

# Estimación y priorización de requisitos no-funcionales para desarrollo de software: Estado del arte

María José Salamea<sup>1</sup>  
Liliana González-Palacio<sup>2</sup>  
Marc Oriol<sup>1</sup>  
Carles Farré<sup>1</sup>

<sup>1</sup>Universitat Politècnica de Catalunya, España

<sup>2</sup>Universidad de Medellín, Colombia

Los requisitos de calidad (también llamados requisitos no-funcionales) son aquellos que permiten asegurar la calidad del software. Incluyen aspectos muy diversos, como disponibilidad, seguridad, rendimiento, escalabilidad, portabilidad y usabilidad, entre otros. Los continuos avances tecnológicos, como el software en la nube o el Internet de las Cosas, presentan nuevos retos en el desarrollo del software para poder garantizar un nivel de calidad satisfactorio de dichos aspectos. Asimismo, las metodologías de desarrollo ágil, cuyo uso viene en aumento, tales como SCRUM, XP, Kanban, no dan el soporte necesario para la gestión de dichos requisitos de calidad. Con el fin de facilitar a los ingenieros del software la toma de decisiones sobre el nivel de calidad necesario en un proyecto, es imprescindible conocer de antemano: 1) qué criterios se van a tener en cuenta para verificar, priorizar, planificar y/o negociar los requisitos de calidad. Asimismo, es necesario: 2) precisar cómo se van a evaluar dichos criterios, y 3) identificar qué factores del contexto del proyecto pueden afectar dicha evaluación. Para intentar dar respuesta a estas 3 cuestiones o preguntas de investigación, los autores de este capítulo han diseñado y están llevando a cabo un estudio sistemático de la literatura. Este trabajo presenta para su discusión la descripción de la metodología seguida en ese estudio, así como algunos de los resultados preliminares obtenidos durante su ejecución.

## 1. INTRODUCCIÓN

La calidad del software es un elemento fundamental para garantizar el éxito de un proyecto software, así como un factor competitivo esencial en las empresas que lo desarrollan [1]. El estándar ISO/IEC 8402:1994 define la calidad del software como *la totalidad de características de un producto o servicio que le confieren su aptitud para satisfacer necesidades establecidas o implícitas* [2]. Dicha calidad engloba diversas características (p.e. seguridad, disponibilidad, rendimiento, etc.) y dichas características suelen estar organizadas bajo la estructura de modelos de calidad, definidos a su vez como la especificación de las características requeridas que un sistema software debe exhibir [3].

Los estudios de mercado muestran un incremento continuado de la proporción de presupuesto destinado a tratar con la calidad del software. Por ejemplo, un informe de la compañía Capgemini reportó que el gasto en el aseguramiento de la calidad y las pruebas de software en compañías tecnológicas, subió del 18% en 2012 al 26% en 2017 [4]. Este incremento se ve justificado debido al impacto que tiene la calidad del software sobre los negocios de las compañías que lo utilizan en sus productos y servicios. Son recurrentes las pérdidas económicas por software defectuoso o de mala calidad. Por ejemplo, la compañía de software Tricentis identificó y analizó 606 casos de graves fallos en software durante el 2017 y estimó que causaron más de 1.7 billones de dólares (USD) en pérdidas [5].

Para mejorar y asegurar la calidad, algunas aproximaciones argumentan que es necesario tratar dicha calidad explícitamente desde el principio mediante su formalización en requisitos [6]. Los requisitos de calidad (también llamados requisitos no-funcionales) son características, condiciones o restricciones con los que los ingenieros del software garantizan el correcto funcionamiento de un aplicativo. Los requisitos de calidad incluyen aspectos muy diversos, tales como disponibilidad, seguridad, rendimiento, escalabilidad, portabilidad, etc. Los nuevos métodos de desarrollo ágil de software (p.e. XP, Scrum, Kanban) abren paso a nuevos retos para poder garantizar un nivel de calidad satisfactorio de dichos aspectos [7].

Entre los distintos retos que existen para considerar y definir los requisitos de calidad en los entornos de desarrollo ágil, está la dificultad de poder estimar el esfuerzo de desarrollo necesario para alcanzar determinados niveles de calidad, así como la priorización de los requisitos de calidad. Ambas actividades, estimación del esfuerzo de desarrollo y priorización, son imprescindibles durante la fase de ingeniería de requisitos para llevar a cabo una verificación más precisa, una negociación mejor informada y una planificación más apropiada de los requisitos de calidad de un proyecto software.

