

Revisiones individuales de software de código y diseño detallado por fuera del PSP: un estudio de mapeo sistemático

Fernando Acerenza

Tesis de Maestría presentada al Programa de Maestría en Ingeniería de Software, Centro de Posgrados y Actualización Profesional en Informática, Instituto de Computación, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magister en Ingeniería de Software.

Director de tesis:

Dr. Diego Vallespir

Montevideo – Uruguay

Diciembre de 2016

RESUMEN

En el contexto del Personal Software Process (PSP) se han encontrado excelentes resultados al aplicar revisiones personales sobre código y diseño detallado. Sin embargo, existen varios estudios que indican que las revisiones personales de software no han sido adquiridas como una práctica habitual en la industria.

Por esto, resulta de interés preguntarse acerca de los motivos por los cuales las revisiones individuales no han sido adoptadas de forma masiva y, además, identificar la evidencia existente del uso de las mismas.

El objetivo de este trabajo es recoger información con el fin de conocer el estado actual de la investigación y de la experiencia en revisiones individuales de software aplicadas a código y diseño detallado por fuera del marco del PSP. Para alcanzar dicho objetivo se llevó a cabo un estudio de mapeo sistemático a partir de cuatro preguntas de investigación vinculadas con los siguientes aspectos: motivos para realizar revisiones individuales sobre código y diseño detallado, motivos para no adoptar estas prácticas, datos de efectividad y costo, y factores que influyen en el rendimiento. Dicho estudio arrojó 12 estudios primarios que fueron clasificados de acuerdo con cada pregunta de investigación.

Los resultados muestran que la investigación con foco en las revisiones individuales de software son aún insuficientes para la ingeniería de software. Pese a esto, se observa que el uso de las revisiones individuales de software son beneficiosas en distintos escenarios, como por ejemplo: donde se aplican técnicas de lectura específicas, cuando se requiere transferir conocimiento o detectar errores en etapas tempranas. La mayoría de los estudios comparan el rendimiento de distintas técnicas de lectura para llevar a cabo estas revisiones y, mediante experimentos y estudios de caso, evalúan técnicas que ya han sido implementadas en la práctica. Además, a partir de los resultados se identifican motivos por los que se realizan revisiones individuales y factores que impactan en su desempeño.

Este estudio sugiere que la investigación enfocada en revisiones de software a nivel individual en la industria se encuentra aún en un primer estadio de desarrollo. Dado que las revisiones personales dentro del PSP han arrojado buenos resultados, resulta llamativo que fuera de él no hayan sido aún abordadas tanto por la industria como por la comunidad académica. Es entonces necesario realizar nuevos trabajos que permitan cubrir el vacío de investigación acerca de las revisiones individuales.

If you don't know where you are, a map won't help.

— Watts S. Humphrey

AGRADECIMIENTOS

Esta tesis no es solo fruto de mi trabajo, sino también del soporte que recibí durante su desarrollo de parte de muchas personas. Por eso quiero agradecer enormemente a todos ellos.

A Carolina por su amor, su paciente e incondicional apoyo y sus incontables revisiones de escritura.

A mis compañeros del grupo de investigación de ingeniería de software por sus aportes para mejorar este trabajo.

A Diego, mi tutor, por su guía, ayuda y disposición a lo largo de este camino de aprendizaje.

A mis compañeros de trabajo, en especial a Silvia, por su comprensión y constante soporte.

Por último, a mis amigos y a toda mi familia. En particular a mi madre por fomentar en mí las ganas de aprender siempre.

