, se indica que los cables no deben dañarse en el interior del equipo debido a aristas cortantes, que las partes metálicas de los equipos deben conectarse a tierra si pueden quedar en tensión por un defecto de aislamiento. Una no conformidad para este apartado puede significar un rediseño de algunas de las partes del equipo, pero no suele implicar grandes cambios.

11. Componentes:

Las normas de seguridad eléctrica exigen que los componentes incluidos en el ESE , cumplan individualmente con su correspondiente norma de seguridad eléctrica, pudiendo el fabricante del ESE aportar prueba de ello (se entiende como prueba un certificado emitido por un laboratorio acreditado conforme a la norma correspondiente).

En el caso de descubrir un componente que no cumpla con su norma, o del cual no exista la documentación necesaria para demostrarlo, puede ser necesario sustituirlo o tener que ser ensayado aparte con el correspondiente sobrecoste.

En el caso de tener que ser sustituido por otro modelo, dependiendo del componente, puede ser necesario repetir la práctica totalidad de la norma (imaginemos que la fuente de alimentación interna del ESE no cumple con su correspondiente norma y la tenemos que cambiar por otra de otro fabricante, nos veremos obligados a repetir todas las partes de la norma en las cuales el equipo se ensaya en funcionamiento).

Por lo tanto es crítico evaluar cuanto antes los componentes incluidos en el ESE y verificar que todos están conforme a su correspondiente norma, para evitar repeticiones innecesarias o ensayos adicionales.

12. Medida de distancias en el aire y líneas de fuga:

En este capítulo evaluamos la separación entre partes peligrosas eléctricamente hablando y partes accesibles, para las cuales las normativas de seguridad eléctrica dan unos valores mínimos (en la figura 73 se puede observar un ejemplo). Si descubrimos que, por ejemplo, la distancia entre las pistas de primario y de secundario en una placa de circuito impreso no cumplen con los valores mínimos exigidos por la norma, el fabricante tendrá que modificar dicha placa para aumentar esa distancia, y puede que tenga que aumentar su tamaño, reposicionar componentes, etc. Todo esto hará que tengamos que repetir todas las partes de la norma en las cuales el equipo se ensaya en funcionamiento.

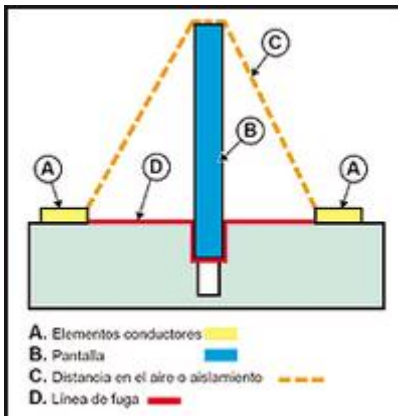


Figura 73. Ejemplo de medidas de líneas de fuga y distancias en el aire. (Fuente: Elaboración propia)

13. Resistencia al calor y al fuego.

Por último, en los ensayos de resistencia al calor y al fuego, evaluamos el comportamiento del equipo y de sus materiales de construcción en cuanto a propagación del fuego. Si se produce una llama en el interior del equipo por culpa del mal funcionamiento de uno de los componentes internos del equipo, o un componente se calienta en exceso también por un mal funcionamiento, los materiales plásticos que forman el equipo deben ser lo suficientemente resistentes al calor y a la propagación de la llama, para contener ese fuego en el interior del equipo, y para no deformarse de manera que partes internas a tensión peligrosa queden expuestas.

Para comprobar que esto se cumple, se realiza la prueba de hilo incandescente, en la cual se somete a los componentes plásticos a temperaturas de entre 650ºC y 850ºC y se comprueba que no se produzcan llamas o que si se producen se extingan dentro de los tiempos que indica la norma.

Una no conformidad en estos ensayos, puede llevar a tener que realizar un cambio en los materiales plásticos utilizados por otros de mayor calidad.

Como se ve por la descripción realizada de los parámetros ensayados, hay ensayos que pueden requerir un rediseño más profundo del equipo, mientras que hay otros que harán que se tengan que cambiar solo algunas partes del mismo. En función del grado de rediseño que se deba realizar se deberán repetir más o menos pruebas de las ya realizadas. De todas formas, se debe tener en mente que cualquier no conformidad va a requerir la modificación de una parte del equipo, y por lo tanto siempre se va a tener que repetir, como mínimo, el ensayo del cual resultó la no conformidad, para verificar que el cambio realizado soluciona el problema.

Considerando esto, se podrían agrupar las diferentes partes de la norma en iteraciones de una semana de la manera que veremos a continuación.

3.4.3.2 División del proyecto en iteraciones

Iteración 1.

Dentro de esta iteración se realizara lo siguiente:

- ✓ Estudio del ESE y de la oferta de los trabajos a realizar.
- ✓ Componentes. Revisión de todos los componentes y de sus certificados asociados.
- ✓ Construcción del ESE.
- ✓ Protección contra el choque eléctrico.
- ✓ Medidas de distancias en el aire y líneas de fuga.

Iteración 2.

Dentro de esta iteración se realizara lo siguiente:

- ✓ Ensayo de calentamiento normal.
- ✓ Medida de potencia y corriente.
- ✓ Ensayo de medida de corrientes de fuga y rigidez dieléctrica.
- ✓ Resistencia a la humedad.

Iteración 3.

Dentro de esta iteración se realizara lo siguiente:

- ✓ Estabilidad y riesgos físicos
- ✓ Resistencia mecánica.
- ✓ Resistencia al calor y al fuego.
- ✓ Funcionamiento anormal.
- ✓ Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…)

Esta seria, por lo tanto, la manera de dividir los diferentes "trabajos" dentro de las iteraciones de mayor "valor" a menor "valor", teniendo en cuenta que en este proyecto el valor consiste en repetir lo mínimo posible (nada idealmente) y en avisar al cliente lo antes posible de posibles cambios en el diseño de su equipo. Podemos ver esta división en la figura 74.

De esta manera, maximizamos la rentabilidad del proyecto y tenemos contento al cliente.

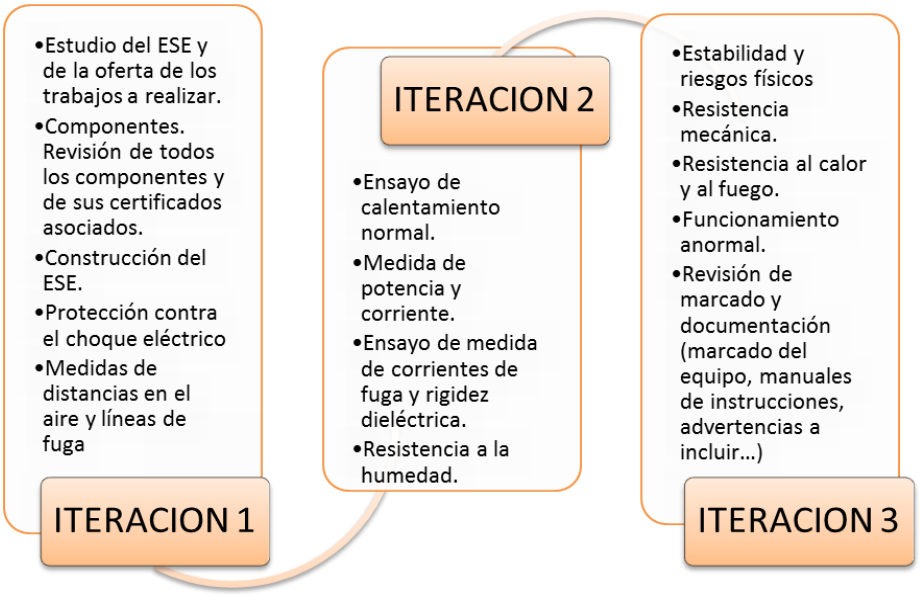


Figura 74. Iteraciones programadas. (Fuente: Elaboración propia)

3.4.4. Reuniones diarias y por iteración

Tal y como hemos visto al estudiar los métodos ágiles, las reuniones son un punto muy importante para poner en común la evolución de los proyectos y buscar soluciones de forma conjunta a los problemas que puedan aparecer.

En el esquema de trabajo actual, no existen unas reuniones prefijadas, produciéndose éstas de manera aleatoria. Cuando el test manager necesita conocer el avance de un determinado proyecto, habla con el técnico asociado al proyecto. Cuando un técnico tiene un problema, habla con el test manager para buscar una solución.

Los métodos ágiles nos dicen que el realizar reuniones dentro de la iteración y entre iteraciones ayuda a mejorar la dirección del proyecto y a que el líder del mismo pueda detectar y dar solución a los problemas que presenten los trabajadores.

Para implementar esta mejora, se fijarán reuniones diarias y de final de iteración, en las cuales participarán todos los técnicos de laboratorio y el test manager. También se fijará una reunión mensual, en la cual además del test manager y técnicos, participará el área comercial. Por otro lado, al inicio de cada proyecto, se fijará una reunión inicial entre el área comercial, test manager y técnico encargado de realizar los trabajos, de manera que puedan ponerse en común todos los detalles del proyecto.

3.4.4.1. Reunión diaria

Como nos indican los métodos ágiles, para que la reunión diaria sea efectiva, debe ser corta e ir enfocada a los puntos clave. Se trata de exponer los problemas surgidos durante la evolución del trabajo diario, de manera que el Test Manager pueda conocer el avance del proyecto y si hay algo que lo esté retrasando. También es el momento en el que el Test Manager debe guiar a los Técnicos en caso de que se esté produciendo un desfase entre el trabajo esperado y el realizado realmente.

Esta reunión no debe llevar más de 5 minutos por técnico.

Una vez conocidos los puntos importantes, el test Manager los analizará por su cuenta y buscará las soluciones necesarias. No es el objetivo de la reunión diaria el discutir la resolución de los mimos, ya que esto alargaría, seguramente, el tiempo mínimo.

3.4.4.2. Reunión de final de iteración

Al organizar las iteraciones por semanas, sería conveniente el realizar una reunión de final de iteración al finalizar cada semana.

En esta reunión, los técnicos expondrán los resultados del trabajo realizado en la iteración al test Manager, el cual decidirá el camino a seguir en la siguiente iteración, en función del resultado.

En caso de detectar no conformidades en el producto que puedan llevar a repeticiones de ensayo, decidirá que partes de la norma pueden seguir siendo ensayadas sin verse afectadas por el rediseño necesario para solventar la no conformidad, o en el caso de que se detecte algo muy grave, puede decidir detener los ensayos y esperar a que el cliente lo solucione.

3.4.4.3. Reunión mensual

En la reunión mensual, como se ha comentado, participarán todos los implicados en el proyecto (área comercial, test manager y técnicos). En esta reunión se expondrán los puntos más relevantes del proyecto y una idea general de cómo se desarrolla su avance, así como cualquier problema que haya podido surgir con el cliente (descontento por descubrir que su producto no está tan bien diseñado como creía, no aportación de la documentación necesaria, etc…). De esta forma, desde el área comercial podrán tomar acciones si consideran oportuno mejorar la satisfacción del cliente con el servicio ofrecido.

3.4.4.3. Reunión inicial de proyecto

En esta esta reunión participarán los implicados en el proyecto (comercial autor de la oferta, test manager y técnico asignado). En ella, se pondrá en conocimiento de todos, la información sobre el proyecto, cualquier incidencia surgida durante la fase comercial, plazos de entrega de resultados y particularidades detectadas en el producto. No se debe esconder ningún tipo de información, de manera que todos tengan conocimiento sobre todo.

3.4.5. Implicación de los Stakeholders

Uno de los puntos más importantes que resaltan los métodos ágiles, es la implicación de los Stakeholders. Es importante implicar a todos los interesados en el proyecto de manera que todos se sientan parte del mismo. De esta manera conseguiremos que todos den lo mejor de si mismos para llevar a cabo los objetivos del proyecto.

En primer lugar, identificaremos a los Stakeholders presentes en un proyecto de seguridad eléctrica.

3.4.5.1. Stakeholders

Como vimos anteriormente, los métodos ágiles definen como Stakeholder a cualquier persona o grupo que puede resultar impactado o impactar en el proyecto.

Dentro de un proyecto de seguridad eléctrica, podríamos diferenciar entre Stakeholders dentro del Laboratorio y Stakeholders externos al Laboratorio.

3.4.5.1.1. Stakeholders externos al Laboratorio.

Dentro de este grupo, tendríamos a todos los interesados en el proyecto que no están dentro de la organización del Laboratorio. En este tipo de proyectos, debido a su alto grado de confidencialidad, únicamente nos podremos comunicar con el peticionario de los servicios ofertados, en cuanto a resultados de ensayo.

Otro stakeholder externo al laboratorio son los proveedores de equipos para realizar los ensayos, y las entidades de calibración de estos equipos.

- Peticionario de los servicios ofertados.

Es decir, el cliente para el cual vamos a realizar los trabajos contenidos en la oferta. El peticionario no tiene por qué ser una única persona. Por lo general nos encontraremos con la persona que realiza la petición, que suele pertenecer a la parte directiva o comercial de la empresa, y una vez ya comenzados los ensayos, el contacto se suele llevar a cabo con alguien de la parte técnica de la empresa, que conoce el diseño del equipo a ensayar y puede resolver cualquier problema durante los ensayos.

- Proveedores de equipos de ensayo y calibración de los mismos.

El test manager deberá gestionar las nuevas compras de material, así como mantener el plan de calibración de los equipos utilizados para los diferentes ensayos realizados.