Por un lado, la estimación del esfuerzo de proyectos de software es el proceso de predecir el esfuerzo requerido para desarrollar o mantener un sistema de software [8]. La falla o cambio en una estimación puede afectar al proceso de implantación, y esto puede generar un impacto en el costo del mismo [9]. Por ejemplo, puede conllevar a una mala asignación de recursos, retrasos y sobrecostos respecto a la planificación inicial [10].

De otro lado, la priorización de requisitos es la determinación de cuáles, entre un grupo de requisitos, se implementarán primero y el orden de dicha implementación [1, 11]. La priorización de qué requisitos deben ser incluidos en cada iteración del producto es un complejo proceso de decisión multicriterio que suele implicar llegar al

equilibrio entre el beneficio para el negocio de cada requisito y el consumo de recursos. Dicha problemática se presenta cuando las necesidades de los clientes deben ajustarse a la capacidad de trabajo del equipo de desarrollo [12]. Ambas actividades están muy ligadas. Por ejemplo, la estimación del esfuerzo para implementar un requisito puede ser un factor determinante en su priorización, asimismo, una mala priorización puede conllevar a un mayor esfuerzo para implementar dicho requisito.

Existen distintos factores y dimensiones a considerar en la estimación y priorización de requisitos, muchos de ellos de carácter cualitativo. A pesar de que existen muchas técnicas de estimación y priorización en desarrollo de software, la mayoría se concentran en analizar los requisitos funcionales, dejando de lado los requisitos de calidad [11].

El objetivo del presente estudio en curso es analizar el estado del arte en cuanto a la estimación del esfuerzo y priorización de requisitos de calidad. Específicamente, interesa descubrir los factores que se deben tener en cuenta para la estimación y priorización de dichos requisitos, así como los mecanismos y herramientas para evaluarlos y qué factores del contexto del proyecto pueden incidir en dicha evaluación. Con todo ello se busca construir una base conceptual sólida que permita posteriormente proponer una técnica para priorizar y estimar el esfuerzo requerido para la implementación de requisitos de calidad en entornos de desarrollo ágil del software.

El resto de este capítulo está estructurado de la siguiente manera: en la sección II se presenta el método seguido para el estudio sistemático. En la sección III se muestra la ejecución del método y algunos resultados preliminares. Por último, en la sección IV se presentan las conclusiones y trabajo futuro.

## 2. MÉTODO

El trabajo se estructuró buscando verificar las siguientes preguntas de investigación:

- RQ1 ¿Cuáles criterios son considerados durante la fase de ingeniería de requisitos para elegir los factores de calidad de un producto software específico?
- RQ2 ¿Cómo (herramientas, métodos, modelos) son evaluados estos factores de calidad?
- RQ3 ¿Qué factores del contexto afectan estos criterios?
- Para responder a dichas preguntas se define y ejecuta una revisión de literatura, que es el objeto de este trabajo.

El método seguido es una combinación de las propuestas hechas para la realización de estudios sistemáticos por Kitchenham [13] y Wohlin [14]. La propuesta de Kitchenham [13] está basada en la búsqueda de artículos mediante el uso de palabras clave en múltiples bases de datos. Esto requiere que la comunidad emplee un conjunto de palabras clave de una manera común para poder obtener sus trabajos [15, 16]. En nuestro caso, no existe una terminología estandarizada o consolidada, por lo que una búsqueda utilizando solo la propuesta por Kitchenham resultaría en la pérdida de varios artículos relevantes (*low recall*) si estos no usan los términos empleados en la búsqueda.

Una metodología que no presenta este impedimento, es la propuesta por Wohlin [14]. Dicha propuesta está basada en la técnica del *snowballing*. En esta aproximación se parte de un conjunto de artículos inicial (*seed*) y se obtienen las referencias bibliográficas dentro del artículo (*backward snowballing*), así como los trabajos que citan a dicho artículo (*forward snowballing*) para orientar la revisión de literatura. Las referencias y citas que cumplen con los criterios de búsqueda son añadidos al conjunto inicial, y se procede a nueva iteración, obteniendo más referencias y citas hasta llegar al punto de saturación (esto es, cuando no se encuentran más artículos relevantes).