ÍNDICE GENERAL

1	INTRODUCCIÓN	1
1.1	Motivación	2
1.2	Objetivos	2
1.3	Mapeo sistemático de la literatura sobre revisiones individuales	4
1.4	Estructura del documento	5
2	ANTECEDENTES: DEFINICIONES FUNDAMENTALES EN INGENIERÍA DE SOFTWARE	7
2.1	Ingeniería de Software	7
2.2	Calidad de Software	8
2.3	Defectos	9
2.4	Verificación y validación	10
2.5	Revisiones de software	11
2.6	Personal Software Process (PSP)	14
2.7	Revisiones individuales fuera del PSP sobre código y diseño detallado	15
3	ANTECEDENTES: ESTUDIOS DE MAPEO SISTEMÁTICO	17
3.1	Aspectos relevantes de los estudios de mapeo sistemático	17
3.2	Trabajos previos de estudios de mapeo sistemático en ingeniería de software	26
4	REVISIONES INDIVIDUALES DE SOFTWARE FUERA DEL PSP: MAPEO SISTEMÁTICO	31
4.1	Planificación del mapeo sistemático	31
4.1.1	Preguntas de investigación	33
4.1.2	Estrategia y criterios de búsqueda	34
4.1.3	Criterio de selección de estudios	36
4.1.4	Estrategia de extracción de datos y esquema de clasificación	37
4.1.5	Síntesis de los datos extraídos	45
4.2	Ejecución del mapeo sistemático	45
4.3	Resultados y Discusión	46
4.3.1	Extracción y análisis de datos	46
4.3.2	Respuestas a las preguntas de investigación	57
4.3.3	Discusión	67
4.4	Amenazas a la validez	69

4.5	Conclusiones	72
5	DIFICULTADES AL APLICAR MAPEOS SISTEMÁTICOS EN INGENIERÍA DE SOFTWARE	75
5.1	Dificultades del mapeo sistemático realizado	75
5.2	Otros problemas relacionados con los mapeos sistemáticos	82
6	CONCLUSIONES Y TRABAJOS A FUTURO	85
6.1	Conclusiones	85
6.2	Trabajos a futuro	91
BIBLIOGRAFÍA 93		
A	APÉNDICE A: PROTOCOLO DEL MAPEO SISTEMÁTICO	101
A.1	Motivación y antecedentes	101
A.1.1	Equipo	101
A.1.2	Introducción y necesidad de un estudio de mapeo sistemático	101
A.1.3	Alcance del estudio	102
A.1.4	Preguntas de investigación	102
A.2	Proceso	103
A.3	Estrategia de búsqueda	104
A.4	Criterios de selección	106
A.4.1	Criterios de Inclusión	106
A.4.2	Criterios de exclusión	106
A.5	Extracción de datos	107
A.6	Síntesis	108
A.7	Limitaciones del estudio	108
A.8	Reporte	108
A.9	Evaluación del Protocolo	109
A.10	Ajuste del Protocolo	109
A.11	Cronograma	109
B	APÉNDICE B: CONCEPTOS DE INGENIERÍA DE SOFTWARE BASADA EN EVIDENCIAS	111

ÍNDICE DE FIGURAS

Figura 1	Esquema de clasificación	38	
Figura 2	Pantalla de búsqueda avanzada de ACM - setiembre 2014		76
Figura 3	Pantalla de búsqueda por comando de IEEE	77	
Figura 4	Proceso de mapeo sistemático	104	

ÍNDICE DE CUADROS

Cuadro 1	Diferencias entre RSL y estudio de mapeo	19
Cuadro 2	Fases de un mapeo sistemático	21
Cuadro 3	Pasos y salidas del proceso de mapeo	32
Cuadro 4	Términos a partir de descomposición en aspectos	35
Cuadro 5	Definición de estrategias empíricas	40
Cuadro 6	Clasificación propuesta por Wieringa	42
Cuadro 7	Clasificación de técnicas de lectura	44
Cuadro 8	Estudios seleccionados	47
Cuadro 9	Textos seleccionados por año	48
Cuadro 10	Medio de publicación de los trabajos	48
Cuadro 11	Lugares de publicación de los trabajos	49
Cuadro 12	Contexto de la investigación	49
Cuadro 13	Instituciones de las investigaciones	50
Cuadro 14	Artefacto sobre el que se aplica la revisión de software	50
Cuadro 15	Clasificación de estrategias empíricas	55
Cuadro 16	Clasificación de tipos de investigación	56
Cuadro 17	Clasificación de técnicas de lectura	56
Cuadro 18	Artículos por preguntas	58
Cuadro 19	Artículos por respuestas a la pregunta RQ1	59
Cuadro 20	Artículos por respuestas a la pregunta RQ2	60
Cuadro 21	Artículos por respuestas a la pregunta RQ3 - Efectividad	61
Cuadro 22	Artículos por respuestas a la pregunta RQ3 - Costo	64
Cuadro 23	Artículos por respuestas a la pregunta RQ4	66
Cuadro 24	Fases del estudio de mapeo sistemático	103
Cuadro 25	Términos a partir de descomposición en aspectos	105