3.4.5.1.2. Stakeholders internos al Laboratorio.

Dentro del Laboratorio se amplía el número de actores que intervienen en el proyecto en mayor o menor medida:

- Comerciales.

Dentro de la organización del Laboratorio hay toda una parte comercial que se encarga de gestionar la petición de oferta del cliente. Estas personas se encargan del contacto inicial con el cliente y siguen el proyecto una vez iniciado.

- Director de operaciones.

Como vimos, a grandes rasgos, es el encargado de que todo funcione bien, de que se disponga de todo lo necesario para llevar adelante el proyecto y de mejorar los procesos.

- Product Manager.

Es el encargado del desarrollo de la línea de negocio, como vimos anteriormente.

- Test Manager:

Es el responsable técnico de los trabajos realizados y se encarga de controlar la facturación y producción de los proyectos.

- Técnicos de Laboratorio.

Son quienes realizan el trabajo técnico y quienes poseen el conocimiento de cómo realizar las tareas.

En la figura 75, se puede ver un resumen de los Stakeholders definidos.

Figura 75. Resumen de los Stakeholders del proyecto. *(Fuente: Elaboración propia)*

Éstos serían, por lo tanto, lo stakeholders asociados al proyecto de seguridad eléctrica. Veremos, ahora, como implicar a estos Stakeholders en el proyecto.

3.4.5.2. Técnicas para lograr la implicación de los Stakeholders

Como vimos en el estudio de los métodos ágiles, existen una serie de técnicas que nos ayudan a implicar a los Stakeholders en el proyecto.

Es importante esta implicación, ya que si todos los implicados en el proyecto entienden y comparten la dirección a seguir, lograremos acometer los objetivos del proyecto de una manera más eficiente.

Veremos a continuación qué técnicas de las estudiadas nos ayudarán a lograr esa implicación en un proyecto de seguridad eléctrica.

Diagramas de estructura.

Con los diagramas de estructura (ver figura 76), presentaremos el esquema de las iteraciones a todos los Stakeholders (con excepción, claro está de los proveedores de equipo). De esta manera, todos podrán estar al tanto del esquema general del proyecto, de cómo debería estar avanzando, y de qué tareas se esperan para cada semana.

De esta manera, los diferentes stakeholders podrán proporcionar feedback para los diferentes pasos de la iteración de manera intuitiva.

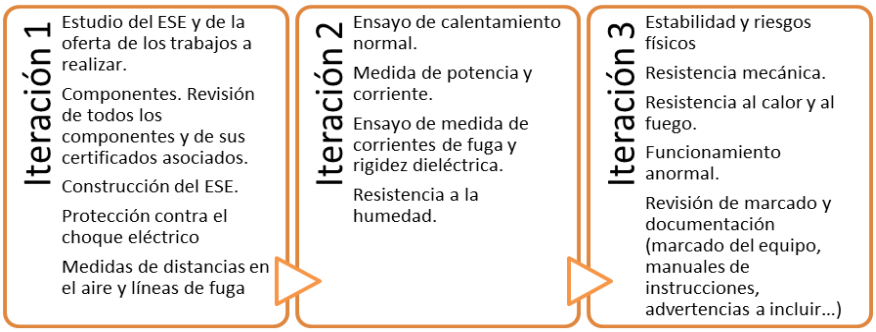


Figura 76. Diagrama de estructura del proyecto. (Fuente: Elaboración propia)

Comunicación con los Stakeholders.

Como vimos en el estudio de los métodos ágiles, la comunicación es algo muy importante. Como medio de comunicación, los métodos ágiles siempre recomiendan el cara a cara, como forma más efectiva de comunicarse entre Stakeholders y permite un feedback más efectivo. Teniendo en mente esta premisa, veremos que medios son los más adecuados para la comunicación entre los diferentes stakeholders.

En este caso concreto, la comunicación con los Stakeholders externos al Laboratorio, puede ser conveniente realizarla de diferentes formas en función de lo que queramos.

A la hora de comunicar no conformidades, pedir información para la realización de los ensayos, puede ser conveniente el realizar las comunicaciones en forma de e-mail en lugar de cara a cara, ya que de esa manera queda constancia por escrito. No obstante, una vez enviada la comunicación por e-mail es recomendable realizar una comunicación vía telefónica o cara a cara, de manera que podamos aclarar lo expuesto por e-mail.

Estas comunicaciones, se realizan entre los Técnicos de Laboratorio y el Peticionario o entre el Test Manager y el Peticionario, y pueden fluir en las dos direcciones, consiguiendo el feedback deseado.

En cuanto a las comunicaciones internas, se pueden dar entre el Test manager y los Técnicos de Laboratorio, entre el Test Manager y el Product Manager, entre los Técnicos de Laboratorio y el Director de Operaciones y entre el Test Manager y el Director de Operaciones. Es recomendable realizar todas estas comunicaciones cara a cara, de manera que podamos expresar lo que queremos comunicar con claridad. Solo en el caso de que sea importante dejar algo por escrito, realizaremos estas comunicaciones vía e-mail.

En las figuras 77 y 78, vemos un resumen de estas comunicaciones:

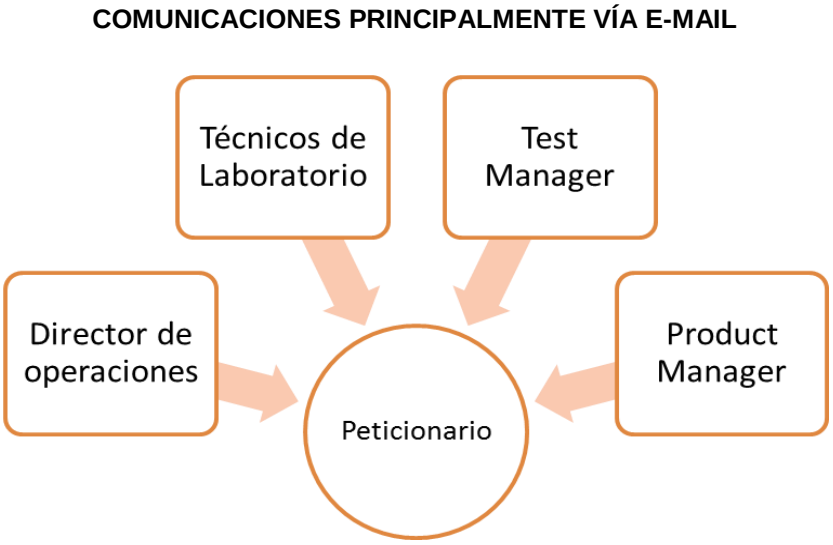


Figura 77. Comunicaciones vía e-mail con el Peticionario. (Fuente: Elaboración propia)

Figura 78. Comunicaciones cara a cara entre todos los Stakeholders. (Fuente: Elaboración propia)

3.4.6. Técnicas para controlar el avance del proyecto

Una de las funciones del Test Manager es controlar el avance del proyecto, de manera que se pueda saber en cada momento si el avance del mismo es el correcto o este avance se ha visto frenado respecto a las previsiones iniciales.

Para controlar este avance, es posible utilizar técnicas descritas por los métodos ágiles. Veremos a continuación cuáles son estas técnicas y cómo llevarlas a cabo.

En primer lugar, asignaremos a cada tarea del proyecto un valor en puntos.

Daremos 1 punto a la unidad más pequeña, que en este caso se corresponde con la tarea de "Revisión de marcado y documentación", que tiene una duración de 2 horas en un proyecto de seguridad eléctrica típico. Con esta consideración, vemos en el siguiente cuadro como queda el reparto de puntos por tarea:

Tareas	Puntos
Estudio del ESE y de la oferta de los trabajos a realizar.	1
Componentes. Revisión de todos los componentes y de sus certificados asociados.	4
Construcción del ESE.	4
Protección contra el choque eléctrico	1
Medidas de distancias en el aire y líneas de fuga	4
Ensayo de calentamiento normal.	8
Medida de potencia y corriente.	2
Ensayo de medida de corrientes de fuga y rigidez dieléctrica.	2
Resistencia a la humedad.	10
Estabilidad y riesgos físicos	2
Resistencia mecánica.	2
Resistencia al calor y al fuego.	4

Tareas	Puntos
Funcionamiento anormal.	4
Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…)	2
Total	50

Tabla 21: Reparto de puntos por tarea.

De esta manera, vemos que un proyecto típico de seguridad eléctrica tiene una duración típica de 50 puntos, que equivalen a 100 horas.

Ahora veremos cómo se reparten esos puntos en las iteraciones que vimos anteriormente.

Iteración 1.

Tareas	Puntos
Estudio del ESE y de la oferta de los trabajos a realizar.	1
Componentes. Revisión de todos los componentes y de sus certificados asociados.	4
Construcción del ESE.	4
Protección contra el choque eléctrico	1
Medidas de distancias en el aire y líneas de fuga	4
Total	14

Tabla 22: Iteración 1

En la primera iteración tendríamos un total de 14 puntos.

Iteración 2.

Tareas	Puntos
Ensayo de calentamiento normal.	8
Medida de potencia y corriente.	2
Ensayo de medida de corrientes de fuga y rigidez dieléctrica.	2
Resistencia a la humedad.	10
Total	22

Tabla 23: Iteración 2

En la segunda iteración tendríamos un total de 22 puntos.

Iteración 3.

Tareas	Puntos
Estabilidad y riesgos físicos	2
Resistencia mecánica.	2
Resistencia al calor y al fuego.	4
Funcionamiento anormal.	4
Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…)	2
Total	14

Tabla 24: Iteración 3

En la tercera iteración tendríamos un total de 14 puntos.

Los proyectos de seguridad eléctrica son llevados a cabo por un único técnico de laboratorio, debido a la complejidad de los mismos. Como se ha comentado, los técnicos son expertos en la aplicación de la normativa, pero además están especializados en normas concretas, en las cuales tienen un nivel de experiencia aún más alto que en el resto. Por lo tanto, el máximo de puntos que pueden llevarse a cabo por semana son 20, que equivaldrían a una jornada de 40 horas semanales.

No obstante, de esas 40 horas, debemos reservar una parte para las reuniones diarias y semanales (por iteración), comunicaciones con el resto de Stakeholders del proyecto, y tiempos no productivos (como ir al lavabo, por ejemplo), que valoraríamos en un 30% del tiempo de jornada semanal. Por lo tanto, quedarían disponibles 28 horas de trabajo, que equivaldrían a 14 puntos por semana/iteración.

Por lo tanto, como habíamos visto que un proyecto típico de seguridad eléctrica constaba de 50 puntos, tenemos que serán necesarias 50/14 = 3,5 iteraciones aproximadamente.

Vemos que el reparto de tareas inicial que realizamos basándonos en el valor de las mismas, debe ser reajustado, ya que en la iteración 2 tenemos un total de 22 puntos.

Un reajuste que no afectaría a la clasificación por valor que se había realizado anteriormente, sería el siguiente:

Iteración 1.

Tareas	Puntos
Estudio del ESE y de la oferta de los trabajos a realizar.	1
Componentes. Revisión de todos los componentes y de sus certificados asociados.	4
Construcción del ESE.	4
Protección contra el choque eléctrico	1
Medidas de distancias en el aire y líneas de fuga	4
Total	14

Tabla 25: Iteración 1 reajustada

En la primera iteración tendríamos un total de 14 puntos.

Iteración 2.

Tareas	Puntos
Medida de potencia y corriente.	2
Ensayo de medida de corrientes de fuga y rigidez dieléctrica.	2
Resistencia a la humedad.	10
Total	14

Tabla 26: Iteración 2 reajustada

En la segunda iteración tendríamos un total de 14 puntos.

Iteración 3.

Tareas	Puntos
Ensayo de calentamiento normal.	8
Estabilidad y riesgos físicos	2
Resistencia al calor y al fuego.	4
Total 14	14

Tabla 27: Iteración 3 reajustada

En la tercera iteración tendríamos un total de 14 puntos.

Iteración 4.

Tareas	Puntos
Resistencia mecánica.	2
Funcionamiento anormal.	4
Revisión de marcado y documentación (marcado del equipo, manuales de instrucciones, advertencias a incluir…)	2
Total	8

Tabla 28: Iteración 4 reajustada

En la cuarta iteración tendríamos un total de 8 puntos.

Con la división por puntos realizada, el test manager puede controlar el avance del proyecto, a partir de los puntos completados por el Técnico de Laboratorio.

Éste a su vez, desvincula el total de puntos por iteración del número de horas. Esto puede ser positivo, en el sentido de que si el técnico de laboratorio es capaz de realizar los 14 puntos en menos de una semana, no lo relacionará con que ya ha acabado el trabajo necesario para esa semana, y continuará con la siguiente tarea.

3.4.7. Resolución de conflictos

Otra de las tareas del Test Manager es resolver los conflictos que puedan aparecer entre los técnicos de laboratorio.

Los conflictos suelen venir ocasionados en su mayor parte por la compartición de recursos. Como recursos entendemos espacios y equipos para realizar los ensayos. El equipo de técnicos de laboratorio en el departamento de seguridad eléctrica está formado por 8 personas. Estas 8 personas, comparten equipos y espacios para la realización de ensayos dentro de los proyectos que cada uno de ellos lleva a cabo.

Los métodos ágiles ofrecen una clasificación en función del nivel del conflicto. En el caso que estamos estudiando, los conflictos no pasarán de un Nivel 1 (problema a resolver).

Por lo tanto al test manager le basta con mediar entre los técnicos y decidir qué proyecto tiene prioridad, en el caso que dos técnicos necesiten utilizar el mismo recurso a la vez.

Para ello debe intentar construir un escenario de consenso en el que consiga que los técnicos acepten la decisión tomada, y ésta acabe siendo algo consensuado entre ellos.

