

Olga Slavkova, Jaume Grané

## To CASE, or not to CASE?

### 1. Introducción

*Los sistemas CASE actuales confieren a un analista de sistemas lo que un procesador de textos a una secretaria. El dilema se plantea a la hora de elegir: ¿hasta dónde debo llegar? ¿qué herramienta necesito?*

CASE, "Computer Aided Software Engineering", aparece como una solución en el mundo cambiante del desarrollo del software. Pero, ¿es realmente la solución, o es una moda pasajera? Para intentar dar respuesta a esta pregunta existen muchos estudios y muchos casos prácticos que demuestran la efectividad del par CASE-Metodología. Existe también una experiencia consolidada que demuestra la efectividad de los conceptos modernos de la Ingeniería del Software de forma independiente al CASE. Sin embargo, todavía no se han asentado estos conceptos de forma universal en la industria del software.

Siendo así, ¿qué papel juega el CASE en todo esto? Actualmente, el mercado presiona como nunca a la industria del software demandando incipientemente más metodología, más calidad, más efectividad, más rendimiento y, sobre todo, menos coste, menos coste (insisten) y menos errores. La industria se encuentra entonces atezada con la necesidad de modernizarse, con la necesidad de convencer a los clientes, con la necesidad de cambiar. En estas condiciones, los analistas de sistemas y las empresas en general que se dedican al desarrollo del software se plantean seriamente la necesidad de profundizar en estos temas, con la sana intención de mejorar el proceso productivo según la demanda del mercado. En esta espiral de cambio, se llega inevitablemente a un punto donde uno debe estudiar la posibilidad de adquirir una herramienta CASE. Esto puede producir más de un dolor de cabeza, y de bolsillo.

Este artículo analiza algunos de los conceptos importantes a tener en cuenta a la hora de seleccionar una herramienta CASE. Previamente, proporciona una pequeña introducción al mundo del CASE para establecer el contexto necesario a las personas no familiarizadas con él. El artículo se centra en los criterios de evaluación, por tanto, se dirige a todas aquellas personas que quieren adquirir una herramienta CASE o que actualmente están usando una y quieren ser críticos con ella. También, claro está, sentirán interés por nuestra exposición aquellas personas que creen que el CASE es una moda pasajera. Sea entonces.

### 2. ¿Qué es una herramienta CASE?

Las herramientas CASE, como su nombre bien indica, proporcionan asistencia automatizada a la ingeniería del software.

Esto es, proporcionan los medios necesarios para automatizar y simplificar, a través del ordenador, las labores manuales y tediosas que se realizan a lo largo del ciclo de vida del software, así como la planificación y la gestión de los proyectos que se realizan en torno a este ciclo de vida. Algo que no hacen las herramientas CASE es pensar por el analista, el diseñador o el gestor; pese a ello, pueden descargarlo de las labores de verificación sintáctica empleada en las técnicas soportadas.

Llegados a este punto, es importante prestar atención a los conceptos *herramienta*, *técnica*, y *metodología*. Una **metodología** comprende diferentes aspectos, pero podemos pensar que representa un conjunto de pasos a realizar para llevar a cabo la ingeniería del software. Las **metodologías** establecen las **técnicas** a emplear para cada uno de los pasos, de manera que quedan fijados perfectamente cuáles son los productos de cada paso y como deben producirse. Estas técnicas representan una forma de hacer, con un objetivo claro al que se llega a través del uso de **diferentes herramientas**. Sea, por ejemplo, el caso de las técnicas de diagramación de flujos de datos (**DFD**). La definición de un **DFD** es una técnica que usa el analista con una intención concreta. En este caso se trata de la representación de su idea acerca de ciertos requerimientos del sistema en un formato no tan ambiguo como el texto, más visual y más compacto. El **DFD** forma parte de varias metodologías, entre las cuales se encuentra el análisis estructurado. Éste, a su vez, propone el uso de otras técnicas. La metodología establece el nivel superior, proponiendo cómo y cuando debe uno usar una técnica y que debe uno intentar hacer con ella. Las herramientas automatizadas, las herramientas CASE, proporcionan ayuda a la ejecución de las técnicas y pueden o no soportar metodologías completas.

#### 2.1. Clasificación de las herramientas CASE

Clasificar una herramienta CASE no es una labor fácil. En principio, hemos de suponer que las herramientas pueden cubrir más de un campo dentro de la ingeniería del software. Esto puede condicionar una clasificación concreta, puesto que unas personas pueden encuadrar una herramienta en una de las clases y otras pueden encuadrarla en otra.