INTRODUCCIÓN

El software se ha convertido en un componente fundamental para la humanidad, a la vez que afecta diferentes aspectos de nuestra vida. Se ha propagado a través de múltiples contextos como el comercio, la cultura y las actividades del día a día. Por su parte, la ingeniería de software, a través de un proceso adecuado y una colección de prácticas y herramientas permite que los profesionales de software puedan construir sistemas complejos y de alta calidad [Pressman, 2015].

Dentro de los desafíos a los que se enfrenta la industria del software se encuentra la obtención de productos de calidad. Para medir la calidad del software se pueden considerar distintos factores, uno de los cuales es la cantidad de defectos con la que el software llega a los usuarios finales.

Por esto, la detección de defectos durante el proceso de desarrollo de software resulta importante a la hora de obtener productos de calidad. Al mismo tiempo, su identificación debe ser lo más temprana posible, de modo de disminuir el costo de corregirlos [Pressman, 2015].

Existen numerosas técnicas para detectar defectos. Entre ellas se encuentran las revisiones de software, las cuales logran detectar defectos de forma temprana e incluso prevenir su inyección [Pressman, 2015; Kan, 2002]. Las revisiones de software son procesos formales o informales, donde se examina un producto de software en busca de defectos o desviaciones con respecto a los estándares y especificaciones. El proceso puede incluir, o no, una reunión entre los revisores [IEEE, 2008].

Las revisiones de software se aplican a los distintos artefactos que se generan durante todo el ciclo de desarrollo de software. Pueden constituir documentos de requerimientos, modelos de diseño, código, entre otros [Kan, 2002]. La salida de un proceso de revisión consta de una lista de cuestiones que no han sido tenidas en cuenta o defectos a corregir y, adicionalmente, se incluye el estatus técnico del ítem de trabajo revisado [Pressman, 2015]. Además, la utilización de las revisiones en software permite obtener importantes beneficios como la disminución en costo y la mejora en calidad y productividad [Kan, 2002].

El proceso de revisión de software incluye determinadas etapas en las que la revisión es llevada a cabo por un individuo, por ejemplo, el paso de chequeo individual dentro de un proceso de inspección de software [Fagan, 1999] o la

fase de preparación individual en revisiones de software formales. Asimismo, la revisión individual puede ser realizada por el autor del artefacto a revisar o por un par [Cohen, 2013; Burnstein, 2006].

Las revisiones de software a nivel individual son tenidas en cuenta por el Personal Software Process (PSP) [Humphrey, 1995]. Este es un proceso que incluye prácticas disciplinadas para la gestión y mejora de desempeño personal de los ingenieros de software. Entre esas prácticas se destacan algunas tareas vinculadas al aseguramiento de la calidad de los productos, como las revisiones personales de diseño detallado y código. Estas son realizadas por el autor del artefacto, quien revisa el código o modelo de diseño utilizando una lista de verificación (*checklist*) en busca de defectos [Pomeroy-Huff et al., 2009].

1.1 MOTIVACIÓN

Las revisiones personales resultan efectivas y eficientes para mejorar la calidad de un producto y la productividad individual [Pomeroy-Huff et al., 2009]. Existen numerosos estudios que presentan datos de las mejoras que introducen el uso de las prácticas del PSP [Hayes y Over, 1997; Elminir et al., 2009; Khan, 2012; Ferguson et al., 1997; Paulk, 2010, 2006; Grazioli y Nichols, 2012]. Algunos ejemplos de buenos resultados son una mejora de la calidad del producto, medida como la densidad de defectos que se encuentran en el *testing* unitario, y un incremento en los defectos encontrados sobre los defectos totales antes de la compilación del código [Hayes y Over, 1997].