3.4.8. Ubicación de los equipos

Como se vio durante el estudio de los métodos ágiles, éstos recomiendan que los equipos se encuentren coubicados.

Los beneficios de los equipos coubicados son la creación de espacios de equipo colaborativo, de manera que cuando uno de los miembros del equipo resuelve un problema que ha tenido lugar durante el proyecto, el resto de miembros aprenden también como solucionarlo por el simple hecho de estar ubicados unos cerca de otros.

Esto es muy importante en un entorno de seguridad eléctrica, ya que los proyectos son muy similares entre sí, aunque los equipos ensayados puedan ser diferentes, por lo que los problemas que surgen durante los ensayos, también son muy similares en todos los proyectos. De esta manera creamos el llamado conocimiento tácito, mediante el cual el grupo crea un conocimiento no escrito de cómo resolver la mayoría de problemas que aparecen durante el transcurso del proyecto. Para esto, nos ayudan tanto la coubicación de los equipos como las reuniones diarias y por iteración que veíamos antes.

Como vimos podemos ver dos bloques diferenciados durante los ensayos de Seguridad Eléctrica. Existe una parte que es puramente ensayo, y otra en la que se realizan inspecciones visuales y documentales.

Para la realización del bloque de ensayos, necesitamos un laboratorio de Seguridad Eléctrica acondicionado en el cual realizar los ensayos.

Para la realización del bloque de inspecciones visuales y documentales, necesitamos una zona en la que poder realizar estas tareas de manera cómoda (escritorio, mesas con sillas…).

Aunque estos dos bloques estén diferenciados es importante que los dos zonas en las que se llevan a cabo estén la una cerca de la otra, de manera que mantengamos la coubicación y no se pierda el contacto entre los dos bloques.

Por ello lo ideal es una zona de oficinas desde la cual se tenga una visión clara de la zona de ensayo.

3.5. GESTIÓN DEL LIDERAZGO Y DEL RENDIMIENTO DE LOS EQUIPOS

Una vez se ha visto como implantar la metodología ágil en muchos de los aspectos del Laboratorio de Seguridad Eléctrica, veremos cómo utilizar las técnicas de los métodos ágiles para liderar y aumentar el rendimiento del equipo de técnicos de laboratorio. Los encargados de estas tareas son tanto el director de operaciones como el test manager.

3.5.1. Liderazgo de servicio

Como vimos anteriormente los líderes deben adoptar un rol de liderazgo de servicio en cual se reconoce que son los miembros del equipo, no el líder, quienes realizan el trabajo y logran los valores del negocio. Por lo tanto se centra al líder en proveer al equipo de lo que necesite, eliminando los impedimentos para progresar, y realizando tareas de soporte para maximizar la productividad del equipo.

Como deberes principales tendríamos que:

1. Proteger al equipo de interrupciones.

En el caso de un proyecto de seguridad eléctrica, estas interrupciones pueden llegar desde distintos frentes, que el Test manager debe atajar:

- Peticiones de ofertas de la parte comercial. Al ser los Técnicos de Laboratorio expertos en su campo, pueden resultar de gran ayuda a la hora de confeccionar las ofertas de los proyectos, pero no debemos olvidar que su principal objetivo es avanzar en las iteraciones. Por lo tanto el Test Manager, que también es experto debe asumir esta función siempre que le sea posible.
- Llamadas externas realizando consultas técnicas. Es muy frecuente que posibles futuros clientes realicen llamadas preguntando por temas técnicos relativos a los ensayos. El test manager debe desviar todas estas llamadas fuera de los técnicos, ya que al no pertenecer a ningún proyecto en concreto, no aportan valor inmediato, pero si que pueden significar un futuro cliente, por lo que deben ser tratadas con la máxima atención.

2. Eliminar los impedimentos para avanzar.

- Problemas con los equipos. En el caso de que un equipo no funcione correctamente o no esté operativo, el Test Manager o el Director de Operaciones, deben hacer todo lo posible para que esto no afecte al avance del proyecto (gestionando su reparación rápidamente o consiguiendo un equipo de sustitución).
- Desorden general. Es importante mantener el orden dentro del Laboratorio, ya que de lo contrario comienzan a surgir ineficiencias que pueden afectar al ritmo de avance del proyecto. Por lo tanto, el Director de Operaciones, debe cuidar de que se mantenga el orden dentro del Laboratorio.
- Tareas de calidad. Por ejemplo, calibraciones de equipos, análisis de normativa, son tareas que no aportan un valor directo al proyecto, pero que deben ser realizadas. El test manager debe gestionar estas tareas de manera que afecten lo mínimo posible al avance del proyecto

3. Mantener la visión del proyecto en mente de todos los trabajadores.

- El test manager debe mantener un control sobre el avance del proyecto y no permitir que los técnicos se desvíen de las tareas proyectadas para la iteración.

4. Procurar que el equipo sienta que se valora su trabajo.

- Los líderes necesitan aprender que motiva a los miembros de sus equipos como individuos y encontrar maneras de recompensarlos por el buen trabajo. Como medida simple, un buen lugar para comenzar es un sincero “gracias” a alguien por su duro trabajo.

3.5.2. Herramientas y técnicas de liderazgo

Para conseguir todo lo anterior, los diferentes líderes deben apoyarse en los principios y técnicas vistos en el estudio de los métodos ágiles.

- Modelado del comportamiento deseado.

El líder debe modelar el comportamiento de los miembros de su equipo a través de su propio comportamiento, de manera que aporte lo indicado en la figura 79:

Figura 79. Valores del líder

Estos cuatro valores ayudarán a que los miembros del equipo colaboren con el líder.

- Permitir que otros actúen.

Para permitir que los miembros de nuestro equipo se sientan seguros tomando decisiones y acciones que hagan avanzar el proyecto de forma productiva, necesitamos fomentar un ambiente de colaboración. Esto implica el fomentar la confianza entre los miembros del equipo y fortaleciendo a los demás compartiendo el poder. También necesitamos crear un ambiente de trabajo seguro en el que las personas no estén preocupadas por preguntar lo que piensan. Las personas aprenden mucho más rápido cuando pueden elevar preguntas sin miedo a represalias o a hacer el ridículo.

Esto es muy importante en los proyectos de Seguridad Eléctrica en los cuales se está continuamente tomando decisiones sobre la interpretación de la normativa, la cual es muy difusa. Los técnicos necesitan sentir la confianza suficiente para interpretar la norma y no tener que estar continuamente realizando consultas sobre si el criterio seguido es o no el correcto.

3.5.3. Rendimiento del equipo

Una de las funciones más importantes tanto del Director de Operaciones como del Test Manager es la de mejorar el rendimiento del equipo.

Para lograr esta mejora en el rendimiento, y por lo tanto aumentar la productividad a la hora de llevar a cabo los proyectos, los métodos ágiles nos dan una serie de pautas que podemos seguir.

En primer lugar podemos utilizarlos para crear equipos capaces los cuales, como vimos, están autoorganizados y autodirigidos.

Los equipos autoorganizados son muy importantes ya que implican que los miembros del mismo tengan autonomía a la hora de decidir cómo realizar las pruebas y cómo hacer avanzar el proyecto. El marcar demasiado estrictamente el camino a seguir puede llevar a la desmotivación del equipo.

En el caso que estamos estudiando, los equipos están formados por expertos capaces de utilizar su conocimiento para hacer mejor su trabajo. El dejarles libertad para organizarse por ellos mismos resultará motivador.

En cuanto a los equipos autodirigidos, son equipos que trabajan de manera colectiva para crear las normas del equipo y toman sus propias decisiones internas. Esto significa que no solo piensan en la mejor manera de cumplir el trabajo que tienen comprometido para la iteración, sino que además resuelven muchos de los problemas del día a día que surgen a lo largo del camino. Los gestores de proyecto o líderes pueden apoyar y reforzar este comportamiento respetando las estimaciones del equipo y sus decisiones y permitiéndoles el cometer errores y solucionarlos.

Esto no significa que el gestor de proyecto abdique la responsabilidad al equipo. En lugar de eso, lo que quiere decir es que se le da al equipo libertad dentro de los confines de la iteración.

Una parte muy importante a la hora de mejorar el rendimiento de un equipo es la motivación del mismo.

Tal y como nos dicen los métodos ágiles la motivación va más allá de pagar un salario a una persona. El pagar un salario solo motiva a que las personas vayan a su trabajo, pero si se quiere obtener el máximo rendimiento se deben buscar otros factores motivadores.

Es importante que tanto el director de operaciones como el test manager busquen la motivación de los miembros del equipo de manera que todos estén alineados hacia el objetivo del proyecto, figura 80.

Figura 80. Alineados hacia el objetivo

Por lo tanto deberán realizar reuniones con cada uno de los técnicos e intentar descubrir que motiva a cada uno de ellos. Esta motivación no tiene por qué ser la misma para todos.

Es posible que a algunos miembros les motive el destinar parte de su tiempo a la realización de ofertas, a otros quizás les motive el realizar tareas de calidad, tales como el análisis de normativas nuevas.

3.5.3.1. Prácticas para aumentar la efectividad del equipo

Hay una serie de prácticas que los líderes pueden llevar a cabo para aumentar la efectividad del equipo, y que pueden ser utilizadas en un proyecto de seguridad eléctrica:

- Reuniones diarias

En estas reuniones, limitadas a 15 minutos como vimos, el test manager y los técnicos pueden discutir los diferentes problemas surgidos y ayudan al test manager a realizar el seguimiento del proyecto.

- Formación y asesoramiento.

Formando y asesorando a los equipos ágiles se les ayuda a mantenerse en la pista del proyecto, superar los problemas, y mejorar continuamente sus habilidades. La formación se realiza simultáneamente en dos niveles, a nivel individual y a nivel del equipo completo.

Aquí incluiríamos el tiempo necesario para mejorar la formación de los técnicos respecto a las normativas, destinando parte de su tiempo a la autoformación de la normativa aplicada. De esta manera es posible mejorar la velocidad a la que se realizan las pruebas.

- Técnicas de Brainstorming.

En el caso de un proyecto de seguridad eléctrica que estamos tratando, podemos utilizar las técnicas de Brainstormig para resolver problemas de operativa que afecten a todos los técnicos. Por ejemplo podemos intentar mejorar la organización del laboratorio de manera que se reduzcan tiempos de búsqueda de herramientas. Para todo esto las ideas de los técnicos que viven el día a día en el laboratorio pueden ser cruciales.

Veamos un ejemplo práctico de una práctica de Brainstorming que se podría llevar a cabo.

Como se vio la base de una sesión de Brainstorming, es que ninguna idea es mala, cuantas más ideas aparezcan más fácilmente podremos obtener buenas ideas que realmente nos ayuden a mejorar o solucionar lo que estamos tratando.

Uno de los problemas que podemos encontrar es el de mantener el orden y limpieza en el laboratorio. Es importante mantener todo ordenado y accesible para evitar tiempo perdidos buscando equipos que no aparecen, herramientas, etc. Para mejorar este orden y limpieza, el test manager puede organizar una sesión de brainstorming entre los técnicos de manera que pueda obtener ideas de las personas que llevan el día a día de los ensayos y mejor conocen los problemas que pueden plantearse por culpa de no mantener el orde.

Para organizar esa sesión, se seguirían las reglas que se vieron en la parte teórica:

1. Elegir un coordinador, que apuntará las ideas y será el facilitador. En este caso, puede ser uno de los técnicos o el mismo test manager.

2. Definir el problema o asunto como un desafío creativo. El test manager puede proponer el tema de orden y limpieza como un reto para reorganizar el laboratorio, facilitar el día a día y aumentar la productividad.

3. Establecer un tiempo límite por sesión. El tiempo recomendado es de unos 25 minutos.

4. Una vez comienza el Brainstorming, se escriben cada una de las ideas en el menor número de palabras posible. Los participantes dan propuestas y el test manager o técnico será el que tome note en una pizarra en blanco o alguna superficie donde todos las vean. Se deben escribir todas las ideas, a pesar de lo absurdas que parezcan. Se debe fomentar el buen ambiente y no las críticas. Se establecerá el turno a seguir señalando el participante que comenzará. Cuando se llega a un punto del desarrollo en que el volumen de ideas aportado decrece apreciablemente, se hará una ordenación o una lectura de las ideas aportadas, produciéndose generalmente una segunda fase creativa.

5.Se revisan todas las ideas para verificar su comprensión, solicitando la información que fuera necesaria, y llegar a un consenso sobre los problemas redundantes, no importantes o no negociables. La tormenta de ideas se dará por finalizada cuando ningún participante tenga ideas que aportar.

6.Se revisa la lista de ideas, agrupando las que puedan ir juntas, y eliminando las duplicadas.

7.Se determinarán cinco criterios para juzgar que ideas solucionan mejor el problema o desafío. Por ejemplo en este caso:

- Que la idea no implique un sobrecoste para el departamento.

- Que la solución no implique el perder tiempo productivo en exceso.

- Que implique al menor número de personas en la aplicación de la misma.

- Que ayude a mejorar al ambiente de trabajo.

- Que no implique el comprar mobiliario por encima de una cierta cantidad en euros.

8.Se definen cuantas ideas votará cada participante. Debería ser como mínimo 1/3 del total.

9.Cada integrante escribirá las ideas seleccionadas.

10.Se cuentan los votos que ha recibido cada idea.

11.Se eliminan las ideas con un menor número de votos.

12.Se repiten los pasos anteriores con la lista reducida hasta que lleguemos a una lista donde todas las ideas tengan el mismo peso.