Esta aproximación no depende de palabras clave, pero al mismo tiempo, tiene también sus inconvenientes. Wohlin [14] no define como obtener el conjunto de artículos semilla, y una mala elección puede llevar a sesgos en los resultados. Para resolver este riesgo, se propone el uso de las pautas definidas en Kitchenham [13] para obtener los artículos semilla que luego guiarán la búsqueda aplicando las técnicas de Wohlin [14], combinando así ambas aproximaciones. El método específico empleado se indica mediante el diagrama BPMN de la Figura 1.

Para la selección de los artículos semilla, según la aproximación de Kitchenham [13], se inicia definiendo la problemática y preguntas de investigación. A partir de aquí, se define una cadena de búsqueda que contenga los términos relevantes para resolver las preguntas de investigación, estructurados mediante los operadores ANDs y ORs. Después de varias iteraciones, se definió dicha cadena de búsqueda como sigue:

*(software OR system) AND (requirement prioritization OR requirement validation OR requirement negotiation) AND (factor OR characteristic OR attribute OR criteria) AND (quality OR non-functional OR security OR performance) AND (decision maker OR stakeholder).*

Posteriormente, se ingresa dicha cadena de búsqueda en una (o varias) base de datos o motor(es) de búsqueda. En nuestro caso, hemos seleccionado el motor Google Scholar. Al ser el motor que, con diferencia, incluye la mayor parte

de artículos indexados, se ha considerado que no era necesario incluir otros motores u bases de datos. El siguiente paso es la definición de los criterios de inclusión y exclusión para la confección de la lista de los artículos semilla (Tabla 1). Cabe destacar que para la selección de los artículos semilla se ha primado la calidad por encima de la cantidad. En este sentido, se ha decidido limitar los artículos a un conjunto de revistas y congresos de calidad y relevantes en el área.

En particular, se han limitado los resultados a artículos publicados en: *Requirements Engineering*, *IEEE Transactions on Software Engineering*, *IEEE Software*, *Information and Software Technology*, *Journal of Systems and Software*, *Empirical Software Engineering*, *Communications of the ACM*, *Software Quality Journal*, y *Systems Engineering*. Dicha lista de revistas y congresos se ha confeccionado en base a la experiencia y conocimiento previo de los autores.