A pesar de los buenos resultados de la aplicación de las revisiones de software que, por ejemplo, se encuentran en el PSP, varios estudios describen que estas técnicas presentan una baja adopción en la industria del software [Laitenberger et al., 2002; Macchi y Solari, 2012; Ciolkowski et al., 2003; Macchi y Solari, 2013].

1.2 OBJETIVOS

Dadas las razones expuestas acerca de los buenos resultados dentro del PSP y de los trabajos que sugieren una baja adopción de revisiones en la industria, cabe preguntarse qué evidencia existe del uso de las revisiones individuales por fuera del PSP. Además, resulta de interés conocer los motivos para realizar revisiones individuales o para no adoptar estas prácticas.

Este trabajo busca profundizar acerca del uso de las revisiones individuales. Para alcanzar el objetivo, se considera un enfoque de investigación basado en evidencias. El fin de la ingeniería de software basada en evidencias es propor-

cionar los medios para integrar la mejor evidencia existente con la experiencia práctica en el proceso de toma de decisiones sobre el desarrollo y mantenimiento de software [Kitchenham et al., 2004].

Las herramientas principales para abordar la ingeniería de software basada en evidencia son las revisiones sistemáticas de la literatura y los estudios de mapeo sistemático. La revisión sistemática de literatura presenta un procedimiento riguroso, mediante el cual es posible evaluar, interpretar y agregar evidencia empírica con el fin de responder una pregunta de investigación. Por su parte, el mapeo sistemático, o estudio de alcance, tiene como propósito encontrar y clasificar estudios primarios dentro de un tópico específico [Kitchenham y Charters, 2007].

Por esto, se decide realizar un estudio de mapeo sistemático con el objetivo de conocer la evidencia que existe hasta la fecha acerca de las revisiones individuales de software aplicadas a código y diseño detallado, por fuera del marco del PSP. También, se pretende identificar cuestiones que no hayan sido investigadas y así proponer posibles caminos a seguir para futuros trabajos, cubriendo, de esa manera, los huecos en el tópico bajo estudio. A su vez, es intención de este trabajo servir como referencia para otros investigadores o estudiantes.

Los objetivos específicos de este trabajo son los siguientes:

- Buscar y clasificar, de forma sistemática, toda la evidencia actual del uso de revisiones individuales de software aplicadas a código y diseño detallado en marcos de trabajo donde no se utilice PSP. Para alcanzar este objetivo se elaboran cuatro preguntas de investigación.
 - ¿Por qué razones se realiza este tipo de revisiones individuales?
 - ¿Por qué razones las revisiones individuales no son adoptadas en la industria?
 - ¿Qué evidencia existe de efectividad y costo al aplicar revisiones individuales?
 - ¿Cuáles son los factores que afectan el desempeño de las revisiones individuales?
- Conocer las características de los estudios que atienden esta área de investigación
- Identificar líneas de trabajo para futuras investigaciones
- Generar una nueva experiencia de conducción de mapeo sistemático en ingeniería de software

- Analizar y reflexionar acerca de las características de un estudio de mapeo sistemático en ingeniería de software

1.3 MAPEO SISTEMÁTICO DE LA LITERATURA SOBRE REVISIONES INDIVIDUALES

Se realizó un mapeo sistemático dirigido por cuatro preguntas de investigación que indagan en el uso de las revisiones individuales de software sobre código y diseño detallado, en particular, en los motivos de adopción y rechazo, en datos de efectividad y costo, y en los factores que influyen en el rendimiento.

Una vez construido el protocolo final del mapeo, se ejecutó la búsqueda de la evidencia y se aplicaron los criterios de inclusión y exclusión. Como resultado del proceso de selección de estudios se obtuvieron 12 estudios primarios. Se desarrolló un formulario de extracción de datos para obtener los datos de los estudios y se definió un esquema de clasificación en orden de agrupar dichos estudios. A partir de los datos extraídos fue posible abordar las preguntas de investigación. Por último, se realizó la síntesis de la información y se reportaron los resultados del mapeo sistemático.