13.Una vez obtenida esta lista se discute grupalmente hasta llegar a una decisión final.

Con esta sesión, el test manager obtendrá un gran feedback de su equipo, que su vez verá que sus ideas son importantes y se llevan a cabo, con lo que se sentirá más motivado en general y en particular a aplicar esas ideas.

Como se vio en la parte teórica, siempre es más fácil aceptar una idea que has pensado tu mismo, o que sobre la que has llegado a un acuerdo, que una idea impuesta por un tercero.

4. CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN

4.1. CONCLUSIONES

En la presente Tesis Final de Máster se ha realizado una investigación de los métodos ágiles buscando como objetivos, crear una guía de estudio que ayude a adquirir los conocimientos necesarios para superar el examen PMI-ACP (la Certificación ACP del Project Management Institute (PMI)) e implantar este tipo de metodología en proyectos no relacionados con el software.

A lo largo de los diferentes puntos desarrollados se ha hecho ver la utilidad de los métodos ágiles como herramienta para llevar a cabo proyectos en los que son los trabajadores de los mismos quienes poseen los conocimientos para llevarlos a cabo.

Hemos visto los diferentes métodos que podemos aplicar, la importancia de implicar a todas las personas que tengan algo que ver con el proyecto y cómo gestionar esta implicación.

También se ha remarcado la importancia que tienen los trabajadores con el conocimiento en este tipo de proyectos. Son una de las piezas claves para lograr el éxito y por lo tanto debemos basar gran parte de nuestra gestión en ellos. Es el llamado liderazgo de servicio en el cual se considera que el gestor del proyecto debe destinar parte de sus esfuerzos en lograr que el equipo pueda realizar su trabajo sin distracciones y proporcionarles todo lo necesario.

Se ha destacado especialmente como uno de los puntos clave el proporcionar valor al cliente. Nuestro objetivo principal es diseñar algo que sea útil para nuestro cliente y que satisfaga todas sus necesidades y expectativas. Para ello, como hemos visto, es necesario mantener un contacto constante con el cliente, de manera que valide nuestro trabajo de manera frecuente y no nos desviemos de lo que realmente se espera de nuestro trabajo.

Por otro lado, durante el estudio práctico realizado, se ha visto cómo podemos aplicar los conceptos investigados a una organización de trabajo existente y que no los utiliza.

Se han aplicado esos conceptos a una empresa no relacionada con la producción de software, con lo que se ha demostrado que, aunque los métodos ágiles fueron inicialmente creados para gestionar proyectos de software, sus conceptos y técnicas son igualmente aplicables a otro tipo de aplicaciones.

En este caso se ha tratado de una empresa de servicios, que ofrece sus capacidades para el ensayo de material eléctrico de acuerdo a diferentes normativas.

El estudio se ha centrado, sobretodo en la parte de gestión, para comprobar como la aplicación de los métodos ágiles podían ayudar a mejorar distintos sectores, como la organización, motivación, rendimiento, etc.

Como conclusión del estudio práctico de la implementación de los métodos ágiles, podemos decir que los efectos de su implantación son beneficiosos, ya que lograrán que:

1. La implicación de todas las personas relacionadas con el proyecto mejore, ya que se valora su opinión y se crean las herramientas para que esa opinión influya en el proyecto.
2. La gestión de los recursos necesarios para llevar a cabo el proyecto (personas, equipos, etc…) sea más eficiente.
3. El cliente final del servicio perciba una mayor transparencia en lo que está siendo realizado y en el avance del proyecto.
4. La comunicación entre las distintas personas dentro del proyecto mejore, con lo que detectaremos antes los errores y evitaremos repeticiones innecesarias.
5. Cualquier conflicto que surja entre las diferentes personas implicadas en el proyecto, pueda resolverse de manera rápida, gracias al uso de las diferentes técnicas que nos ofrecen los métodos ágiles.

Por lo tanto, podemos concluir que los métodos ágiles son una excelente herramienta para gestionar proyectos en los cuales el conocimiento de las personas que los realizan es muy importante, proyectos en los cuales el producto final a crear es cambiante y puede variar según varíen las necesidades del cliente durante el desarrollo del mismo. Estos métodos nos ayudan a gestionar todos esos inconvenientes que seguramente no aparecerán en proyectos más convencionales. Eso sí, para que un método ágil funcione plenamente, debemos adaptarlo a nuestro proyecto en concreto, por lo tanto, necesitamos flexibilidad a la hora de aplicarlos.

4.2. LÍNEAS FUTURAS DE INVESTIGACIÓN

Los métodos ágiles son herramientas cambiantes que se adaptan continuamente a los diferentes proyectos, por lo tanto surgen continuamente nuevos métodos o variaciones de los existentes.

a) Una línea futura de investigación podría ser el profundizar en los métodos ágiles aquí expuestos, buscando sus puntos débiles y proponiendo líneas de mejora para estos, adaptadas a todo tipo de proyectos, no solo de software.

b) De la misma manera se podría investigar sobre los nuevos métodos que aparezcan, desde una posición crítica, buscando también sus puntos débiles y tratando de buscar mejoras a estos.

c) Otra posibilidad es ampliar el caso práctico realizado, tratando de extender el uso de la metodología ágil no solo a la parte productiva considerada (laboratorio de seguridad eléctrica), sino a los distintos laboratorios dentro de la división, como son el de compatibilidad electromagnética, el de vibraciones, el de ensayos climáticos o acústica.

d) Otra posible futura de investigación, consistiría en la aplicación práctica de cálculos de valoración de inversiones, como son el VAN y la TIR, en proyectos de inversión de maquinaria, equipos de ensayos, acreditaciones para nuevas normativas, entre otros, que permitan expandir las líneas de negocio existentes.

5. BIBLIOGRAFÍA

5.1 Libros consultados

- Cockburn, Alistair. *Agile Software Development*. Highsmith Series

- Griffiths, Mike, PMI-ACP Exam Prep

- PMI-ACP Handbook (http://www.pmi.org/en/Certification/~/media/PDF/Certifications/PMI-ACP_Handbook.ashx)

- Agile Software Development: TheCooperativeGame – 2nd Edition (AlistairCockburn)

- CoachingAgileTeams, Lyssa Adkins

- Lean-Agile Software Development: Achieving EnterpriseAgility, Alan Shalloway,Guy Beaver, James R. Trott

- Chin, Gary (2004). Agile Project Management: How to Succeed in the Face of Changing Project Requirements. AMACOM.

- Martinez, Gustavo (2011). Coding, quality check and documentation (300%): Get them from the same development team!. VPD.

5.2 Artículos consultados

- Métodos de desarrollo de software: el desafío pendiente de la estandarización; Ricardo A. Gacitúa Bustos

- Notas de Scrum; Autor: Villarreal, Gonzalo

- Metodologías tradicionales vs. Metodologías ágiles; Roberth G. Figueroa , Camilo J. Solís Armando A. Cabrera

- Metodologías Ágiles; Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos

- Una representación gráfica del testing ágil; Carlos M. Zapata, Ph. D. Adrian S. Arboleda, Ing. Carlos E. Castrillón.

5.3 Páginas web consultadas

http://www.pmi.org/

http://www.pminstitute.com/

http://www.javiergarzas.com/metodologias-agiles

http://agilemanifesto.org/iso/es/

http://agile-spain.org/

http://www.scrum.org/

http://www.extremeprogramming.org/

http://es.wikipedia.org/wiki/Laboratori_General_d'Assaigs_i_Investigacions

http://www.applus.com/es/aboutUs/ourHistory

6. ANEXO DE EXÁMENES PRÁCTICOS

6.1. Examen práctico capítulo 2.1: Estructura ágil

QUESTIONS

1. The Extreme Programming (XP) core values include:
   A. Solutions, communication, feedback
   B. Simplicity, control, feedback
   C. Simplicity, communication, feedback
   D. Solutions, control, feedback

2. The three pillars of Scrum are:
   A. Transparency, Communication, Adaptation
   B. Inspection, Adaptation, Evolution
   C. Adaptation, Inspection, Communication
   D. Transparency, Inspection, Adaptation

3. In the daily Scrum meeting, what three questions are asked?
   A. What has been achieved since the last meeting? What will be done before the next meeting? What obstacles are in the way?
   B. What has been started since the last meeting? What might get done before the next meeting? What obstacles are in the way?
   C. What has been started since the last meeting? What will be done before the next meeting? What obstacles are in the way?
   D. What has been achieved since the last meeting? What might be done before the next meeting? What obstacles are in the way?

4.What are the names of the backlogs maintained in Scrum?
   A.Product backlog, iteration backlog
   B. Product backlog, sprint backlog
   C. Project backlog, sprint backlog
   D. Project backlog, iteration backlog

5.In Scrum, the definition of “done” is created by everyone EXCEPT :
   A.Development team
   B. Product owner
   C. ScrumMaster
   D. Process owner

6.In Extreme Programming (XP), the core practices include:

A.Whole team, planning games, small releases

B.One team, process game, small releases

C.Whole team, planning games, quick releases

D.One team, process game, quick releases

7.Select the correct combination of lean principles:

A.Eliminate waste, empower the team, build quality in, deter decisions, amplify learning

B.Exterminate waste, empower the team, build quality in, deter decisions, optimize learning.

C.Eliminate waste, empower the team, build quality in, defer decisions, amplify learning

D.Eliminate waste, empower the team, build quality in, defer decisions, optimize learning

8.Which of the following are some of the planned opportunities for Inspection and Adaptation in the Scrum method?

A.Sprint retrospective, velocity review meeting, daily scrum meeting

B.Sprint planning meeting, sprint retrospective, sprint risk meeting

C.Sprint retrospective, daily scrum meeting, sprint review meeting

D.Sprint planning meeting, daily scrum meeting, retrospective planning meeting

9.When using Kanban development, the acronym WIP stands for:

A.Waste in progress

B.Work in progress

C.Waste in process

D.Work is progressing

10.Which of the following is an Agile Manifesto value?

A.Working solutions over complete documentation

B.Working software over comprehensive documentation

C.Working solutions over comprehensive documentation

D.Working software over complete documentation

11.In the Agile Manifesto, what is valued more than processes and tools?

A.Customer collaboration

B.Individuals and interactions

C.Working software

D.Responding to change

12.Which of the following Agile Manifesto values deals most closely with WIP?

A.Customer collaboration over contract negotiation

B.Individuals and interactions over processes and tools

C.Working software over comprehensive documentation

D.Responding to change over following a plan

13.When agile teams use the term “timeboxed” what do they mean?

A.Work shall take a minimum amount of time

B.Work can take no more than a maximum amount of time

C.Work must be done by a given time, plus or minus 20 percent

D.Work must happen at a set time

14.Which Agile Manifesto value is concerned with team empowerment?

A.Customer collaboration over contract negotiation

B.Individuals and interactions over processes and tools

C.Working software over comprehensive documentation

D.Responding to change over following a plan

15.The five core principles of Kanban development include:

A.Minimize the workflow, pull flow, make process policies explicit, improve collaboratively.

B.Visualize the workflow, manage flow, make process policies explicit, improve collaboratively.

C.Minimize the workflow, manage flow, make process policies explicit, improve collaboratively.

D.Visualize the workflow, pull flow, make process policies explicit, improve collaboration.

ANSWERS

1.Answer: C

Explanation: The three correct XP core values from the options presented are: simplicity, communication, and feedback. “Solutions” and “control” are not core values.

2.Answer: D

Explanation: The three pillars are Transparency, Inspection, and Adaptation. Evolution and communication are both admirable goals, but they are not Scrum pillars.

3.Answer: A

Explanation: The three questions asked in the daily Scrum meeting are: “What has been achieved since the last meeting?” “What will be done before the next meeting?” and “What obstacles are in the way?” Work started but not completed, or work that might be done before the next meeting, is not discussed to keep the meeting results-oriented.

4.Answer: B

Explanation: The backlogs maintained in Scrum are the product backlog and the sprint backlog. Scrum calls iterations “sprints,” so Scrum teams have a sprint backlog rather than an iteration backlog. Likewise, there is a product backlog but not a project backlog. Regardless of the term used, it is the practice that is really important. To be able to work effectively with people across methodologies, however, we do need to understand the differences in terminology.

5.Answer: D

Explanation: The whole team, including the development team, product owner, and ScrumMaster, is responsible for creating a shared definition of “done.” Since “process owner” is a made-up term, this is the correct choice for someone who would NOT be involved in defining done.

6.Answer: A

Explanation: The correct set of XP core practices include: whole team, planning games, and small releases. The choices of “one team,” “process game,” and “quick releases” are not XP core practices.

Answer: C

Explanation: The valid lean principles from this list are: eliminate waste, empower the team, build quality in, defer decisions, and amplify learning. "Exterminate waste," "Optimize learning," and "Deter decisions" are not lean principles. Make sure you read each answer choice carefully on questions like these—"defer" looks very much like "deter," especially when you are under pressure.

7.Answer: C

Explanation: Scrums planned opportunities for Inspection and Adaptation are the sprint retrospective, the daily scrum meeting, and the sprint review meeting. Velocity review meetings, sprint risk meetings, and retrospective planning meetings are not recognized Scrum events.

8.Answer: B

Explanation: WIP stands for "Work in progress" or "Work in Process," not "waste" or "progressing."

9.Answer: B

Explanation: The Agile Manifesto contains the value: "Working software over comprehensive documentation." The other choices express similar ideas, but they are not correctly stated.

10.Answer: B

Explanation: The Agile Manifesto value is "Individuals and interactions over processes and tools." Therefore, "individuals and interactions" is the correct choice.