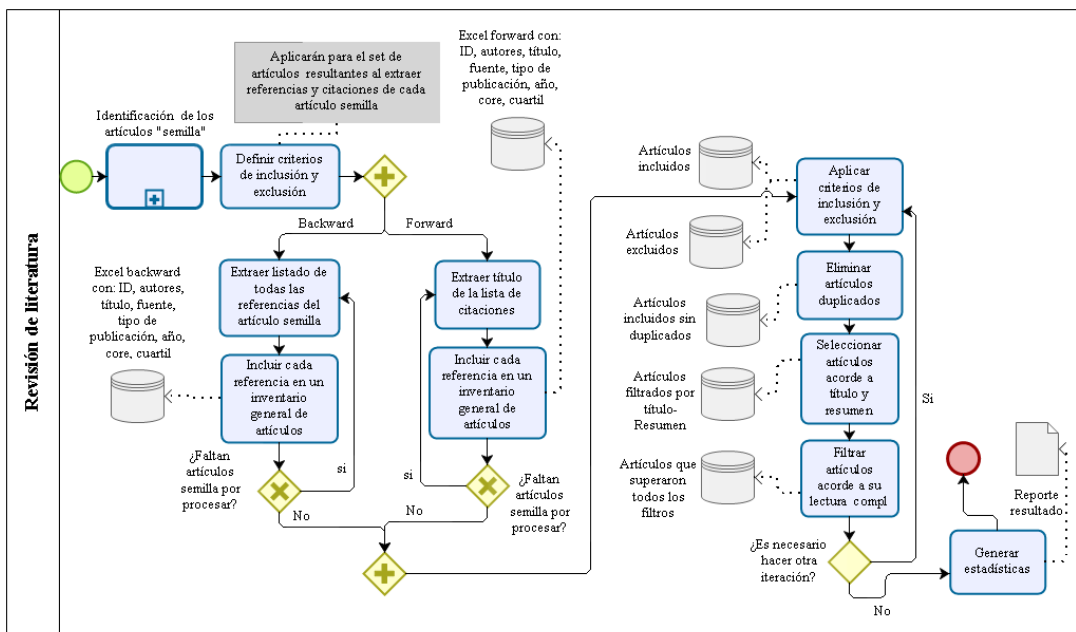


Figura 1. Método para el estudio sistemático

Tabla 1. Criterios para filtrado de artículos semilla

Criterios de Inclusión (CI)	
CI1:	contribuciones publicadas entre los años 1999-2019.
CI2:	artículos escritos en idioma inglés.
CI3:	contribuciones en revistas o congresos de calidad y cuya temática entre dentro del ámbito de las preguntas de investigación.
CI4:	artículos cuyo enfoque estén dentro del alcance de la priorización y estimación de requisitos de calidad en desarrollo de software.
Criterios de Exclusión (CE)	
CE1:	estudios publicados antes del año 1999.
CE2:	artículos escritos en idiomas diferentes al inglés.
CE3:	contribuciones que no estén en revistas o congresos de calidad o cuya temática no entre dentro del ámbito de las preguntas de investigación.
CE4:	artículos cuyo enfoque estén fuera del alcance de la priorización y estimación de requisitos de calidad en desarrollo de software.

Los artículos recuperados mediante Google Scholar se filtran acorde a los criterios definidos en la Tabla 1. En este punto se emplea un protocolo para registrar los artículos encontrados, tal como se muestra en la Tabla 2.

Tabla 2. Protocolo para extracción de información

Item Type	Publication Year	Author	Title	Publication Title	ISBN/ISSN /DOI	Url	Abstract

En la Tabla 2, *item type* puede ser diligenciado con una de las siguientes opciones: libro, capítulo, conferencia, artículo en revista, reporte, tesis, otro. Con los criterios de inclusión y exclusión previamente definidos se aplica un primer filtrado para elegir los artículos semilla y comenzar con la aproximación tipo *snowballing*.

En este punto se refinan los criterios de inclusión y exclusión para obtener el listado de artículos que van a formar parte del resultado final del estudio sistemático (Tabla 3). En estos criterios, ya no se limitan los artículos a los congresos o revistas definidas anteriormente, aunque para asegurar la calidad de estos, sí se define como criterio que los artículos deben ser publicados en revistas de primer o segundo cuartil JCR, o en congresos con ranking CORE-A o CORE-A\*. Asimismo, se limitan los resultados a aquellos artículos que presentan una propuesta concreta (técnica o método).

**Tabla 3.** Criterios para filtrado de artículos del estudio

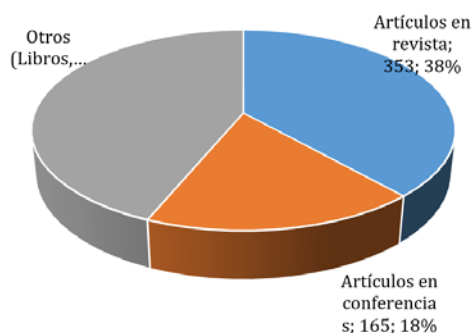
<b>Criterios de Inclusión (CI)</b>
<b>CI1:</b> contribuciones publicadas entre los años 1999-2019.
<b>CI2:</b> artículos escritos en idioma inglés.
<b>CI3:</b> contribuciones en revistas cuyo cuartil sea 1 o 2, o presentadas en eventos con ranking CORE-A o CORE-A*.
<b>CI4:</b> artículos cuyo enfoque estén dentro del alcance de la priorización y estimación de requisitos de calidad en desarrollo de software.
<b>CI5:</b> artículos que detallen técnicas, métodos para priorizar y estimar requisitos de calidad en desarrollo de software.
<b>Criterios de Exclusión (CE)</b>
<b>CE1:</b> estudios publicados antes del año 1999.
<b>CE2:</b> artículos escritos en idiomas diferentes al inglés.
<b>CE3:</b> contribuciones que no estén en revistas cuyo cuartil sea 1 o 2, ni presentadas en eventos con ranking CORE-A o CORE-A*.
<b>CE4:</b> artículos cuyo enfoque estén fuera del alcance de la priorización y estimación de requisitos de calidad en desarrollo de software.
<b>CE5:</b> artículos cuyo enfoque sea la revisión sistemática de literatura o comparativos entre estudios ya realizados.

Con los criterios definidos, se procede a efectuar dos actividades simultáneas: el *backward snowballing* y el *forward snowballing*. En el *backward snowballing* se extrae (por cada artículo semilla) el listado de todas las referencias del escrito. Mientras que en el *forward snowballing* se genera un inventario de las citas que ha logrado cada artículo semilla. Estas dos actividades se repiten acorde al conjunto de artículos semilla o base. Luego se procede a aplicar los criterios de inclusión y exclusión. Posteriormente se eliminan los artículos repetidos entre los listados de citas y referencias de todos los artículos base. Para los artículos seleccionados hasta este punto se diligencia un protocolo de recolección de información.