Los resultados obtenidos proveen algunas respuestas a las preguntas de investigación. Por un lado, se logra un acercamiento a los motivos por los cuales se realizan revisiones individuales. Respecto a la no adopción, no se encontró demasiada evidencia ya que solamente un artículo de los obtenidos abordaba esta cuestión. Por otra parte, se analizan resultados de efectividad y costo de las revisiones individuales. Con respecto a la última pregunta, la literatura encontrada describe ciertos factores que impactan en el rendimiento de las revisiones individuales.

Además, mediante la agrupación de los estudios en el esquema de clasificación y los datos extraídos se da cuenta de los tipos de investigaciones y características de los estudios existentes en esta área de investigación. La mayoría de los estudios compara el rendimiento de distintas técnicas de lectura para llevar a cabo las revisiones de software sobre código o diseño detallado. Gran parte de los trabajos, mediante experimentos y estudios de caso, evalúa técnicas que ya han sido implementadas en la práctica. A su vez, se presenta la distribución de los estudios por tipo de investigación, año de publicación, foros de publicación, instituciones participantes, técnicas de lectura utilizada y artefacto bajo revisión.

El principal hallazgo es la escasa investigación existente con foco específico en revisiones individuales de software. Esto resulta llamativo dado que dentro del PSP los resultados de conducir revisiones individuales sobre código y di-

seño detallado han sido excelentes. A partir de ello, se identifica la necesidad de que la comunidad de investigadores en ingeniería de software comience a generar investigación específica en el marco de las revisiones individuales de software sobre código y diseño. En particular, en busca de proveer evidencia del desempeño de las revisiones individuales en marcos donde no se utiliza el PSP.

Por otro lado, durante la realización de este estudio se han identificado distintas dificultades para llevar a cabo el proceso de mapeos sistemático, las cuáles son presentadas en este trabajo junto con las decisiones tomadas para abordar estos problemas. Además, se incluyen lecciones aprendidas señaladas por la literatura.

Finalmente, a partir de este estudio se identifican huecos de evidencia y se plantean líneas de investigaciones futuras. Además, este trabajo contribuye como referencia y punto de partida de futuras investigaciones.

1.4 ESTRUCTURA DEL DOCUMENTO

Esta tesis está organizada de la siguiente manera. En el capítulo 2 se presentan definiciones básicas de conceptos relacionados con las revisiones individuales de software y su contexto, una introducción a las revisiones de software donde se identifica su importancia y características básicas; e información relacionada al proceso PSP. Además, en el capítulo 3 se presenta una introducción a los estudios de mapeo sistemático y se discute acerca de la importancia, beneficios y limitaciones de este tipo de estudio. En el capítulo 4 se describe la metodología utilizada en este trabajo. A su vez, se presentan los resultados que se desprenden de la investigación. Por otro lado, en el capítulo 5 se señalan los problemas afrontados durante la realización de este estudio, así como lecciones aprendidas que identifica la literatura. Por último, en el capítulo 6, se da lugar a las observaciones finales y a pautas para trabajos futuros.

La tesis también contiene dos apéndices. En el Apéndice A se presenta el protocolo del mapeo sistemático. En el Apéndice B se incluye el reporte técnico de PEDECIBA 15-08, titulado "*Conceptos de ingeniería de software basado en evidencias*".

ANTECEDENTES: DEFINICIONES FUNDAMENTALES EN INGENIERÍA DE SOFTWARE

En este capítulo, de forma de ubicar este trabajo dentro del marco conceptual, se señalan las definiciones de conceptos relacionados con las revisiones individuales de software. Estos conceptos son: Ingeniería de software, Calidad, Defectos, Verificación y validación, Revisiones de Software y el Personal Software Process.

2.1 INGENIERÍA DE SOFTWARE

En general, dentro de la literatura pueden encontrarse diferentes definiciones para la ingeniería de software. Aquí se presentan las definiciones y características que se consideran más atinadas.

En el estándar para la terminología en ingeniería de software se define a la ingeniería de software como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, a la operación y al mantenimiento de software, y el estudio de estos enfoques. Es decir, la aplicación de la ingeniería al software [IEEE, 1990].

La ingeniería de software abarca los procesos, métodos y herramientas que permiten que se construyan sistemas informáticos complejos de calidad y en el tiempo adecuado. El proceso de software incorpora un marco de cinco actividades (comunicación, planificación, modelado, construcción y despliegue) que son aplicables a todos los proyectos de software. La práctica de la ingeniería de software es una actividad de resolución de problemas, la cual sigue una serie de principios básicos [Pressman, 2015].