Una clasificación basada en la funcionalidad de la herramienta [PRE93] se resume en el **cuadro 1**. Otra podría basarse en la estructura interna de la herramienta y su nivel de integración en otros entornos CASE, como se indica en la **figura 1** [PRE93]. Por último, podríamos clasificar las herramientas CASE en función de la plataforma sobre la que corren (su entorno de soporte). Por ejemplo, plataformas DOS puras, DOS-Windows, UNIX, sistemas propietarios IBM, DEC, Apple, etc.

- A. Herramientas de planificación organizativa del entorno de los sistemas**
- B. Herramientas de gestión de proyectos**
- de planificación de proyectos
  - de seguimiento de requisitos
  - de gestión y métrica
- C. Herramientas de soporte**
- de documentación
  - de utilidades para sistema
  - de control de calidad
  - de bases de datos y de gestión de la configuración
- D. Herramientas de análisis y diseño**
- de análisis y diseño estructurado
  - de prototipado y simulación
  - para el diseño y desarrollo de interfaces
  - herramientas programables de análisis y diseño
- E. Herramientas de programación**
- de codificación convencional
  - de codificación de cuarta generación
  - de programación orientada a los objetos
- F. Herramientas de integración y prueba**
- de análisis estático
  - de análisis dinámico
  - de gestión de pruebas
- G. Herramientas para la creación de prototipos**
- H. Herramientas de mantenimiento**
- de ingeniería inversa
  - de reingeniería
- I. Herramientas para la integración estructural de otras herramientas**

Cuadro 1: Clasificación CASE basada en la función de la herramienta

Podríamos ir un poco más lejos: ¿por qué no tomar el conjunto de estas clases? Evidentemente, ello conduce a una división del dominio de las herramientas CASE en al menos 3 dimensiones: **función, organización interna y entorno de soporte**. Las herramientas CASE del mercado suelen encajar en un estrecho margen dentro de cada dimensión, de manera que su alcance queda restringido a entornos hardware concretos, a una parte concreta de la ingeniería del software o a un nivel de integración concreto. Este enfoque sobre las herramientas actuales determina uno de los factores a tener en cuenta para su elección.

### 3. ¿Hasta dónde se puede llegar?

Establecer el alcance de aplicación de las herramientas CASE modernas no es tarea fácil. Para ello, proponemos considerar algunos de los criterios que caracterizan genéricamente a una herramienta CASE. De esta manera, el lector podrá formarse una idea de hasta dónde puede llegar una herramienta concreta, a partir de sus características. Una herramienta CASE, como cualquier otro sistema de información basado en ordenadores, presenta una componente lógica y una componente física o implementación. Si analizamos las herramientas CASE siguiendo este criterio, podemos advertir que en el **plano lógico**, la herramienta puede presentar las siguientes características:

- Potencia del repositorio.
- Potencial analítico.
- Soporte metodológico.
- Capacidad de generación de código.
- Soporte de técnicas diversas; estructuradas y no estructuradas.