El *snowballing* se hace en varias iteraciones, tomando la decisión, acorde a los resultados, sobre si es necesario hacer una nueva iteración o finalizar.

### 3. RESULTADOS Y ANÁLISIS

En este apartado se presentan los resultados al seguir la combinación metodológica presentada en la sección anterior. Para la elección de los artículos semilla, se ejecutó la búsqueda en Google Scholar con la cadena de búsqueda definida. Los resultados de dicha búsqueda se presentan en la Figura 2.

**Figura 2.** Resultados de la búsqueda inicial

Los 920 resultados se procesaron acorde los criterios de inclusión/exclusión para los artículos semilla definidos en la Tabla 1. En una primera iteración se aplicaron los tres primeros criterios de inclusión/exclusión (CI/CE 1-3), puesto que estos no requieren la lectura del artículo. Al aplicar dichos filtros, quedaron 41 documentos distribuidos en los distintos congresos y revistas seleccionados (Tabla 4).

**Tabla 4.** Distribución de contribuciones luego del primer filtro

<b>Nombre revista o evento</b>	<b>#</b>
Requirements Engineering	10
IEEE Transactions on Software Engineering	5
IEEE Software	3
Information and Software Technology	4
Journal of Systems and Software	7
Empirical Software Engineering	3
Communications of the ACM	1
Software Quality Journal	1
Systems Engineering	7
Total	41

A estos 41 documentos se les aplicó el cuarto criterio de inclusión/exclusión (CI/CE 4). Primero se procesaron leyendo su título y resumen, resultando en 22 artículos. Por último, se realizó una lectura completa de los 22 artículos, resultando en los 7 artículos semilla que se presentan en la Tabla 5.

**Tabla 5.** Artículos semilla seleccionados

Título	Año
Priority assessment of software process requirements from multiple perspectives	2006
A framework for prioritization of quality requirements for inclusion in a software project	2013
A systematic literature review of software requirements prioritization research	2014
Dynamic decision models for staged software product line configuration	2014
Adaptive Requirements Prioritization (ARP): Improving Decisions between Conflicting Requirements	2015
An industry experience report on managing product quality requirements in a large organization	2017
Understanding requirement prioritization artifacts: a systematic mapping study	2017

Con los 7 artículos seleccionados como semilla se aplicaron las técnicas de *backward snowballing* y *forward snowballing* empleando los criterios de inclusión y exclusión que se enuncian en la Tabla 3.

Asimismo, se diligenció el protocolo para extracción de información detallada que incluye ítems como: palabras clave, país, continente, filiación del autor principal (empresa o academia), aparición de temas como: priorización, negociación, estimación; también se verificó si hacían mención de requisitos de calidad específicos (seguridad, fiabilidad, robustez, entre otros); se extrajeron también las preguntas de investigación de cada artículo, se verificó si había algún tipo de validación en caso que se hicieran propuestas nuevas.

Los resultados del *snowballing* se presentan en las Tablas 6 y 7. En ambas iteraciones se obtuvieron 15 artículos nuevos, resultando 37 artículos en total (7 artículos semilla + 15 artículos primera iteración + 15 artículos segunda iteración). Después de la segunda iteración se constató que al aplicar nuevamente el *snowballing* no se encontró más resultados relevantes, con lo que se dio por concluida la extracción de artículos.

**Tabla 6.** Resultados del proceso de snowballing - Iteración 1

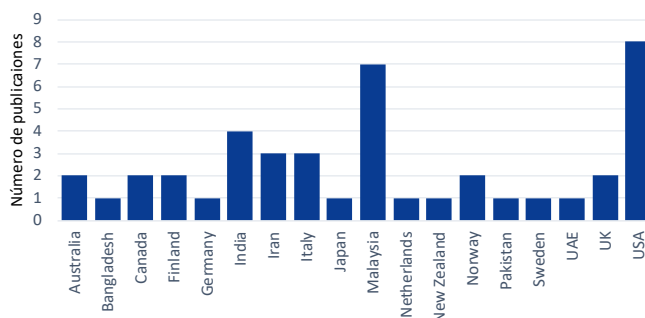
Primera iteración (Snowballing sobre los 7 artículos semilla)	Backward	Forward
Snowballing antes de aplicar los criterios de inclusión/exclusión	521	449
Artículos resultantes de la aplicación de los criterios CI/CE 1-3	65	116
Artículos resultantes de la aplicación de los criterios CI/CE 4-5 (lectura por título y abstract)	19	17
Artículos resultantes de la aplicación de los criterios CI/CE 4-5 (lectura completa)	7	8
Nuevos artículos al finalizar la iteración		15