Sommerville, por su parte, presenta la ingeniería de software como una disciplina de la ingeniería que se encarga de todos los aspectos relacionados con la producción de software, desde la especificación del software hasta el mantenimiento del mismo. Las actividades fundamentales que detalla son la especificación (definición del alcance y de las restricciones), el desarrollo (diseño y programación), la validación (se verifica el software para asegurar que este es lo que el cliente requiere) y la evolución del software (modificaciones). Por último, agrega que la ingeniería de software es un enfoque sistemático de la producción de software que toma en cuenta el costo, el cronograma y cuestio-

nes de dependencia, así como las necesidades de los clientes y productores de software [Sommerville, 2011].

2.2 CALIDAD DE SOFTWARE

La calidad se define como el grado en que un sistema, componente o proceso satisface los requerimientos especificados y cumple con las necesidades o expectativas del cliente [IEEE, 1990]. Pressman sostiene que la calidad es un proceso de software efectivo aplicado de tal manera que crea un producto útil que provee valor medible tanto para quien lo produce como para quien lo usa [Pressman, 2015].

La calidad del software no abarca solamente el hecho de si la funcionalidad del software se ha implementado correctamente, sino que también depende de atributos no funcionales del sistema. Existe un conjunto importante de atributos de calidad del software, tales como: comprensibilidad, portabilidad, seguridad, verificabilidad, usabilidad, confiabilidad, adaptabilidad, reusabilidad, disponibilidad, eficiencia, robustez y facilidad de aprendizaje [Sommerville, 2011].

Por otra parte, la gestión de la calidad del software se ocupa de asegurar que el software presente un bajo número de defectos y de que alcance los estándares requeridos de mantenimiento, fiabilidad, portabilidad, etc. Incluye la definición de estándares para los procesos y productos y establece procesos que permiten comprobar el seguimiento de estos estándares [Sommerville, 2011].

El aseguramiento de la calidad (QA) es la definición de procesos y estándares que deberían conducir a productos de alta calidad y a la introducción de procesos de calidad en el proceso de fabricación. A su vez, el control de calidad se entiende como la aplicación de estos procesos de calidad para eliminar los productos que no alcanzan el nivel requerido de calidad [Sommerville, 2011]. Otra descripción, del concepto de QA, que puede encontrarse dentro de la literatura señala que el aseguramiento de la calidad establece la infraestructura que da soporte a métodos de ingeniería de software sólidos, gestión racional de proyectos y acciones de control de calidad [Pressman, 2015]. En el mismo sentido, el estándar de la IEEE, define el aseguramiento de la calidad como un patrón planificado y sistemático de todas las acciones necesarias para ofrecer una adecuada confianza en que un artículo o producto cumple con los requisitos técnicos establecidos. Dentro de las acciones mencionadas se encuentran las actividades de verificación y validación [IEEE, 1990].

En software, el sentido más estricto de la calidad del producto es comúnmente reconocido como la falta de defectos en el producto. También es el significado más básico de conformidad con los requisitos, ya que si el software contiene

demasiados defectos funcionales, no se cumple el requisito básico de proporcionar la función deseada del software. Esta definición se expresa generalmente de dos maneras: tasa de defectos (por ejemplo, número de defectos por millón de líneas de código fuente, por punto de función, u otra unidad); y fiabilidad (por ejemplo, el número de fallos por n horas de funcionamiento, o el tiempo medio entre fallas) [Kan, 2002].

2.3 DEFECTOS

Existen varios conceptos relacionados con los defectos. Dentro del estándar de la IEEE se define anomalía como cualquier condición, en la documentación u operación del software, que genera un desvío de las expectativas basadas en productos de software previamente verificados o documentos de referencia. A su vez, se define una falta como un paso, proceso o definición de datos incorrecto en un programa de computadoras. Por su parte, una falla refiere a la incapacidad de un sistema o componente para realizar las funciones requeridas dentro de los requerimientos de desempeño especificados. Si bien algunas de las definiciones del término error coinciden con la de falta, existe otra definición, en este mismo glosario de la IEEE donde se lo define como una acción humana que produce un resultado incorrecto [IEEE, 1990].