11.Answer: C

Explanation: The value of "Working software over comprehensive documentation" is most closely related to WIP. This value speaks to the need to be focused on results and completed deliverables, rather than partially completed or in-progress work. If things like comprehensive documentation are holding up the ability to complete a deliverable—in other words, if a need for comprehensive documentation is keeping a lot of work in progress—it should be reduced as much as possible to support the ultimate goal of creating a useful, functional product.

12.Answer: B

Explanation: If work is timeboxed, it means it has a maximum duration assigned to it. This ensures the team spends an appropriate amount of time on the work without allowing waste.

13.Answer: B

Explanation: "Individuals and interactions over processes and tools" speaks to the concept that the benefits of empowerment outweigh the benefits of processes and tools on a project. The "customer collaboration over contract negotiation" choice is close, but that value is less about empowerment than about the need to respond to new information and changes instead of strictly adhering to old contracts and plans.

14.Answer: B

Explanation: The five core principles of Kanban development include: visualize the workflow, manage flow, make process policies explicit, and improve collaboratively. "Pull flow," "minimize the workflow" and "improve collaboration" all sound close to the real phrases, but they are not Kanban principles.

6.2. Examen práctico capítulo 2.2: Entregas basadas en el valor

QUESTIONS.

1.Which of the following is NOT an advantage of limiting work in progress (WIP)?

A.It reduces the potential need to rework a large collection of flawed, partially completed items.

B.It helps optimize throughput to make processes work more efficiently.

C.It brings bottlenecks in the production process to the surface so they can be identified and resolved.

D.It maximizes resource utilization to make processes work more efficiently.

2.Project X has a IRR of 12%, and project Y has a IRR of 10%. Which project represents the better rate of return?

A.It depends on the payback period

B.Project Y

C.Project X

D.Project X or Y, depending on NPV

3.Which of the following is NOT a recognized prioritization scheme?

A.Prioritization based on business value

B.Prioritization based on reducing risk

C.Prioritization based on iteration velocity

D.Prioritization based on business value and risk

4.What is a product roadmap?

A.A list of reports and screens

B.A view of release candidates

C.Instructions for deployment

D.A backlog prioritization scheme

5.Which of the following is true about agile contracts?

A.They only work when the specs are fully defined.

B.They only work for time and materials agreements.

C.They can accommodate changes.

D.They cannot easily accommodate changes.

6.We put risks in the backlog to:

A.Avoid having to keep separate risk lists

B.Keep the team focused on risks

C.Ensure that they get worked on in the early iterations

D.Prevent the team from forgetting about the risks

7.Which of the following combinations is a valid mix for story prioritization?

A.Customer importance and business urgency

B.Risk reduction and technical dependency

C.Customer value and risk reduction

D.Team preference and customer value

8.To cut costs, company X has decided to cancel one of its money-losing projects. Project A has a IRR of -4%, project B has a IRR of -5%, and project C has a IRR of -6%. Which project should be cut?

A.The more negative the results are, the better, so cut project A.

B.Less-negative results are bad, so cut project B.

C.To reduce losses, choose project C.

D.Cancel projects A, B, and C, as they are all losing money.

9.A sponsor is trying to determine which project has the greatest business value—one that returns $4 million in 3 years or one that returns $5 million in 4 years. The interest rate is 5 percent for borrowing capital to develop the projects. The best approach to determine the highest value project is:

A.Select the "$5 million in 4 years” project, since it returns the highest amount.

B.Select the "$4 million in 3 years" project, since it has the shorter payback period.

C.Calculate the NPV of the projects, and choose the project with the highest value.

D.Calculate the NPV of the projects, and choose the project with the lowest cost.

10.Which of the following statements best describes the role of the project charter?

A.Forms the basis of the prioritized backlog and identifies the work to be completed by the project

B.Describes the issues and risks the project may encounter and describes mitigation strategies for avoidance

C.Defines the what, why, when, who, and how details of the project and provides authority to proceed

D.Outlines the roles and responsibilities for the project stakeholders and identifies any third-party companies

11. Your new team spends 10 minutes waiting for people to turn up for the daily stand-up meeting, which then lasts 15 minutes. The stand-up meeting is then followed by another 5 minutes of the team discussing the hockey game last night, and yet another 5 minutes brainstorming issues to solve. Calculate the total cycle time of this process (assuming that the hockey discussion is not categorized as a value-adding activity).
    A.15 minutes
    B.20 minutes
    C.30 minutes
    D.35 minutes

12. Your sponsor has asked for clarification on when releases of your product will ship and what those releases will contain. Which agile deliverable would best answer their needs?
    A.Product demo
    B.Product roadmap
    C.Product backlog
    D.Product owner

13. You have been asked to outline the basics of agile contracting for your steering committee. Which of the following statements best matches the recommended approach taken to contracting on agile projects?
    A.The contract is worded to allow for early completion of scope, and acceptance is based on items matching the original specification.
    B.The contract is worded to allow for reprioritization of scope, and acceptance is based on items matching the original specification.
    C.The contract is worded to allow for early completion of scope, and acceptance is based on items being fit for business purpose.
    D.The contract is worded to allow for reprioritization of scope, and acceptance is based on items being fit for business purpose.
    E.

14.In agile methods the terms MoSCoW and “Priority 1, Priority 2, Priority 3" are forms of:
    A.Estimation
    B.Risk identification
    C.Prioritization
    D.Reporting

15. Customers or business representatives are involved in the prioritization of work for which primary benefit?
    A. To proactively engage the business between acceptance testing cycles
    B. To answer business-related questions about the requirements
    C. To aid communications about when features will be delivered
    D. To better reflect the business needs and wants for the project

16. Which of the definitions shown below best describes the agile concept of incremental delivery?
    A. The product is built and evaluated in chunks that are selected and developed by the development team.
    B. The product is built and evaluated in chunks that, by the end of the project, comprise the complete product, including agreed-upon change requests that arose during the process.
    C. The product is built and evaluated in chunks that, by the end of the project, comprise the complete product, excluding agreed-upon change requests that arose during the process.
    D. The product is built and evaluated in agreed-upon increments that, by the end of the project, comprise the complete product, factoring in the support demand that will be required after delivery.

17. Your team on average has to wait 3 business days (24 business hours) to get approval to make changes to their user acceptance environment. Making the changes usually takes about an hour. Which of the answers below represents the correct process efficiency time, rounded to one decimal place?
    A. 4.0%
    B. 4.1%
    C. 4.2%
    D. 4.3%

18. You have been asked to explain what the project team has been doing during iteration 0, an early portion of the project. Which of the following graphs would be a good choice?
    A. Budget burn up chart
    B. Risk burn down graph
    C. Work breakdown structure
    D. Project organization chart

ANSWERS

1.Answer: D

Explanation: Since this question is looking for the answer that is NOT an advantage of limiting WIP, if an option is true, that means it is not the answer we are looking for. Limiting WIP does reduce the potential need for rework. It also improves process efficiency and helps us find production bottlenecks. The only option listed here that is NOT an advantage of limiting WIP is the one that refers to maximizing resource utilization. Limiting WIP focuses on optimizing throughput, not resources, and we may actually decrease resource optimization to get more throughput.

2.Answer: C

Explanation: The answer is project X, simply because it has the larger internal rate of return (IRR), We don't need to consider the payback period or the NPV because the question asks which project has the better rate of return, which is its IRR.

3.Answer: C

Explanation: To get the correct answer, we need to examine each option to find the one that is NOT a prioritization scheme (or that sounds least likely to be one). Prioritization based on business value is the most common prioritization method, so it is certainly a valid prioritization scheme. Prioritization based on risk is also a valid prioritization scheme. Likewise, prioritization based on both business value and risk is valid. The only option presented here that is NOT a valid prioritization scheme is iteration velocity. This term measures the work done and the team's capacity; it is not a prioritization scheme. So prioritization based on iteration velocity is the correct answer.

4.Answer: B

Explanation: Creating a product roadmap is a release planning technique. A roadmap is not a list of screens and reports, because we can build a roadmap for a product even if the project has no screens and reports. The other options of deployment instructions and a backlog prioritization scheme are also incorrect.

5.Answer: C

Explanation: Since agile contracts have to take into account the difficulty of defining the required product characteristics in advance, they need to work even when the specs are not fully defined. Agile contracts can be used with time and materials agreements but they also work well for fixed price projects where a cost limit cannot be exceeded and functionality is trimmed to meet the budget. A major reason agile contracts were developed was to accommodate changes, so that is the correct answer.

Answer: C

Explanation: Here we have a few reasonable choices to consider, but only one option is the main reason. Putting risks in the backlog does prevent us from having to maintain a separate list, but that is not the reason we do it. It also helps keep the team focused on the risk, but again, that is not the main reason we do it. Preventing the team from forgetting about the risk is a lot like the previous idea, and is also not the real reason we put risks into the backlog. Instead, the main reason is to ensure that those items get worked on in early iterations, since delivering value early also means removing opportunities for value reduction.

6.Answer: C

Explanation: Here we need to eliminate the incorrect options to find the remaining valid combination. At first glance, “customer importance and business urgency" sounds like it might be a plausible option—but since both components are based on the customer s preference, they are similar enough that this isn’t really a valid mix. “Risk reduction and technical dependency" can be eliminated because this combination doesn’t take into account the customer s priorities—a mix is not balanced (and therefore not valid) if it doesn’t consider value to the customer. “Team preference and customer value” is not a valid mix because team preference is not a valid selection criteria on its own. Although the team might prioritize a story for good reasons (for example, because it reduces risk), we don’t know that from the question. The correct option is “customer value and risk reduction.” These are the classic elements that we seek to balance in prioritizing work, so this option is the most valid story prioritization mix of the choices given here.

7.Answer C:

Explanation: Without more information, such as ROI (though we are actually looking at loss instead of return in this case), the logical option is to cut the project that has the highest rate of loss. This means that we would cut project C, since it has the largest negative internal rate of return. Canceling all the projects is not the answer, since the question says that company X has decided to cancel ONE of its projects. On the exam, look out for questions like this that appear to present multiple “correct” answers but include information within the question that rules out certain choices.

8.Answer: C

Explanation: To evaluate the value of two projects that are completed at different times, we can use net present value (NPV) to translate the amounts into today’s values. Therefore, the approach of calculating the NPV and choosing the project with the highest value is the way to go. The other answers may make valid arguments, but the question is asking for the BEST approach, which is to calculate NPV.

9.Answer: C:

Explanation: The project charter defines the what, why, when, who, and how details of the project and provides authority to proceed. It may also list the scope that ends up in the backlog and the risks that should go into a risk register or risk list, but these are all just portions of the charter that do not "Best describe the role."

10.Answer: D

Explanation: Since the total cycle time is the value-add time + the nonvalue-add time, we do not actually need to determine what falls under the value-add category. Instead, we just need to add up all the times involved: 10+15 + 5 + 5 = 35 minutes.

11.Answer: B

Explanation: The product roadmap shows release dates and the high-level contents of releases, so it would be the best deliverable for answering these questions. A product demo might be good for showing the sponsor what has already been built, but demos are not targeted at communicating the release schedule for upcoming features, and neither is the backlog. The product owner will likely know the answers to these questions, but the question asked "Which agile deliverable", and the product owner is not a deliverable.

12.Answer: D

Explanation: Two components common to agile contracts are an ability to reprioritize work and the goal of satisfying the business, rather than conforming to a spec. The closest match to these characteristic is the option "The contract is worded to allow for reprioritization of scope and acceptance is based on items being fit for business purpose."

13.Answer: C

Explanation: MoSCoW and "Priority 1, Priority 2, Priority 3" are forms of prioritization. They are ways the business representatives or customer can classify requirements or user stories to rank the work that needs to be done by the project.

14.Answer: D

Explanation: While involving the customer does help with communications and business question clarifications, the primary reason we engage them is to better reflect the business needs and wants for the project. Keeping the customer busy is not a concern.

15.Answer: B

Explanation: The best explanation here is “The product is built and evaluated in chunks that, by the end of the project, comprise the complete product, including agreed-upon change requests that arose during the process.” The other choices contain concepts that are contrary to the incremental delivery approach or are completely unrelated. We don t just let the team, rather than the business or customer, select work, and we wouldn’t exclude agreed-upon changes. As promoted by the Agile Manifesto, agile methods can harness change requests to deliver competitive advantage. And support demand has nothing to do with incremental delivery.

16.Answer: A

Explanation: Process cycle efficiency = Total value-add time/Total cycle time. So, in this example, the value-add time is 1 hour, and the total cycle time = 24 + 1 = 25 hours. This gives us 1/25, which is 0.04, or 4 percent. If you used 24 for the total cycle time, you would have calculated 1/24 for an answer of 4.2%, which is incorrect.

17.Answer: B

Explanation: Since iteration 0 is concerned with establishing tools and environments and proving approaches, a risk burn down graph is a good chart to show what the team has been working on. While there may not be business functionality to speak of, hopefully the team has reduced some technical risks and proven some key approaches that will be used.

6.3. Examen práctico capítulo 2.3: Implicación de los Stakeholders

QUESTIONS.

1.User stories feature which of the following components?

A.Role, Function, Benefit

B.Reason, Functionality, Benefit

C.Risk, Function, Benefit

D.Risk, Feature, Business Reason

2.Wireframes, personas, and user stories can all play a part in understanding stakeholder objectives. Which of the circumstances outlined below would be a good fit for the use of personas?

A.When the conversation is centered on the high-level flow of a process

B.When we are trying to better understand stakeholder demographics and general needs

C.When we need to capture the high-level objective of a specific requirement

D.When we want to communicate what features will be included in the next release

3.When comparing communication styles, which of the following is true?