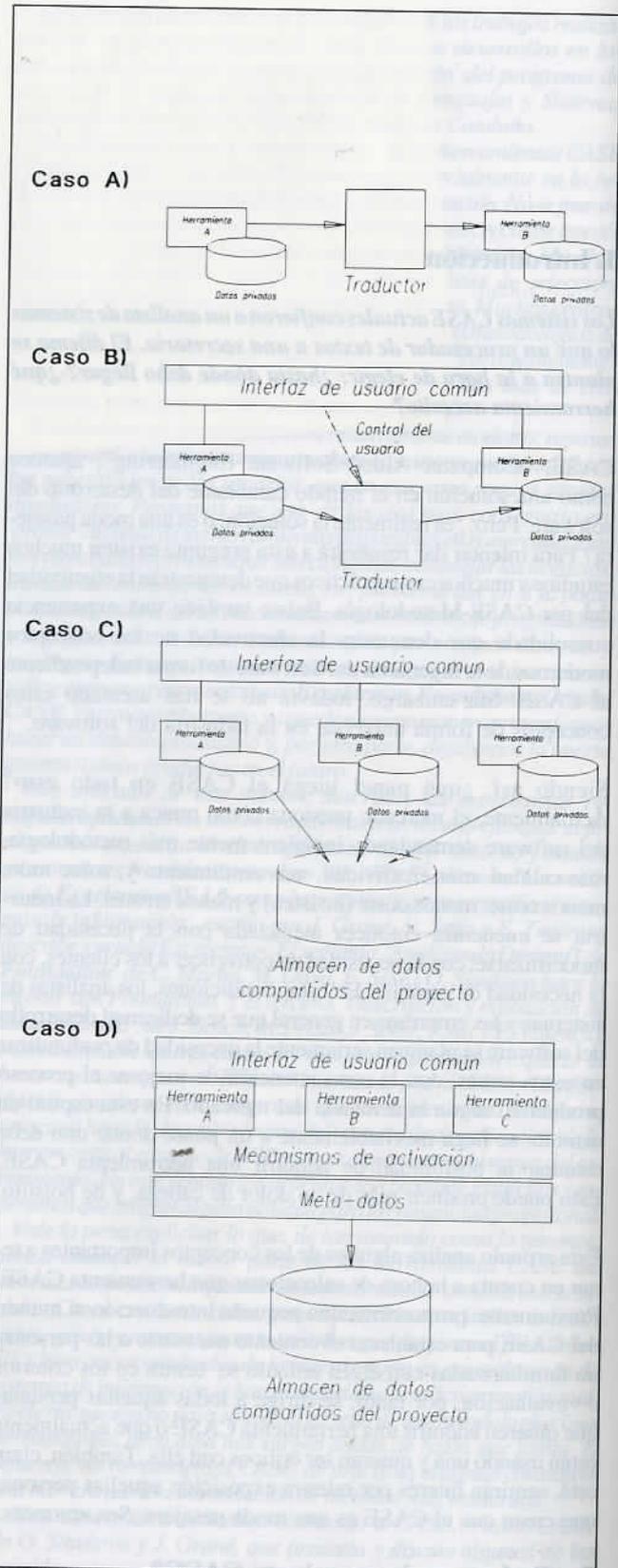


Figura 1: Cuatro niveles de Integración CASE:

- a) intercambio de datos;
- b) acceso común a herramientas;
- c) integración de datos;
- d) integración total

- Reingeniería.
- Integración del control de la calidad.

En cuanto al **plano físico** podemos considerar lo siguiente:

- Ergonomía.
- Rendimiento.
- Consumo de recursos del sistema.
- Compatibilidad con otros entornos.
- Portabilidad.
- Concurrencia.

Extendiendo estos conceptos, en [MIS 90] las facilidades de una herramienta CASE se clasifican en cuatro categorías:

- Soporte para la metodología
- Componentes de un sistema CASE y facilidad de su manejo
- Facilidad de introducción de la herramienta en la empresa
- Soporte técnico de la herramienta después de adquirida

### 3.1. Soporte para la metodología

Asumiendo que la organización ya ha adoptado una metodología concreta, el responsable de gestión de sistemas tiene que considerar que la metodología implementada por el CASE corresponde a la ya implementada en el entorno de la organización. En caso contrario (por ejemplo cuando se introduce simultáneamente un entorno de trabajo nuevo y una metodología nueva) se corre el riesgo de fallar debido a reacciones negativas por la parte de los analistas y diseñadores. Además, la falta de experiencia no permitiría detectar a tiempo los errores de ajuste entre el nuevo sistema CASE, la nueva metodología y el entorno de la organización.

### 3.2. Componentes de un CASE y facilidad de su manejo

El responsable de seleccionar la herramienta CASE debería verificar, tanto la existencia de los componentes que necesita la organización, como su facilidad de aprendizaje y manejo. En este sentido, podemos atender a los siguientes criterios:

#### 3.2.1 Primitivas predefinidas de imágenes

La construcción de los diagramas se facilita considerablemente por la existencia de un conjunto adecuado de primitivas predefinidas de imágenes (por ej, circuitos, rectángulos, rectas, flechas y otros símbolos gráficos de la metodología correspondiente). Por otra parte el usuario debe disponer de facilidades para definir sus propias primitivas y almacenarlas para su uso posterior. Otra facilidad en el caso de sistemas en tiempo real e iterativos sería el control sobre los procesos y flujos de datos.

#### 3.2.2 Creación de diagramas

Una primera medida del esfuerzo requerido para la creación de un diagrama es el número de teclas claves necesarias para colocar un objeto gráfico en la pantalla y la existencia del cor-

respondiente menú. La segunda medida está relacionada con la facilidad de controlar la localización de los nombres en el diagrama y de asociar nombres con los objetos.