Tomando en cuenta estas definiciones puede decirse entonces, que un *error* humano puede generar una *falta* que puede generar una *falla*. En el contexto del proceso de software los términos *defecto* y *falta* son sinónimos.

Pressman, por su parte, define la diferencia entre un defecto (falta) y un error basado en el alcance o contexto de la detección. Un error se define como aquel problema encontrado antes de la liberación del software al usuario final, y un defecto es descrito como un problema identificado únicamente después de la liberación del software [Pressman, 2015].

Los defectos pueden encontrarse en programas, en diseño, o incluso en los requerimientos, las especificaciones u otros documentos. Un defecto, es de hecho, cualquier cosa que disminuya la capacidad del programa para satisfacer por completo y de manera efectiva las necesidades del usuario. Puede verse, por lo tanto, como algo objetivo, que se puede identificar, describir y contar [Humphrey, 2009]. Humphrey subraya que para mejorar la calidad de un programa, resulta esencial que los ingenieros de software aprendan a manejar todos los defectos que se introducen en los programas.

Dos actividades relacionadas con la existencia de defectos son: la identificación de defectos (proceso de determinar qué defecto o defectos causaron una falla) y la corrección de defectos (proceso de cambiar el sistema para remover

los defectos). Estas actividades están comprendidas dentro de los procesos de verificación y validación.

2.4 VERIFICACIÓN Y VALIDACIÓN

Según el estándar de IEEE, la verificación, se entiende como el proceso de evaluación de un sistema o componente que permite determinar si los productos de una cierta fase de desarrollo satisfacen las condiciones impuestas en el inicio de esa fase. Por otro lado, validación es el proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar si cumple los requisitos especificados [IEEE, 1990].

En las actividades de verificación del software se pretende demostrar que un sistema cumple con los requerimientos y lograr descubrir defectos del programa antes de su puesta en servicio. Las actividades de verificación y validación pueden realizarse mediante técnicas dinámicas o estáticas [Sommerville, 2011].

El *testing*, o pruebas, constituye una de las técnicas dinámicas. La IEEE define el *testing* como el proceso de operación de un sistema o componente, bajo condiciones específicas, observación de los resultados, y evaluación de algún aspecto del sistema o componente [IEEE, 1990].

De forma de derivar un conjunto de pruebas que tengan gran probabilidad de descubrir errores en el software, se diseñan casos de prueba. Existen dos categorías diferentes de diseño de casos de prueba: de caja negra y de caja blanca. Por un lado, la categoría de caja blanca tiene foco en la estructura de control del programa. Los casos de prueba son derivados de tal forma que permiten asegurar que todas las sentencias sean ejecutadas al menos una vez durante la verificación y que se evalúen todas las condiciones lógicas. Por otro lado, las pruebas de caja negra son diseñadas para validar los requerimientos funcionales, sin tener en cuenta el funcionamiento interno del programa [Pressman, 2015].

Dentro de las técnicas estáticas se puede identificar el análisis automatizado de código, la revisión manual de código (o de otros artefactos de software) y la verificación formal (a partir de una especificación formal).

Algunas técnicas de revisión son: revisión de escritorio, recorrida, inspección y revisión técnica formal. Las técnicas estáticas, y en particular las revisiones, pueden aplicarse sobre cualquier artefacto generado durante el ciclo de desarrollo. Vale agregar que las técnicas dinámicas y estáticas pueden aplicarse de forma complementaria, donde una técnica puede encontrar defectos donde otra no ha podido [Sommerville, 2011; Pressman, 2015].

2.5 REVISIONES DE SOFTWARE

Las revisiones de software constituyen un filtro en el proceso de software. Esto es, las revisiones se aplican en varios momentos durante la ingeniería de software y sirven para descubrir defectos factibles de ser removidos [Pressman, 2015].