**Tabla 7.** Resultados del proceso de snowballing - Iteración 2

Segunda iteración (Snowballing sobre los 15 artículos semilla)	Backward	Forward
Snowballing antes de aplicar los criterios de inclusión/exclusión	515	449
Artículos resultantes de la aplicación de los criterios CI/CE 1-3	31	116
Artículos resultantes de la aplicación de los criterios CI/CE 4-5 (lectura por título y abstract)	11	17
Artículos resultantes de la aplicación de los criterios CI/CE 4-5 (lectura completa)	7	8
Nuevos artículos al finalizar la iteración		15

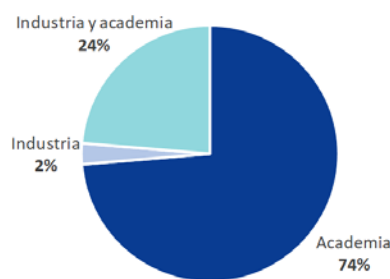
Algunas estadísticas relevantes se presentan a continuación. En la Figura 3 se presenta la distribución geográfica de los estudios para identificar en qué país se ha realizado mayor contribución al tema de investigación.

Para este trabajo se identificó el país de afiliación de todos los autores, contabilizando los países involucrados en cada artículo (se contabiliza una sola vez cada país para cada artículo, aunque el artículo tenga dos o más autores del mismo país). En los resultados se observa que el país con mayor número de estudios en el tema es Estados Unidos con 8 publicaciones, seguido de Malaysia con 7.



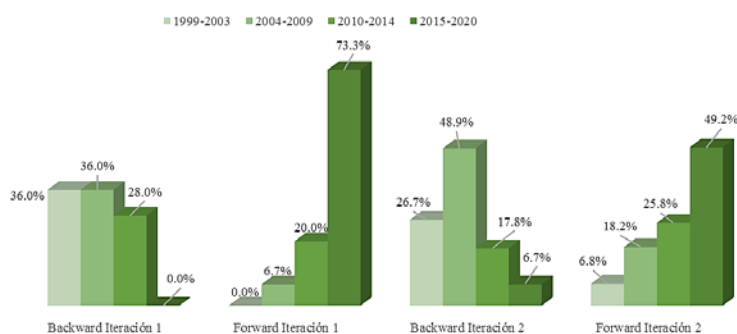
**Figura 3.** Estadísticas de número de publicaciones por país

Se ha analizado también el porcentaje de artículos según si sus autores provenían del sector académico o industrial (Figura 4). Se evidencia que la mayoría de contribuciones (74%) tienen solo autores académicos. El 24% de los trabajos incluyen una combinación de autores de la industria y académicos, mientras que solo el 2% tiene autores exclusivamente de la industria.



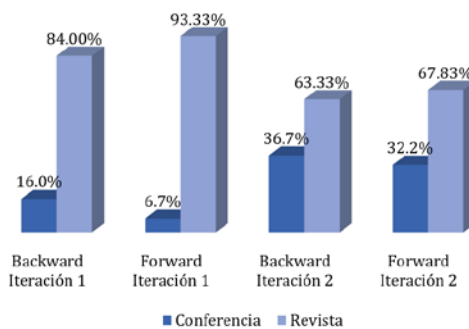
**Figura 4.** Porcentaje de estudios de acuerdo a la filiación (industria o academia)

En la Figura 5 se muestra la distribución de los artículos revisados por rango de años. Como cabía esperar, se observa que el *backward snowballing* permite obtener los artículos publicados en los primeros años del periodo considerado, mientras que la variante *forward* obtiene los publicados en los últimos años. En términos agregados, los periodos 2004-2009 y 2015-2020 son los más fructíferos ya que registran un mayor número de artículos.