#### 3.2.3 Edición de los diagramas

En este caso los criterios a considerar son los siguientes:

- el esfuerzo necesario para eliminar, reponer o modificar el objeto gráfico;
- la prevención de la inconsistencia del modelo como consecuencia de la eliminación o relocalización de un objeto;
- la relocalización de uno o más objetos dentro del mismo diagrama (sin tener que eliminarlos);
- la relocalización de uno o más objetos en diferentes diagramas;
- el número de diagramas que puede relocalizarse simultáneamente;
- el esfuerzo necesario para renombrar un objeto gráfico (el mecanismo del manejo y control que puede ejecutar el usuario sobre él);
- la ampliación o reducción independiente de un objeto gráfico o de un nombre;
- la visualización de diagramas más grandes que la pantalla de trabajo (posibilidad de rotación, 'zooms' positivos y negativos);
- el soporte de ventanas para la visualización de los diagramas y su edición (el número máximo de ventanas activadas simultáneamente, la independencia de los diagramas y su edición en cada una de las ventanas);
- la creación de diagramas en páginas;
- la visualización del diagrama (posiblemente en páginas) antes de su impresión.

#### 3.2.4 Interfaz de entrada

El soporte de ratón u otros interfaces gráficos se considerarán ventajosos para el sistema CASE al ser la modelización gráfica.

#### 3.2.5 Interfaz de salida

Debemos considerar:

- facilidades de impresión y/o plotting tanto *offline* como independientemente de otros trabajos de análisis o diseño;
- formatos de la impresión (fondos y tamaños de los caracteres, selección de las páginas a imprimir, configuración de la página).

#### 3.2.6 Soporte e integración del diccionario

La falta de diccionario integrado en el software no permitiría su clasificación como sistema CASE. Los objetos se catalogan automáticamente en el momento de su creación, o la integración se realiza mediante el uso del diccionario de las interrelaciones entre los objetos de un modelo. Sobre el soporte del diccionario del sistema se valorarán los siguientes criterios:

- la extensión del soporte del diccionario sobre las diferentes fases del proceso;
- el tamaño del diccionario (i.e., la capacidad de catalogar objetos o interrelaciones);

- el esquema de clasificación empleado por el diccionario (se valora la clasificación automatizada y la agrupación según un orden predefinido de varios objetos tales como elementos informativos, flujos de datos, entidades externas, almacenes de información, procesos y módulos del diseño del programa);
- la navegación dentro del diccionario (se refiere al acceso a las entradas desde cualquier punto del sistema);
- la existencia de estándares de documentación en la organización y minimización de los problemas de adopción del CASE;
- la definición de estructuras de datos complejas y las interrelaciones entre diferentes elementos de estas estructuras;
- el mantenimiento de las interrelaciones jerárquicas entre los procesos o módulos de programas;
- la facilidad de identificar los 'items' aislados de información, los procesos, módulos y datos o estructuras no definidos, y la duplicación de la definición de la misma información;
- interrelaciones cruzadas entre ítems de información y procesos.

### 3.2.7 Análisis automático y generación de listados

El análisis depende de la metodología particular que soporta cada herramienta CASE. Incluye la verificación por jerarquía de los DFD, los procesos y los módulos así como la generación de listados con el resultado de dicho análisis.

### 3.2.8 'Layout' de la pantalla

Si existe esta facilidad, se pueden evaluar los tres criterios:

- uso de diagramas en el 'layout';
- nivel de síntesis entre los gráficos y el texto;
- integración del diccionario con los 'layouts'.

### 3.2.9 Otras consideraciones

- Las facilidades de integración y coordinación entre las diferentes tareas desarrolladas por los miembros del colectivo;
- la interfaz con bases de datos externas en el caso de empresas que han estandarizado los *items* informativos y necesitan continuar su uso sin integrarlos en el entorno del CASE;
- la reacción del sistema ante los errores y el nivel de dificultad de su recuperación por el usuario;
- el aseguramiento de la consistencia del modelo y/o de los mensajes generados;
- la tolerancia a cierta cantidad de fallos en el uso;
- la calidad de la ayuda 'on-line' del sistema (acceso desde cualquier punto del sistema, uso rutinario de los menús y los 'prompts', claridad de los mensajes del sistema, tutoriales 'on-line' y proyectos demostrativos);
- el tiempo de respuesta del sistema;
- la ayuda para gestión de la configuración (coordinación entre los *items* de información y la especificación de requerimientos);
- la generación automatizada del código desde las especificaciones del diseño;
- el tipo del lenguaje de especificación desde el cual es posible la generación del código.