A.Paper-based communication has the lowest efficiency and the highest richness.

B.Face-to-face communication has the highest efficiency and the lowest richness.

C.Paper-based communication has the highest efficiency and the lowest richness.

D.Face-to-face communication has the highest efficiency and the highest richness.

4.Your project team has some conflict, and you are trying to diagnose what level of conflict they are at. You notice that statements such as “Team B has no clue again P are becoming commonplace. What level of conflict would you say the team is experiencing?

A.Level 1

B.Level 2

C.Level 3

D.Level 4

5.High-visibility project displays are sometimes called:

A.Project radiators

B.Information refrigerators

C.Information radiators

D.Project distributors

6.The INVEST mnemonic for user stories looks for attributes that include:

A.Independent, Negotiable, Smart

B.Valuable, Easy-to-use, Timely

C.Negotiable, Estimatable, Small

D.Independent, Valuable, Timely

7.Bum down charts display the following attributes on their axes:

A.X axis = points remaining; Y axis = calendar time

B.X axis = estimated time remaining; Y axis = calendar time

C.X axis = calendar time; Y axis = estimated effort remaining

D.X axis = calendar time; Y axis = calendar time

8.Wireframe models help agile teams to:

A.Test designs

B.Confirm designs

C.Configure reports

D.Track velocity

9.What is velocity not used for?

A.Gauging the work capacity of the team

B.Checking our release plan validity

C.Getting a sense of work done per iteration

D.Defining feature requirements

10.The definition of done is frequently discussed with stakeholders so that:

A.Functionality can be negotiated until the last responsible moment.

B.Everyone has a clear understanding of what completion means.

C.Team members get to improve their negotiation skills.

D.Active listening reveals previously undiscussed requirements.

11.Wireframes, personas, and user stories can all play a part in understanding stakeholder objectives. Which of the circumstances outlined below would be a good fit for the use of user stories?

A.When the conversation is centered on the high-level flow of a process

B.When we are trying to better understand stakeholder demographics and general needs

C.When we need to capture the high-level objective of a specific requirement

D.When we want to communicate what features will be included in the next release

12.Agile modelling aims to:

A.Capture the intent of the design in a barely sufficient way

B.Capture the intent of the design in detail-oriented way

C.Deliver extraneous documentation for the project

D.Recognize that the value of modeling increases with time spent

13.Which of the following is a valid requirements hierarchy progression from large to small?

A.Task, User Story, Epic

B.Feature, User Story, Task

C.Epic, User Story, Feature

D.Epic, Task, Feature

14.Your team seems to unproductively debate even trivial decisions. To help them along, you could try:

A.Fist-of-five voting

B.Bare fist fighting

C.Wideband Delphi

D.Retrospectives

15.Which of the following statements is true for measuring team velocity?

A.Velocity is not accurate when there are meetings that cut into development time.

B.Velocity measurements are disrupted when some project resources are part-time.

C.Velocity tracking does not allow for scope changes during the project.

D.Velocity measurements account for work done and disruptions on the project.

16.You have been assigned to lead a geographically distributed agile team. To assist with communication the best option would be to:

A.Ask team members to send photos of themselves so you know what they look like.

B.Choose a common language for project communications.

C.Set up some initial face-to-face meetings for everyone to meet.

D.Define common working hours so everyone can better communicate.

17.The relationship between leadership and management in agile methods is:

A.Leadership replaces all aspects of management.

B.Leadership is subservient to management.

C.Management and leadership are used together.

D.Management and leadership are not compatible.

18. Wireframes, personas, and user stories can all play a part in understanding stakeholder objectives. Which of the circumstances outlined below would be a good fit for the use of wireframes?
    A. When the conversation is centered on the high-level flow of a process
    B. When we are trying to better understand stakeholder demographics and general needs
    C. When we need to capture the high-level objective of a specific requirement
    D. When we want to communicate what features will be included in the next release

ANSWERS

1.Answer: A

Explanation: The format of a user story is “As a <Role>, I want < Functionality >, so that <Business Benefit >. Risks and reasons are not common attributes of user story templates.

2.Answer: B

Explanation: Personas would be a good fit when we are trying to better understand stakeholder demographics and general needs.

3.Answer: D

Explanation: Recall that in the communication effectiveness graph, paper-based communication is in the lower left corner (low effectiveness, low richness) and face-to-face communications (two people at a whiteboard) are in the upper right corner (high effectiveness, high richness).

4.Answer: C

Explanation: “Team B has no clue again” falls into the category of overgeneralizations, presumptions, and magnified positions, which indicates a level 3 type conflict. At level 2, they wouldn’t have said the “no clue again” overgeneralization, and at level 4, the language likely would have been more ideological and hostile.

5.Answer: C

Explanation: The term Alistair Cockburn coined for these high-visibility displays of project data is “information radiators.” Cockburn compares this to the practice of locking information away in plans that few people get to see (information refrigerators).

6.Answer: C

Explanation: INVEST is the mnemonic for Independent, Negotiable, Valuable, Estimatable, Small, and Testable. So option C is the only choice with a valid combination of the attributes. Smart and Timely are not part of the acronym.

7.Answer: C

Explanation: Burn down charts always show calendar time on the X axis. They may show either points remaining or estimated time remaining on the Y axis. Option D, which plots calendar time against calendar time, would just produce a straight line that does not tell us much.

8.Answer: B

Explanation: Wireframes are used to confirm designs; they are not detailed enough to test designs or configure reports.

9.Answer: D

Explanation: Velocity is a versatile metric that does indeed provide insight into team capacity (how much work the team can likely deliver in an iteration), release plan validity (if we are averaging 50 points per iteration and have 500 points of work remaining, then it looks like we are on track to complete the project within the 11 iterations remaining), and finally work done per iteration (the team is averaging 50 points per iteration). It is not used in defining the requirements of a feature, but velocity does impact which features can be built within a given period of time.

10.Answer: B

Explanation: The reason to have frequent discussions on the definition of done is to help mitigate the differences in opinion that can result when new functionality is described from a variety of viewpoints. The definition of done is not intended to be used to negotiate functionality, improve negotiation skills, or surface new requirements (although that may occur). Instead, we have these discussions to make sure everyone has a common understanding of what completion or success will look like.

11.Answer: C

Explanation: User stories are used when we need to capture the high-level objective of a specific requirement

12.Answer: A

Explanation: Agile models are lightweight, barely sufficient (just enough detail) models to capture the high-value benefits of modeling without too much of the lower-value development time taken to create very detailed or polished models. We want to focus on the product being developed, rather than on generating extraneous documentation.

13.Answer: B

Explanation: The question asks for the progression from large to small. Since Task is the smallest planning unit, options that end in Epic or Feature are not a valid progression. (Note: Although you will not be tested on the alternate requirements hierarchies that include epics, you may still see the term “epics” as a distractor in answer choices.)

14.Answer: A

Explanation: When struggling to make simple decisions effectively, we should introduce participatory decision models such as fist-of-five voting. Wideband Delphi is a group estimation approach that would only really help if the question said they were having difficulty agreeing on estimates. Retrospectives are group-based reflection and learning sessions that would probably not help with decision making, and bare fist fighting is, sadly, now in conflict with PMI s Code of Ethics and Professional Conduct.18

15.Answer: D

Explanation: Velocity is a measure of work done (and team capacity) net of all the interruptions and other things that occur on projects. So it does account for meetings, part-time resources, and scope changes that are all common on today's projects.

16.Answer: C

Explanation: If possible, setting up some initial face-to-face meetings for everyone to meet is an effective way of improving remote communications later in the project. Once people have met face-to-face, it is generally much easier to follow up with e-mail, phone calls, etc. Defining common working hours or a common language might appear to help, but it can also be viewed as not being respectful. Sending photos is also unlikely to assist much, and it is certainly not the best option.

17.Answer: C

Explanation: Agile methods employ a combination of management and leadership. Leadership neither totally replaces nor is subservient to management, and since the two approaches can be used together, they are not incompatible.

18.Answer: A

Explanation: Wireframes would be a good fit when the conversation is centered on the high-level flow of a process.

6.4. Examen práctico capítulo 2.4: Impulsar las prácticas de rendimiento del equipo

QUESTIONS.

1.Which are the recommended pairings of adaptive leadership and team phases?

A.Supporting and Norming, Delegating and Performing, Directing and Forming, Coaching and Storming

B.Directing and Forming, Coaching and Storming, Delegating and Norming, Supporting and Performing

C.Directing and Forming, Supporting and Storming, Delegating and Norming, Coaching and Performing

D.Directing and Forming, Supporting and Norming, Coaching and Performing, Delegating and Storming

2.__is the agile name given to undocumented information supported through team communications.

A.Unwritten knowledge

B.Tribal knowledge

C.Tacit knowledge

D.Common knowledge

3.Self-organizing teams are most readily characterized by their ability to:

A.Do their own filing

B.Sit where they like

C.Make local decisions

D.Make project-based decisions

4.Which of the following is not a recommendation for one-on-one coaching?

A.Meet them a half-step ahead

B.Create positive regard

C.Partner with managers

D.Let the team resolve conflicts

5.High-performing teams feature which of the following sets of characteristics?

A.Consensus-driven, empowered, low trust

B.Self-organizing, plan-driven, empowered

C.Consensus-driven, empowered, plan-driven

D.Constructive disagreement, empowered, self-organizing

6.Which of the following sets of tools is least likely to be utilized by an agile team?

A.Digital camera, task board

B.Wiki, planning poker cards

C.WBS, PERT charts

D.Smart board, card wall

7.Which of the following is a valid list of the quadrants of emotional intelligence?

A.Self, Others, Recognize, Optimize

B.Self, Others, Regulate, Recognize

C.Self, Team, Regulate, Recognize

D.Self, Team, Recognize, Optimize

8.The three questions answered in daily stand-up meetings aim to:

A.Identify problems, discuss accomplishments

B.Identify opportunities, discuss accomplishments

C.Fix problems, discuss accomplishments

D.Fix problems, discuss work planned

9.Who typically has the best insight into task execution?

A.Project managers

B.Team members

C.ScrumMasters

D.Agile coaches

10.Which of the following are recommended brainstorming techniques ■

A.Quiet writing

B.Decision spectrum

C.Fist-of-five voting

D.Participatory decisions

11.Servant leadership roles include:

A.Shielding team members from interruptions

B.Leading conflict resolution

C.Determining which features to include in an iteration

D.Assigning tasks to save time

12.Which of the following statements is false for distributed teams?

A.Should consider instant messaging tools

B.Should have an easier Storming phase

C.Need to spend more effort communicating

D.Have a higher need for videoconferencing

13.Another project manager of an agile project in your organization comes to you for advice. She is having trouble getting her team to take ownership of the project and get comfortable selecting the work to be done. She keeps finding herself making the decisions and directing their work. What advice can you offer her?

A.Play round-robin at stand-up meetings, assigning a different person each day as the decision maker to get them comfortable in the role.

B.Implement an incentive plan and officially report any lack of participation to team members* functional managers.

C.Meet them halfway and work with their functional managers to align each team member's goals with the project goals.

D.Explain to them that agile teams self-select their work and tell them to get on with it.

14.Tuckman’s stages of team formation and development progress in what sequence?

A.Norming, Storming, Performing, Forming

B.Norming, Storming, Forming, Performing

C.Forming, Storming, Performing, Norming

D.Forming, Storming, Norming, Performing

15.Blanchard and Hersey s adaptive leadership phases, in sequence, are:

A.Supporting, Directing, Coaching, Delegating

B.Supporting, Coaching, Directing, Delegating

C.Delegating, Coaching Supporting, Directing

D.Directing, Coaching, Supporting, Delegating

16.Which of the following emotional intelligence pairings is an appropriate combination?

A.Self-management deals with influence

B.Self-awareness deals with self-control

C.Social skills deals with self-confidence

D.Social awareness deals with empathy

17.The primary reason constructive disagreement is valued on high-performing agile teams is to:

A.Weed out the weak

B.Test requirements for robustness

C.Generate buy-in for decisions

D.Build negotiation skills

18.At what team formation and development phase is conflict likely to be highest?

A.Forming

B.Fuming

C.Storming

D.Debating

ANSWERS.

1.Answer: A

Explanation: The pairings align the Tuckman model with the Blanchard and Hersey model. The correct pairings are Directing and Forming, Coaching and Storming, Supporting and Norming, and finally Delegating and Performing. Option A is correct, even though it has them out of sequence; the question just asked for the pairings, not the correct sequence.

2.Answer: C

Explanation: While the terms "tribal knowledge," "common knowledge," and "unwritten knowledge" may still convey the idea, the correct term is "tacit knowledge."

3.Answer: C

Explanation: This question can be confusing because teams may have the opportunity to make some project-based decisions, but remember to always look for the BEST choice in the options presented. Self-organizing teams primarily have control over local decisions related to the project execution. For example, they may decide what to do next and how to solve a technical problem. Sponsors typically make external decisions, such as increasing the budget or extending the schedule. Doing their own filing or sitting where they like may or may not occur, but these choices are not readily associated with characteristics of a self-organizing team. Therefore, choice C is the best option.

4.Answer: D

Explanation: The recommendations for one-on-one coaching are to meet team members a half-step ahead, create positive regard, partner with managers, and guarantee safety. While we should let the team work through conflict, this is not a recommendation for one-on-one coaching.

5.Answer: D

Explanation: Through a process of elimination, we can determine that the correct answer is choice D. High-performing teams work in high-trust, rather than low-trust, environments, and they are consensus-driven, not plan-driven.