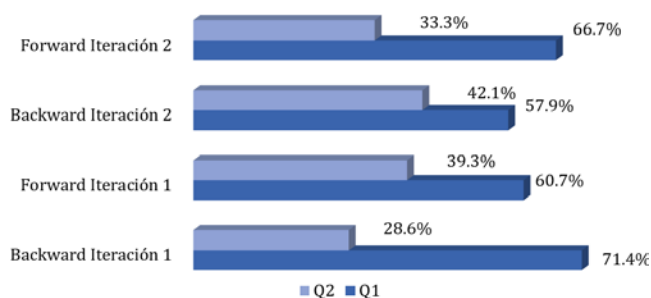
**Figura 5.** Distribución de artículos por rango de años

En cuanto al tipo de publicación, la Figura 6 muestra un claro predominio de los artículos publicados en revistas frente a los publicados en actas de conferencias. A diferencia del gráfico de la Figura 5, aquí las diferencias en las distribuciones no se observan entre los dos diferentes tipos de *snowballing* sino entre las dos iteraciones. Claramente, en la primera iteración la diferencia entre revistas y conferencias es mucho más abrumadora que en la segunda iteración. Esto puede ser debido a que los artículos resultantes durante la fase de obtención de artículos semilla son mayoritariamente de revista, y estos pueden tener una cierta tendencia a citar más (y ser más citados por) artículos de revista que de congreso. De aquí se desprende la necesidad de ejecutar más de una iteración en el proceso de snowballing, pues se observa que, durante la segunda iteración, dicho posible sesgo es mitigado.



**Figura 6.** Distribución de artículos revisados acorde al tipo de publicación

Si se analiza con más de detalle la distribución de los artículos dentro de cada categoría y en la Figura 7 se puede observar, por ejemplo, que la mayoría de los artículos de revista pertenecen al primer cuartil. A pesar que hay ligeras diferencias en la distribución de artículos en cada iteración, dichas diferencias no son estadísticamente significantes.



**Figura 7.** Distribución de artículos en revistas por cuartil

Por lo que refiere a la calidad de las conferencias, en la Figura 8 se observa un claro predominio de artículos publicados en el rango de menor calidad (A) según el índice CORE. Ello no es de extrañar porque, a diferencia de la distribución en cuartiles, la distribución de las conferencias en los rangos del índice CORE no está equilibrada: la proporción de conferencias A es 3.5 veces mayor que la de conferencias A\*. En todo caso, no deja de llamar la atención que en la primera iteración no se halle ningún artículo publicado en conferencias de la máxima calidad. Esto puede ser debido a que en la primera iteración se encontraron pocos artículos de congreso (solo 4 artículos de congreso durante el *backward* de la primera iteración y 2 artículos de congreso durante el *forward*). Siendo la proporción de conferencias A 3.5 veces mayor que la de conferencias A\*, el hecho de no haber encontrado ningún artículo de congreso A en la primera iteración es consistente con dichas proporciones.

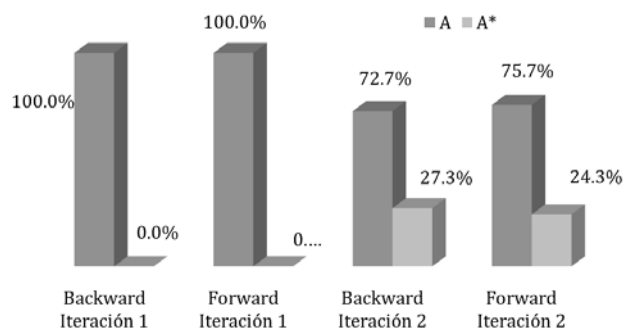


Figura 8. Distribución de artículos en conferencias por ranking

Por último, cabe destacar que en el *wordcloud* la herramienta utilizada (librería *wordcloud* de R) elimina automáticamente un conjunto de palabras habituales y de poca relevancia (*a, and, the, ...*). Además de estas palabras, se han eliminado los siguientes términos por ser también muy habituales y poco relevantes: *achieve, addition, aim, also, although, architectural, available, among, based, can, case, company, considered, different, domain, existing, first, high, large, making, multiple, number, obtained, order, outcome, paper, perspective, phase, phenomenon, presented, problem, proposed, research, results, set, several, show, study, terms, three, two, use, used, using, various*. Se puede observar que la mayoría de los términos presentados fueron usados en la cadena de búsqueda. Lo anterior no implica que estén presentes en la gran mayoría de artículos. Asimismo, aparecen nuevos términos interesantes como, por ejemplo, *satisfaction* y *technical debt*.