### 3.3. Facilidad de introducción de la herramienta en la empresa

En este caso deben considerarse (por orden de su prioridad) los siguientes factores:

- **el nivel del asesoramiento de aprendizaje** (tutoriales y proyectos de demostración ejecutables por pasos; tiempo mínimo hasta ser aptos para realizar tareas simples; necesidad de memorizar el uso rutinario; facilidad de uso de la documentación del sistema; facilidad de memorización de los comandos mnemónicos y las teclas especiales);
- **la compatibilidad de hardware con el entorno de la herramienta CASE** (compatibilidad entre los procesadores, impresoras, plotters y sistemas operativos existentes en la organización con el entorno CASE; necesidad de complementos especiales como ratón, extensión de la memoria, coprocesador matemático, adaptadores gráficos, digitalizador);
- **las facilidades de aprendizaje del manejo de los componentes del sistema** (cursos de entrenamiento y su adaptación a las necesidades de la empresa);
- **la gestión operativa** (asesoramiento por especialistas de la instalación del nuevo sistema, su mantenimiento y su uso).

### 3.4. Soporte técnico de la herramienta después de adquirida

Antes de la adaptación del CASE, la empresa debe examinar el tiempo de respuesta y la eficacia de la 'hot line' del vendedor.

## 4. ¿Qué herramientas son necesarias?

Si admitimos que la razón principal para la introducción de una herramienta CASE en una organización es la necesidad de aumentar la calidad y la productividad, deben considerarse las necesidades que la herramienta CASE ha de cubrir en relación al proceso de control de la calidad y al control de la producción que se realiza en la organización.

Como estos procesos no son homogéneos en el conjunto de organizaciones, las necesidades efectivas dependen también del tipo de organización.

### 4.1. El control de la calidad y la producción del software

Actualmente, el control de la calidad en la producción del software consiste en la obtención de un modelo efectivo del proceso de 'fabricación' del software, con el propósito de realizar una medición cuantitativa de sus atributos.

La medición cuantitativa se realiza con las **métricas** del software. Existe una estrecha relación entre los modelos y las métricas. Los atributos medibles del modelo del software se cuantifican por sus correspondientes medidas y en base a estas mediciones se calcula el resultado de las métricas (p.ej. se evalúa el software). Dependiendo del tipo de métrica, este resultado puede ser una evaluación del producto o de la producción del software. En el entorno CASE existe potencial para automatizar la medición y el control de la calidad. Esto permite una gran

precisión de las mediciones con gastos relativamente bajos. Para cada CASE hay que determinar los atributos de la calidad y los tipos de métricas para su control [TAT91].

Entre las **métricas de proceso** de mayor utilidad en el caso del CASE se tienen las relacionadas con la medición de consumo de recursos por el proceso de software. El recurso cuyo consumo más influye en el precio del software es el esfuerzo realizado por el personal informático. Se puede medir automáticamente la duración de las sesiones de trabajo, la ocurrencia y duración de los períodos no activos, la cantidad de llaves usadas, la cantidad y los tipos de transacciones realizadas por CASE, la cantidad y el tipo de los repositorios modificados, etc. La unidad de medida estándar para el esfuerzo es horas-trabajo.

También puede aplicarse la medida de **Halstead** [FEN91]:

$$\text{Esfuerzo} = \text{Volumen de trabajo} \times \text{Dificultad de realizar una unidad del trabajo}$$

En general, los resultados de las métricas del proceso evalúan la relación entre las actividades del trabajo con sus efectos sobre las aplicaciones desarrolladas en el entorno CASE.

Las **métricas de productos** más útiles en el caso de las herramientas CASE son las que evalúan el tamaño del producto. Para la medición del tamaño del producto desarrollado en el entorno CASE (modelos de datos, diagrama de flujo, especificaciones de los listados, esquemas y código generado, etc.) puede aplicarse la métrica de los puntos funcionales de **Albrecht** [FEN91]. En particular puede medirse el tamaño:

- a) del modelo de datos general y del detallado;
- b) del modelo de flujo de datos general y del detallado;
- c) de la interfaz con el usuario;
- d) de la especificación funcional;
- e) del diseño (depende de la estructura del proceso de diseño);
- f) del código generado (depende del nivel del lenguaje de programación).

La métrica de tamaño puede ser útil para evaluar la productividad de las tecnologías del CASE y del personal que usa estas tecnologías mediante la fórmula:

$$\text{Productividad} = \text{Dimensión del producto} / \text{Tiempo empleado}$$

La necesidad de evaluar el nivel de uso por el personal de la herramienta se justifica por su clara influencia sobre la productividad, el coste y la calidad de las aplicaciones. La experiencia del personal también influye. Un conjunto de medidas de la experiencia personal con tipos concretos de proyectos, sistemas, herramientas, métodos y lenguajes de programación está incorporado en el modelo **COCOMO** de **Boehm** [FEN91].