6.Answer: C

Explanation: While tools like work breakdown structures and PERT charts can still work on agile projects, these tools are used less than the other choices presented in the question. This is a result of the frequent reprioritization of work and the high rates of changes. Agile teams tend to use tools that take a minimal amount of time to update and keep current.

7.Answer: B

Explanation: Vertically the emotional intelligence model has two columns—Self and Others. Horizontally it has two rows representing Recognize and Regulate. “Optimize” and “Team" are not model category names.

8.Answer: A

Explanation: The three questions asked at stand-up meetings are: I) “What have you worked on since the last meeting?” 2) “What do you plan on finishing today?” and 3) “Are there any roadblocks or impediments to your work?” These questions aim to discuss accomplishments, discuss work planned, and identify problems. Fixing problems or identifying opportunities, while worthy goals, are not objectives of the stand-up meeting.

9.Answer: B

Explanation: Servant leadership recognizes that the “doers” of the work, the team members, are closest to the work and therefore have the best insight into its execution. Project managers, ScrumMasters, and agile coaches should defer to the teams decision regarding how best to execute the work.

10.Answer: A

Explanation: Recommended brainstorming techniques include quiet writing, round-robin, and free- for-all. Decision spectrum, fist-of-five voting, and participatory decision models are team decision-making tools.

11.Answer: A

Explanation: Servant leaders recognize that it is the team that adds value, so their role is to shield the team from interruptions. They should not immediately take the lead in resolving conflicts but first let the team try to resolve issues on their own. And the team determines the iteration features; they are not dictated by the project leader. Task assignment is not a servant leadership role; agile teams are encouraged to select their own work, based on the backlog and their skills.

12.Answer: B

Explanation: Distributed teams have a tougher time, rather than an easier time, getting through the Storming phase since it is difficult to resolve conflicts when people are geographically distributed. The other statements are true for distributed teams.

13.Answer: C

Explanation: The project manager in this situation should assume a coaching role with her team to help the team members get to the point where they are comfortable selecting their own work. The recommendations for one-on-one coaching include meeting team members a half step ahead, guaranteeing safety, partnering with managers, and creating positive regard. Assigning someone as a decision maker at stand-up meetings isn't an effective approach, since teams should be consensus- driven. Incentive plans can be useful and a project manager should work with a functional manager to help resolve issues, but what the team really needs is guidance, not strictly rewards and punishments. Simply explaining that agile teams self-select their work isn't enough to get team members to the point where they are comfortable assuming more ownership of the project.

14.Answer: D

Explanation: The correct sequence is forming, storming, norming, performing.

15.Answer: D

Explanation: The correct sequence is directing, coaching, supporting, delegating.

16.Answer: D

Explanation: The only correct choice is the combination of social awareness and empathy. Self-management is paired with self-control, not influencing; self-awareness is paried with self-confidence; and social skills is paired with self-control.

17.Answer: C

Explanation: Constructive disagreement is a form of healthy conflict, where team members work through issues to find a solution that is right for the team and the project. As a result, constructive disagreement generates team buy-in for decisions. Agile methods do not seek to "weed out the weak," and while constructive disagreement could help clarify requirements and build negotiation skills, those are not the primary reasons it is valued on a project.

18.Answer: C

Explanation: The stage where conflict is highest is the storming phase. When the team is in the forming stage, they are still getting to know each other and move from that phase into storming. Fuming and debating are not team formation and development phases.

6.5. Examen práctico capítulo 2.5: Planificación adaptativa

QUESTIONS.

1.Your team committed to delivering 10 story points this iteration, but it looks like you will only complete 8. You should:

A.Extend the iteration.

B.Add more resources to the team,

C.Complete 8 points, and put 2 back in the backlog.

D.Adjust the iteration plan from 10 points down to 8.

2.Which of the following collections of planning units is most typical for agile projects?

A.A release plan containing multiple projects, each with multiple iterations

B.A project plan containing multiple releases, each with multiple iterations

C.A project plan containing multiple iterations, each containing multiple releases

D.An iteration plan containing multiple projects, each containing multiple releases

3.When converting size estimates to duration, remember to:

A.Ignore distractions and use ideal time

B.Divide the timebox capacity by the number of stories

C.Factor in distractions and use available time

D.Calculate the payback period for the estimated duration

4.The project management office is auditing your agile project and asks to see your iteration plans, notice that only the next couple of iterations have plans. As a result, they give the project a “red flag” for having incomplete plans. The most responsible thing to do is:

A.Explain the principles of progressive elaboration.

B.Create detailed iteration plans for the remainder of the project.

C.Ignore them, since they clearly have no right to be reviewing your project.

D.Have the team create detailed plans for the rest of the project.

5.Which of the following statements correctly describes agile planning?

A.Plan at multiple levels, and have managers create iteration plans.

B.Use appropriate estimate ranges, and exclude diversions/outside work.

C.Plan at multiple levels, and have team members create iteration plans.

D.Use fixed-point estimates, and base projections on completion rates.

6.Which of the following is not a characteristic of agile estimation?

A.Team-based

B.Collaborative

C.Iterative

D.Fixed-point

7.When conducting an iteration planning meeting using a Scrum approach, which of the following statements is not true?

A.The product owner is responsible for the backlog priorities.

B.The team is responsible for the estimates.

C.The team breaks down user stories into tasks.

D.The ScrumMaster selects the topic items off the backlog.

8.You are leading a team with an average velocity of 50 points per iteration. Another team of the same size in your organization is working on a project with similar complexity. The other team s velocity is averaging 75 points per month. Your team should:

A.Undertake affinity estimating to check their estimates

B.Work longer hours

C.Ignore the difference

D.Acquire additional resources

9.Estimates should be presented as ranges to:

A.Allow for change requests

B.Keep the sponsors flexible

C.Allow for scope creep

D.Represent uncertainty in the estimates

10.Which of the following pieces of information are you least likely to find in an agile charter document?

A.A list of who will be engaged on the project

B.A description of how it will be undertaken

C.An explanation as to why the project is being done

D.A precise estimate for the project

11.An executive wants help evaluating a proposed three-year project against two proposed one-year projects. Which business case metric would be most helpful?

A.NPV

B.KOI

C.MMF

D.Velocity

12.Your team is averaging 40 story points per two-week iteration. They have 200 points' worth of functionality left in the user story backlog. How many weeks do you expect it will take until development is completed?

A.2.5

B.5

C.10

D.20

13.When using story points to estimate a project, which of the following statements is most accurate?

A.The team owns the definition of what constitutes a story point.

B.There should be a company-wide standard definition of a story point.

C.The definition of a story point is refactored every iteration.

D.Story points can be used for iteration planning but not release planning.

14.On agile projects, generally midcourse adjustments are:

A.Not necessary

B.The norm

C.The exception

D.Not required

15.When calculating final project costs, which of the following expressions best outlines the basic concept?

A.Time + (Rate x Other project costs)

B.(Time x Rate) - Other project costs

C.Time x Rate x Project duration

D.(Time x Rate) + Other project costs

16.Affinity estimating is the process of:

A.Averaging the over- and under-estimations.

B.Checking that stories given the same size estimate are of equivalent magnitude.

C.Checking that stories in the same functional areas are of equivalent magnitude.

D.Estimating your favorite stories first.

17.The term “progressive elaboration” means:

A.Scope always keeps growing.

B.Plans are refined as more details emerge.

C.Development velocity is increasing.

D.Estimates are changed to actuals after completion.

18.You are a full-time ScrumMaster on an agile team. A team member becomes ill partway through an iteration in which the team committed to deliver 25 story points. Which action is most appropriate

A.Work the remaining team longer hours.

B.Send work home to the sick team member.

C.Start development yourself to assist the team.

D.Deliver what you can within the sprint.

ANSWERS.

1.Answer: C

Explanation: An iteration is timeboxed, so it must be left at its normal duration. You wouldn't change the plan for the iteration, and you typically wouldn't choose to expand the team. Instead, work that is not completed within the iteration is returned to the backlog, so the choice of completing 8 points and returning 2 points to the backlog is the correct option.

2.Answer: B

Explanation: The correct hierarchy is that a project contains one or more releases, which in turn contain one or more iterations. Therefore, the correct choice is a project plan containing multiple releases, each with multiple iterations.

3.Answer: C

Explanation: When converting size estimates to duration, we need to consider the availability of resources, so the choice of "Factor in distractions and use available time" is correct. If someone is only 50 percent available, a task will take twice as long as it would if the person was fully dedicated to the project, and perhaps even longer due to a loss of productivity from task switching.

4.Answer: A

Explanation: An incomplete set of iteration plans may be a surprise to a PMO that is not familiar with agile methods. When faced with this type of situation, you should explain the benefits of agile planning and how an agile approach ties into the concepts of progressive elaboration and rolling wave planning, which are discussed in the PMBOK® Guide.8 Making up plans too early is a poor use of time on an agile project and could mislead stakeholders. The choice of ignoring the request is incorrect, because it is counter to the "Respect" principle defined in PMIs Code of Ethics and Professional Conduct.9

5.Answer: C

Explanation: The only correct combination is to plan at multiple levels and have team members create iteration plans. All of the other choices contain incorrect elements. Managers do not create iteration plans—teams do. Diversions or outside work are included when determining availability, and we use range estimates on agile projects, not fixed-point estimates.

6.Answer: D

Explanation: Agile estimation is team-based, collaborative, and iterative. We do not use fixed-point estimates on our project. Instead, we estimate in a range.

7.Answer: D

Explanation: The product owner is responsible for the backlog priorities, and the team is responsible for the estimates and breaking down the stories into tasks. The only statement that is not true is that the ScrumMaster selects the top items off the backlog.

8.Answer: C

Explanation: Velocity is team-specific and unique to that team. In other words, a story point for one team may not have the same value as a story point for another team. Therefore, it is not appropriate to compare velocities between teams. The best choice would be to ignore the difference.

9.Answer: D

Explanation: We present estimates as ranges to show the level of uncertainty in the estimates and to manage stakeholder expectations.

10.Answer: D

Explanation: Due to the uncertainty and potentially higher levels of change and risk involved in agile projects, you would be least likely to see a precise cost or time estimate in an agile charter. This does not mean it's out of the question, especially if a project is working with a fixed budget, but it is the LEAST likely choice.

11.Answer: A

Explanation: The correct choice is NPV, or net present value, because it converts multiyear returns on investment to a value in today s terms. It is the best option from the choices presented for evaluating projects with different durations.

12.Answer: C

Explanation: This question doesn't provide information to indicate any differences in the team's availability or known project distractions, so we can do a fairly straightforward calculation to get the answer. If we average 40 points per iteration, we should get through a 200-point backlog in 5 iterations (200 / 40 = 5). Each iteration is 2 weeks long, so 5 iterations is equivalent to 10 weeks (5x2= 10). The key to answering this question correctly is noticing that the question asked for the number of weeks, not the number of iterations.

13.Answer: A

Explanation: The correct option is that the team owns the definition of the story point. Having the team define the story point creates a sense of ownership and a common understanding. In contrast, company-decreed definitions lack this sense of team ownership. The other options are incorrect because frequently changing the definition (refactoring it every iteration) makes it difficult to track velocity and check release plans. Story points, and velocity in terms of the number of points completed, are used for both iteration and release planning.

14.Answer: B

Explanation: We expect midcourse adjustments on agile projects, so the option that states they are the norm is the correct choice.

15.Answer: D

Explanation: Final costs are (Time x Rate for resources) + Any additional project costs.

16.Answer: B

Explanation: Affinity estimation and triangulation check that stories assigned the same size value are of equivalent effort.

17.Answer: B

Explanation: Progressive elaboration is the ongoing refinement of plans as more information becomes available.

18.Answer: D

Explanation: In such a situation, we deliver what we can within the sprint. Sending work home or working the team longer are counterproductive actions in the long-term that would go against the agile principle of maintaining a sustainable pace. Starting work on development yourself would leave nobody to do the ScrumMaster role. It is better to just deliver what you can within this iteration and explain the variance.

6.6. Examen práctico capítulo 2.6: Detección y resolución de problemas

QUESTIONS.

1.Which activity does not actively reduce a problems position on the cost of change curve?

A.Pair programming

B.Test-driven development

C.Velocity

D.Acceptance test-driven development

2.Your team has identified a number of problems during a retrospective session and you now want to help understand and fix the issues. The sequence of activities you should undertake is:

A.Generate insights, gather data, decide what to do

B.Gather data, generate insights, decide what to do

C.Decide what to do, gather data, generate insights

D.Decide what to do, generate insights, gather data

3.The term used for describing errors that are missed by quality assurance and control processes and that make it into production is:

A.Cost of change

B.Undocumented features

C.Escaped defects

D.Change requests

4.Trends in project data are useful since they can offer insights and are what type of indicator?

A.Lagging metrics

B.Leading metrics

C.Lead times

D.Little's Law

5.The concept of control limits can be described as:

A.Visual controls that show acceptable ranges

B.Change control procedures to help manage flow

C.Budget processes that help teams estimate better

D.Exception reporting techniques with escalation paths

6.In software development, the process of frequently integrating new and changed code is known as:

A.Integrating continuously

B.Continuous integration

C.Integration control

D.Constant integrations

7.The shorthand summary of test-driven development activities is:

A.Red, Green, Yellow

B.Red, Green, Refactor

C.Green, Red, Clean

D.Green, Red, Refactor

8.On agile projects, the term “cycle time” usually refers to:

A.The average duration of an iteration

B.Time taken for work from start to finish

C.The time for SMEs to review the product

D.Time between releases of a product

9.A short period of proof-of-concept work is known as:

A.Risk stops

B.Spikes of iteration

C.Risk-based spike

D.Test-first design

10.Which of the following is not a form of frequent verification and validation?