#### 4. CONCLUSIONES

En este capítulo se presenta el trabajo en curso de una revisión sistemática de la literatura sobre la estimación y priorización de requisitos de calidad. Para ello se aplicó una combinación de las pautas de desarrollo de estudios sistemáticos de Kitchenham [13] y Wohlin [14]. Mediante la combinación de ambos protocolos, se buscó mitigar los riesgos que planteaba cada una de las aproximaciones por separado. Con la aplicación de las pautas de Kitchenham, se obtuvo una primera lista de artículos de manera sistemática que ha sido luego utilizada como semilla para aplicar el protocolo de *snowballing* de Wohlin. Se ha aplicado tanto *backward snowballing* como *forward snowballing* en 2 iteraciones.

En los resultados de los artículos obtenidos se evidencia que la mayoría de contribuciones provienen de la academia, con un bajo porcentaje de artículos escritos por autores de industria. También se observó que Estados Unidos ha hecho la mayor cantidad de contribuciones al tema de estimación y priorización de requisitos en el rango observado (1999-2019).

Como trabajo futuro se espera finalizar el estudio sistemático mediante la aplicación de técnicas de extracción de datos de los artículos resultantes, así como técnicas de análisis y síntesis sobre dichos datos para poder responder a las preguntas de investigación planteadas.

También se espera aplicar el método descrito en este capítulo en otras áreas de investigación de la ingeniería del software.

#### REFERENCIAS

- [1] Abbas, N., Gravell, A. y Wills, G. (2010). The impact of organization, project and governance variables on software quality and project success. En Agile Conference. San Diego, USA.
- [2] Howarth, T. y Greenwood, D. (2017). Construction quality management: Principles and practice. Routledge.
- [3] Kabir, M., Rehman, M. y Majumdar, S. (2016). An analytical and comparative study of software usability quality factors. En 7th IEEE International Conference on Software Engineering and Service Science. Lisboa, Portugal.
- [4] Sogeti. (2017). World Quality Report 2017 – 2018. Recuperado: [https://www.sogeti.com/globalassets/global/downloads/testing/wqr-2017-2018/wqr\\_2017\\_v9\\_secure.pdf](https://www.sogeti.com/globalassets/global/downloads/testing/wqr-2017-2018/wqr_2017_v9_secure.pdf).

- [5] Tricentis. (2018). Software Fail Watch. Recuperado: <https://www.tricentis.com/resources/software-fail-watch-5th-edition/>.
- [6] Oriol, M. et al. (2019). Data-driven elicitation of quality requirements in agile companies. En International Conference on the Quality of Information and Communications Technology. Boston, USA.
- [7] López, L. et al. (2018). How practitioners manage quality requirements in rapid software development: A Survey. En 1st International Workshop on Quality Requirements in Agile Projects. Madrid, Spain.
- [8] Sehra, S. et al. (2017). Research patterns and trends in software effort estimation. Information and Software Technology, 91, 1-21.
- [9] Naik, P. y Nayak, S. (2017). Insights on research techniques towards cost estimation in software design. International Journal of Electrical & Computer Engineering, 7(5), 235-246.
- [10] Vazquez, P., Panizzi, M. y Bertone, R. (2019). Estimación del esfuerzo del proceso de implantación de software basada en el método de Puntos de Caso de Uso/Estimating the effort of the software implantation process based on the Use Case Points method. Brazilian Journal of Development, 5(2), 1809-1822.
- [11] Ali, A. et al. (2018). Requirements prioritization: A comparison between traditional and agile (Scrum and FDD). En 16th International Conference on Statistical Sciences. Mexico, Mexico.
- [12] Leshob, A., Hadaya, P. y Renard, L. (2019). Software requirements prioritization with the goal-oriented requirement language. In International Conference on e-Business Engineering. Buenos Aires, Argentina.
- [13] Kitchenham, B. y Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering version 2.3. Engineering, 45, 1051-162.
- [14] Wohlin, C. (2016). Second-generation systematic literature studies using snowballing. En 20th International Conference on Evaluation and Assessment in Software Engineering. Berlin, Germany.
- [15] Budgen, D. et al. (2008). Using mapping studies in software engineering. PPIG.
- [16] Petersen, K. et al. (2008). Systematic mapping studies in software engineering. En 12th International Conference on Evaluation and Assessment in Software Engineering. London, UK.