Todos los detalles importantes del proceso se encuentran en los repositorios; por tanto, la evaluación puede reducirse, en el ca-

so del CASE, al contenido de los repositorios, contando las estructuras y los objetos específicos para el modelo de la medición. Esto facilitaría la aplicación automatizada de las métricas para el control de calidad del proceso de software y del producto de software.

#### 4.2. Relación entre control y nivel de madurez de la organización

El control que ejercen los directivos de la organización sobre el proceso de fabricación de software en cada una de sus fases es el criterio para la clasificación de las organizaciones según su nivel de madurez. La diferencia entre un nivel y otro se debe a las diferencias de visibilidad sobre el proceso y su control. La integración de la herramienta CASE en el entorno de la organización permite aumentar su nivel de madurez. El nivel de la integración se puede evaluar efectivamente con una colección de métricas automatizadas apropiadas al entorno de la organización.

En [PFL91] se definen 5 niveles de madurez del proceso y sus características según el modelo de Capacidad de Madurez de Software del Instituto de Ingeniería de Software de EEUU. En [CER93] se dan los criterios precisos para definir el nivel de madurez de una organización:

##### 4.2.1 NIVEL 1: proceso inicial

El primer nivel se caracteriza por su aproximación "ad hoc" al proceso de desarrollo del software (p.ej., falta de estructuración adecuada; falta de control desde la entrada al proceso hasta la salida del producto).

El criterio para seleccionar la herramienta CASE a implantar en el entorno de este nivel sería la que permitiera estructurar el proceso y definir sus entradas y salidas. Para la evaluación del efecto de uso de la herramienta, puede aplicarse una colección de métricas básicas de software (tales como métricas de esfuerzo y de tamaño del producto).

##### 4.2.2 NIVEL 2: proceso repetible

Este proceso tiene definidas las restricciones a sus entradas y salidas. El sentido del término repetible es equivalente al de la 'caja negra': las entradas adecuadas producen salidas esperadas, pero sin tener visibilidad sobre el proceso de la producción. Las herramientas CASE apropiadas para este nivel son las que estructuran y controlan la entrada para la especificación de los requerimientos, su especificación y su análisis. Por depender este proceso de factores subjetivos, las herramientas CASE recomendables son las que capturan la información y la hacen visible y accesible para el resto del grupo.

Las restricciones del proyecto pueden seguirse durante el proceso con herramientas de gestión de proyectos. La salida puede controlarse mediante sistemas de ayuda para la configuración del proceso. Paralelamente, herramientas que producen documentación harán visible los productos que examinarán procesos de mayor nivel de madurez.

En este nivel las métricas son aplicables para evaluar la configuración del proyecto (p. ej. la métrica de **Henry-Kafura** [FEN91], que se basa en el flujo de información que debe entrar y salir del sistema de información).

#### 4.2.3 NIVEL 3: proceso definido

Las actividades funcionales del proceso en este nivel están claramente definidas por sus especificaciones estrictas de entrada y salida. Esto permite controlar los atributos del proceso con las métricas de complejidad (por ejemplo, la complejidad ciclomática de McCabe) y de calidad (generalmente forman parte de modelos FCM, '**Factor Criteria Metric model**', el más conocido de los cuales es el modelo de Boehm).

Es apropiado el uso de las herramientas CASE para el análisis, la estructuración y el asesoramiento del proceso en cada una de sus actividades intermedias. Además, pueden ser útiles las herramientas de reingeniería cuando partes de sistemas ya creados se incluyen en un sistema nuevo y se necesita comprobar su integración y/o crear las partes que faltan para aquélla. El efecto del 'reuso' sobre la productividad en una organización se ha estudiado en [BAN91].

#### 4.3.4 NIVEL 4: proceso dirigido

En este nivel, además del análisis estático como en los niveles anteriores, es posible el análisis dinámico del proceso. Pueden controlarse sus actividades y seguirse los efectos de los cambios en una de las actividades dentro de las otras. Aquí el efecto '**feedback**' de proyectos procesados anteriormente se usa para revisar las especificaciones de la entrada y la salida, y para establecer prioridades entre las diferentes actividades del proceso. El enfoque del uso de las herramientas CASE está en el análisis del proceso en lugar del análisis del producto.

Las métricas son aplicables en los bucles '**feedback**' para recolectar la información, por versiones específicas del producto, sobre la cantidad de defectos del diseño, la cantidad y los tipos de problemas representados. Para efectuar un mejor control sobre todas las actividades del proceso se aplica una colección de métricas relacionadas con el proceso entero para la evaluación de sus atributos y la interacción entre sus actividades.