A.Pair programming

B.Unit testing

C.Iteration demos

D.Iteration planning

11.You are managing an agile project and measure small amounts of variance in task durations. You should:

A.Undertake root-cause analysis to eliminate it

B.Engage the team in diagnosing the problem

C.Diagnose the issue as part of your ScrumMaster role

D.Accept some variance as inevitable

12.You are a team member on a software development project and have been asked to follow a test- driven development process. The sequence of activities you would undertake is:

A.Write code, write test, refactor

B.Write test, refactor, write code

C.Write test, write code, refactor

D.Write code, refactor, write test

13.When solving problems in a team setting, goals for problem resolution should be SMART. The characteristics of SMART goals are:

A.Simple, Marketable, Attainable, Risk Tolerant, Timely

B.Simple, Malleable, Attainable, Relevant, Testable

C.Specific, Measurable, Attributable, Relevant, Testable

D.Specific, Measurable, Attainable, Relevant, Timely

14.The purpose of continuous integration is to:

A.Always have a version of the software ready for release

B.Shorten lengthy compile times

C.Facilitate the practice of pair programming

D.Find code issues as soon as possible

15.Fast failure is the term used to describe:

A.Finding fatal design flaws early

B.Developing without proper testing

C.The last step of the ATDD process

D.An element of continuous integration

16.On a timeline graph used in the gathering data phase for team-based problem-solving, the line underneath the X-axis usually represents:

A.Team velocity at the time

B.Team satisfaction/feelings

C.Team risk rankings at the time

D.Team hours of work

17.When diagnosing problems, we ask “why” five times:

A.To get to the root cause of an issue

B.Because asking six times would be ridiculous

C.Because people unconsciously deceive themselves about problems three or four times

D.To fully gather data around the issue

18. You are a team lead helping your team decide what to do about some problems identified in a retrospective. What are some of the techniques you can use?
    A. Smart Subjects, Short Goals, Retrospective Planning Game
    B. Smart Subjects, Retrospective Poker Game, Smart Goals
    C. Short Subjects, Smart Goals, Retrospective Planning Game
    D. Retrospective Poker Game, Short Subjects, Circle of Questions

ANSWERS.

1.Answer: C

Explanation: The only item in this list not concerned with finding problems early is velocity. All the other activities help reduce the costs of changes.

2.Answer: B

Explanation: The sequence of activities for problem solving is gather data, generate insights, and then decide what to do.

3.Answer: C

Explanation: Escaped defects are errors that are missed by the quality assurance and control processes and make it into production.

4.Answer: B

Explanation: Trends are leading metrics, since they allow us to extrapolate from the series and determine what might happen in the future if the trend continues unchecked.

5.Answer: A

Explanation: Control limits are visual controls that show acceptable ranges. They do not help manage flow, improve estimates, or escalate issues.

6.Answer: B

Explanation: The process of frequently integrating new and changed code is known as continuous integration. The other choices here are either madeup or are not widely used terms.

7.Answer: B

Explanation: The shorthand summary of test-driven development is Red, Green, Refactor or Red, Green, Clean, in that order. This summary refers to the stages of first writing tests that will fail (generating a Red status), then writing code until it passes the tests (generating a Green status), and finally refactoring or cleaning up the code.

8.Answer: B

Explanation: Cycle time is the period of time from the start to the finish of work.

9.Answer: C

Explanation: A short period of proof-of-concept work is known as a risk-based spike.

10.Answer: D

Explanation: Iteration planning is not a way of verifying or validating a product. Instead, it is concerned with planning, scheduling, and capacity management.

11.Answer: D

Explanation: The question states that the variance amounts are small. Since some amount of variance is inevitable, we should simply accept it. It would be inappropriate to undertake root-cause analysis or other forms of investigation for this common cause variation.

12.Answer: C

Explanation: The correct sequence is to write a test (which should fail initially), write code until it passes the tests, and then refactor the design to clean things up before moving on to the next item.

13.Answer: D

Explanation: SMART stands for Specific, Measurable, Attainable, Relevant, Timely.

14.Answer: D

Explanation: The purpose of continuous integration is to find code issues as soon as possible. The other options are not related to continuous integration.

15.Answer: A

Explanation: Fast failure is the term used when a project hits an unsolvable roadblock early in its lifecycle.

16.Answer: B

Explanation: The timeline graph usually shows team satisfaction/feelings below the line to help the team recall how they felt and promote the recollection of additional data.

17.Answer: A

Explanation: When diagnosing problems, we ask “why” five times to get to the root cause of an issue.

18.Answer: C

Explanation: Techniques for deciding what to do about problems include Short Subjects, Smart Goals, and Retrospective Planning Game. “Short Goals” and “Retrospective Poker Game” are not valid techniques.

6.7. Examen práctico capítulo 2.7: Mejora continua

QUESTIONS.

1.PMIs Code of Ethics and Professional Conduct focuses on which of the following?

A.Service, Protection, Education, Honesty

B.Responsibility, Respect, Fairness, Honesty

C.Respect, Valor, Honesty, Transparency

D.Responsibility, Transparency, Fairness, Honesty

2.When conducting a retrospective, recognized “close the retrospective” activities include:

A.Plus/Delta; Helped, Hindered, Hypothesis; Return on Time Invested; Appreciations

B.Plus/Minus; Helped, Hindered, Hoped; Return on Time Invested; Applause

C.Plus/Delta; Helped, Hindered, Hoped; Return on Investment; Appreciations

D.Plus/Minus; Helped, Hindered, Hypothesis; Return on Time Delivered; Appreciations

3.You have been asked to review a different project team s recently enhanced methodology to assess its effectiveness and desirable characteristics. The types of characteristics that you should be looking for include evidence of:

A.A preference for face-to-face communications, significant process weight, recommendations for larger teams to use lighter methods

B.A preference for face-to-face communications, not too much process weight, recommendations for larger teams to use lighter methods

C.A preference for face-to-face communications, significant process weight, recommendations for larger teams to use heavier methods

D.A preference for face-to-face communications, not too much process weight, recommendations for larger teams to use heavier methods

4.The concept of knowledge sharing on an agile project is best characterized as:

A.Encouraged where possible and where the team shows an interest

B.Central to many of the practices undertaken

C.Undertaken if there is time left at the end of an iteration

D.Undertaken principally through stand-up meetings

5.Continuous improvement is a core component of which of the following set of agile practices?

A.Pair programming; daily stand-up meetings; WIP limits

B.Story points; daily stand-up meetings; demos, reviews, and retrospectives

C.Pair programming; daily stand-up meetings; demos, reviews, and retrospectives

D.Story points; daily stand-up meetings; WIP limits

6.When considering process tailoring, it is useful to keep in mind that the network of XP practices is:

A.Redundant

B.Duplicated

C.Optional

D.Balanced

7.The recommended stages of executing a retrospective, in sequence, are:

A.Decide what to do, set the stage, gather data, generate insights, close the retrospective

B.Decide what to do, gather data, set the stage, generate insights, close the retrospective

C.Set the stage, decide what to do, gather data, generate insights, close the retrospective

D.Set the stage, gather data, generate insights, decide what to do, close the retrospective

8.You are engaging in some process analysis and have been advised to watch out for the standard anti-patterns of poor methodology practice. The types of things you should be on the lookout for are processes that display signs of being:

A.One-of-a-kind, disciplined, heavy, embellished

B.One-size-fits-all, disciplined, heavy, embellished

C.One-size-fits-all, intolerant, heavy, embellished

D.One-of-a-kind, intolerant, embellished

9.Process tailoring is best undertaken on agile projects when:

A.There are difficulties in implementing agile practices

B.Experienced practitioners want to address an issue

C.The team needs new processes to keep them engaged

D.A boost in team velocity is needed to meet the schedule

10.Your project management office (PMO) has suggested your project could benefit from some self- assessment work at the next retrospective. Which of the following benefits would they most likely be looking to achieve from a self-assessment?

A.Improve personal and team practices

B.Gain insights for salary performance reviews

C.Identify personal traits for human resources counseling

D.Assess compatibilities for pair programming assignments

11.When undertaking process analysis, what are the success criteria that characterize a useful methodology?

A.The project got stopped, sponsorship remained intact, the team would work the same way again

B.The project got shipped, leadership remained intact, the team would work the same way again

C.The project got shipped, sponsorship remained intact, the team would work the same way again

D.The project got shipped, sponsorship remained intact, the team engaged in continuous improvement

12.When adopting a new agile practice, the general recommendations are to:

A.Develop the new approach yourself, try out the practice on local projects, inspect and review the findings early

B.Investigate the claims yourself, try out the practice on local projects, inspect and review the findings early

C.Investigate the claims yourself, try out the practice in small doses, inspect and review the findings early

D.Develop the new approach yourself, try out the practice in small doses, inspect and review the findings early

13.As a manager of an agile team, when should you collect lessons learned?

A.At the end of the project

B.Throughout the project

C.When projects go well

D.When projects go poorly

14.Information exchanges in agile methods are designed to:

A.Facilitate knowledge sharing

B.Leverage electronic knowledge-sharing tools

C.Maximize resource utilization

D.Generate stable specifications

15.Your sponsor is asking about tailoring the company's newly adopted agile methodology. Your advice should be:

A.Tailoring it will be a good way to learn the methodology

B.Tailoring it will be a good way to ease into the initial adoption process

C.We should tailor it first, then consider adopting it

D.We should try it first, then consider tailoring it

ANSWERS

1.Answer: B

Explanation: PMIs Code of Ethics and Professional Conduct focuses on Responsibility; Respect, Fairness, and Honesty. While the other attributes listed here are worthy, they are not the characteristics PMI is asking us to aspire to and maintain.

2.Answer: A

Explanation: The recognized activities that are used to close the retrospective are Plus/Delta; Helped, Hindered, Hypothesis; Return on Time Invested; and Appreciations. The options of “Plus/ Minus,” “Helped, Hindered, Hoped,” and “Return on Investment” are not valid retrospective closing activities.

3.Answer: D

Explanation: Favorable characteristics include a preference for face-to-face communications, not too much process weight, and recommendations for larger teams to use heavier methods. The option of significant process weight is generally the opposite of the barely sufficient goal we should be striving for. However, as teams get larger, we will inevitably need to use heavier weight methodologies to compensate for the reduced face-to-face communications and the increased difficulty of maintaining tacit knowledge.

4.Answer: B

Explanation: Knowledge sharing is central to many of the practices undertaken. It is true that stand-up meetings help the team members share information, but they are not the principal event for knowledge sharing. This practice is too important to be considered optional if the teams show an interest, or if there is time.

5.Answer: C

Explanation: Continuous improvement is core to the practices of pair programming; daily stand-up meetings; and demos, reviews, and retrospectives. It has little to do with WIP limits or story points.

6.Answer: D

Explanation: XP practices are considered to be balanced. If they were redundant, duplicated, or optional, then they would not likely be required and would not have been recommended in the first place.

7.Answer: D

Explanation: The recommended stages of executing a retrospective, in sequence, are: set the stage, gather data, generate insights, decide what to do, close the retrospective. You may be tempted to think that deciding what to do comes first, but it comes after we've set the stage.

8.Answer: C

Explanation: The anti-patterns we are warned to look out for are signs of the methodology being one-size-fits-all, intolerant, heavy, or embellished. If a methodology is one-of-a-kind, this might also be a warning sign, since it means the methodology has not been repeated, but it is not as concerning as a claim that the methodology is a one-size-fits-all approach. Such a claim demonstrates a lack of situational awareness. The characteristic of being disciplined is not something to be wary of, since agile methods are very disciplined; we should not mistake being disciplined for being process-heavy.

9.Answer. B

Explanation: Process tailoring is not to be undertaken just for implementation issues, to entertain the team, or as a scheme to increase velocity. Velocity may improve as a result, but the question asked when it is "best" to tailor processes. The correct answer is that it is best for experienced practitioners to undertake process tailoring when there is an issue to address.

10.Answer A

Explanation: The best option provided is to improve personal and team practices. While some of the other benefits might be tangential benefits, they are certainly not the focus of self-assessment.

11.Answer B

Explanation: The success criteria indicators are: the project got shipped, leadership remained intact, and the team would work the same way again. The fact that sponsorship remained intact is interesting, but this factor is not usually dependent on the project approach. While it is commendable that the team is engaging in continuous improvement, such efforts are not considered one of the success criteria tor a useful methodology.

12.Answer C

Explanation: When adopting a new agile practice, we should always investigate the claims about the practice ourselves to try and validate whether they are real. We should then try it out in small doses, so the impacts are contained if it does not work, and we should inspect and review the findings early to make sure the new approach is actually working and not causing more problems than it is solving. Trying out an approach on local projects might help us see the impacts better, but if our local projects are very large, then this could be a bad move. Likewise, developing new approaches ourselves is risky and time-consuming.

13.Answer B

Explanation: Lessons learned should be captured throughout the project when the information is still fresh and people remember the most details. This allows the lessons to be used in the remainder of the project.

14.Answer A

Explanation: Information exchanges in agile methods are designed to facilitate knowledge sharing. We deliberately do not rely on leveraging electronic tools; instead, we focus on face-to-face communications. The goal is not to increase resource utilization or generate stable specifications either.

15.Answer: D

Explanation: Agile methods should be tried as-is first before considering modifications such as tailoring. We should first understand how they work before attempting to change them. If we change them first and then encounter problems, how will we know if the problems are related to tailoring method or genuine project issues?