#### 4.3.5 NIVEL 5: proceso optimizado

Este nivel se caracteriza por el modelo dinámico del proceso, que puede ser cambiado y optimizado antes de su finalización, como resultado del '**feedback**' desde actividades anteriores. Si tras cada actividad, el resultado de la medición de la salida del proceso es negativo, hay la opción de revisarlo o rehacerlo.

Las métricas para este Nivel 5 realizan un control dinámico sobre el proceso. Las herramientas CASE apropiadas para este nivel son las que dinámicamente evalúan, configuran o rehacen el proceso (por ejemplo, sistemas de simulación o de programación de procesos).

**En resumen**, las herramientas CASE facilitan notablemente la labor de los ingenieros de software, pero a su vez su diversidad dificulta la selección de la más apropiada: elección que se complica por la falta de clasificación de las posibles combinaciones hardware/software y de normas para los sistemas CASE. También influye el coste y el tiempo que consume la evaluación de una herramienta de este tipo. Para que sea efectiva la selección, tienen que considerarse y/o compararse diversas facilidades de las herramientas.

## 5. Consideraciones adicionales en la selección

### 5.1. La subjetividad de la elección de una herramienta CASE

Elegir algo de entre un conjunto de posibilidades parecidas no siempre viene determinado por criterios objetivos. Muchas veces, las personas tendemos a elegir las cosas atendiendo a las ideas preconcebidas que nos formamos acerca de ellas. ¿Acaso Usted no valoraría más una herramienta CASE concreta que se pareciera a los programas que está acostumbrado a manejar su equipo de analistas? ¿No valoraría más una herramienta CASE que soportara la metodología que usa cada día? Preguntándolo al revés: ¿le interesaría una herramienta CASE que no tuviera nada que ver con el interfaz de usuario que está acostumbrado a usar; o bien un CASE que no soportara la metodología que emplea habitualmente? Seguramente las respuestas a estas preguntas son independientes de la calidad del producto (CASE) que se esté analizando, al menos en cierto grado.

Este grado de ligazón entre las apreciaciones subjetivas del usuario y el potencial objetivo de los productos determina que cada uno de los parámetros de evaluación tenga un peso final en la selección de la herramienta CASE. Estos pesos dependen de cada persona.

### 5.2. Relación coste-beneficio de una herramienta CASE

De forma independiente al coste de la herramienta y a la plataforma tecnológica asociada que se esté considerando, se deben tener en cuenta ciertos factores que pueden introducir un sesgo importante en la amortización de la inversión en CASE.

Como toda herramienta, un sistema CASE exige un cierto grado de preparación para su uso. Esta preparación se consigue a través de la experiencia y el entrenamiento. La curva de aprendizaje, como medida de la evolución de este entrenamiento, determina un coste adicional asociado a cada tipo de herramienta. Este coste, nada despreciable, corresponde al esfuerzo adicional por parte de la organización en cuanto a la dedicación de recursos humanos y materiales. Sobre este concepto podemos distinguir lo siguiente:

- costes derivados del **esfuerzo necesario** para la adaptación de los ingenieros del software a las nuevas plataformas de trabajo (valga como ejemplo la familiarización con la herramienta en sí, los sistemas operativos, etc.);

- costes derivados de la **implantación de técnicas** concretas asociadas con la herramienta en la organización (costes que pueden ser generados tanto por los ingenieros del software como por la parte directiva de la empresa u organización);
- costes derivados de las **actividades de formación** complementarias, incluyendo recursos humanos y materiales.

Adicionalmente, deben considerarse los costes derivados de la experimentación que suele producirse de hecho en las primeras etapas de la implantación de una herramienta CASE. En muchas organizaciones se plantea iniciar un proyecto, llámese piloto, para adquirir experiencia en la herramienta (en general se hace con toda metodología) por aquéllo de "sólo se aprende con casos reales". Esta técnica, si bien forma al personal en el 'campo de batalla', supone un coste que puede hacer tambalear el presupuesto y la planificación del proyecto.

## 6. Conclusiones

En este artículo se han presentado algunos de los factores a considerar dentro del mundo de las herramientas CASE, en el marco de su evaluación o selección. Quizás algún lector esperaría haber encontrado una disertación, con un talante estadístico, acerca de qué herramienta es la mejor de las disponibles actualmente en el mercado. Es cierto que es posible realizar un estudio estadístico comparativo basado en una determinación previa de las características cuantitativas de las herramientas, pero no se debe nunca olvidar el factor humano. Podemos intentar determinar la herramienta más potente, la herramienta más sencilla de manejar, la herramienta que más guíe al usuario, la herramienta que más ... ¿Es esto lo que realmente nos interesa?

En sentido estricto, se han proporcionado algunos de los criterios más importantes que determinan una clasificación y cuantificación de las características de las herramientas CASE y de los usuarios (organizaciones) que deben manejarlas. En esta línea, se puede concluir que el evaluador o seleccionador de una herramienta CASE debe interpretar los valores absolutos de la herramienta, desde el punto de vista físico o lógico, para decidir, desde su punto de vista subjetivo, la idoneidad de la herramienta. Tomando los diferentes parámetros expuestos aquí, se intuiría que cada persona podría diferenciar grupos afines dentro del campo de las herramientas CASE y dentro del campo de los usuarios. En estas condiciones, podríamos ver una evaluación concreta como una correspondencia entre estos grupos de afinidad. En la **figura 2** se indica genéricamente cómo un grupo de herramientas puede ser idónea para un grupo de usuarios, habiendo diferenciado los grupos por afinidades. La idea que subyace en la figura nos lleva a considerar la elección concreta de una herramienta a partir de la agrupación subjetiva de las siguientes consideraciones sobre el análisis de los tres tipos de aspectos de la herramienta:

- **lógicos** (su función),
- **físicos** (su implementación),
- **del usuario final** (la organización).

Todos estos factores deben considerarse, sin olvidar en ningún momento los parámetros de índole económica, para establecer el nexo ideal entre las necesidades del usuario y las características de la herramienta. Debemos subrayar que la valoración subjetiva del usuario es también un factor a tener en cuenta. Por último nos gustaría hacer una analogía entre el proceso de elegir una herramienta, al que esperamos haber dado algunas bases, y los típicos juguetes donde el niño debe encajar una pieza geométrica en un conjunto de huecos también geométricos. En este caso, el niño debe acertar qué pieza encaja con un hueco determinado. Algunos niños se percatan al instante de la pieza correcta por simple observación. Otros, sin embargo, intentan forzar la pieza que, al final, encaja a duras penas. Algunas piezas son claramente mayores que el hueco donde se quieren encajar; otras son mucho menores, aunque pueden encajar en algunos de sus ángulos. Lo mismo ocurre con las herramientas CASE.

## 7. Referencias

- [PFL91] S.L. Pflieger. *Process maturity as framework for CASE tool selection*. Information and Software Technology, vol.33, #9, november 1991, pp. 610 - 615.
- [CER93] J. Cervera, L. Fernandez, T.S. Felic. *Selection of CMM map CASE TOOLS: New Approach Based on the Capability Maturity Model (CMM)*. CIL'93 (Barcelona, 20-22 abril 1993).
- [DAT92] Datapro. *How to Select a CASE Tool*. Corporate Software & Solutions, november 1992, pp. 1-7.
- [MIS90] S.K.Misra. *Analysing CASE system characteristics: evaluative framework*. Information and Software Technology, vol.32, No.6, july/august 1990, pp. 415-422.
- [TAT91] G. Tate. J. Verner. *Software Metrics for CASE Development*. IEEE, 1991, pp. 565 - 570.
- [BAN91] R.D.Banker, R.J.Kauffman. *Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study*. MIS Quarterly, september 1991, pp.375-401.
- [PRE93] Roger S. Pressman. *Ingeniería del software, un enfoque práctico*. 3ª edición. McGraw-Hill 1993.
- [FEN91] Fenton N. E. *Software metrics: a rigorous approach*. Norwich Ltd. Great Britain, 1991.

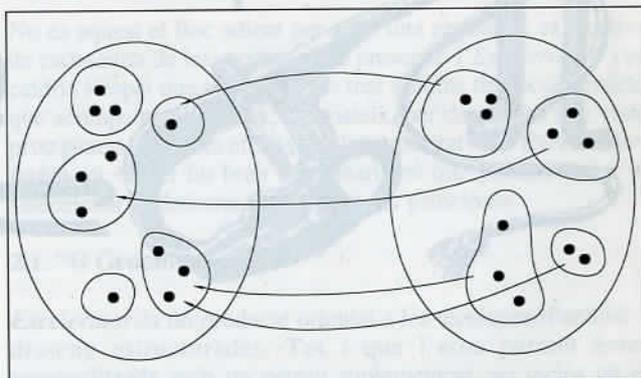


Figura 2: Correspondencia entre herramientas CASE y usuarios afines