Las revisiones de software son procesos formales o informales donde se examina un producto de software en busca de defectos o desviaciones con respecto a los estándares y especificaciones. El proceso puede incluir una reunión entre los revisores, aunque también puede no requerir reuniones. Las revisiones se aplican a distintos artefactos que son generados durante todo el ciclo de desarrollo de software. Pueden ser documentos de requerimientos, modelos de diseño y código, entre otros. Por otra parte, la salida de un proceso de revisión consta de una lista de cuestiones que no fueron tenidas en cuenta o defectos a corregir y, adicionalmente, el estatus técnico del ítem de trabajo revisado [IEEE, 2008; Kan, 2002; Pressman, 2015; Burnstein, 2006].

Se pueden considerar distintos tipos de revisiones de software, desde una reunión informal donde se discuten aspectos técnicos, hasta una presentación de la arquitectura del software a cierta audiencia conformada por gerentes, técnicos y el cliente. La intención de realizar una revisión de software radica en encontrar defectos y problemas que pueden tener un impacto negativo en el software a ser desarrollado. Cuanto antes se descubra y corrija un defecto, menor será la probabilidad de que el mismo se propague hacia otros artefactos de trabajo y se amplifique, resultando en un esfuerzo significativo mayor para ser corregido [Pressman, 2015]. Si se realizan acciones para detectar defectos de forma temprana, entonces estos son más fáciles de detectar. El costo de repararlos es menor; el tiempo de retrabajo se reduce; la productividad mejora y existe menos impacto en el cliente [Burnstein, 2006].

A nivel general, las revisiones clásicas son definidas como la reunión de un grupo de revisión, cuyo propósito es evaluar un artefacto de software o un conjunto de artefactos de software [Burnstein, 2006]. Por su parte, la IEEE define las revisiones técnicas incluyendo ciertas características que coinciden con aspectos de las revisiones grupales de software. Algunos de ellos son: necesidad de asignación de roles (líder de la revisión, revisores técnicos, documentador), existencia de etapa de preparación de la revisión y una o varias reuniones entre los miembros de la revisión [IEEE, 2008].

Si bien existen variantes en los detalles de las revisiones, el proceso de revisión grupal suele estructurarse en tres fases: actividades previas a la revisión (planificación, preparación de la revisión, revisión individual), reunión de re-

visión (el autor del documento realiza un pasaje por el artefacto junto con el equipo de revisión, se consolidan todos los comentarios o defectos encontrados), actividades posteriores a la revisión (corrección por parte del autor de los problemas reportados; los revisores chequearán que se consideraron todos los comentarios y defectos reportados) [Sommerville, 2011].

Los revisores tienen como objetivo principal identificar componentes con problemas o componentes que requieran una mejora, identificar errores específicos o defectos en el artefacto de software y asegurar que el artefacto respeta los estándares de la organización. Otro objetivo de las revisiones es cumplir con un nivel informativo, comunicacional y educativo, mediante el cual los revisores aprenden sobre el contenido de los artefactos y, de esa forma, entienden el papel de su propio trabajo. Por otro lado, las revisiones, en general, representan o están asociados con hitos y dan soporte para la línea base de un artefacto de software [Burnstein, 2006].

Otro tipo de revisión, que ha sido utilizado desde los orígenes de la programación, es la revisión personal, también llamada revisión de escritorio. Antes de la existencia de herramientas interactivas, la revisión personal fue usada por los programadores para detectar defectos de forma de mejorar la calidad del código y la productividad. Cada individuo leía su propio código cuidadosamente, recorriendo el programa a través de casos de prueba, para detectar defectos de sintaxis o lógicos [Burnstein, 2006]. Asimismo, las revisiones individuales se pueden entender como aquellas que son llevadas a cabo por una sola persona, pero el artefacto bajo revisión puede haber sido creado por él mismo o por otro individuo.

Las revisiones informales pueden realizarse como una simple revisión de escritorio de un artefacto de software junto a un colega, como una reunión casual (de más de dos personas) para revisar un producto de trabajo o pueden encontrarse dentro de los aspectos relacionados con la revisión de la programación de a pares. En cuanto al rendimiento, una manera de mejorar la eficacia de una revisión informal es desarrollar un simple conjunto de *checklists* para cada artefacto importante generado por el equipo de software [Pressman, 2015].

Una *checklist*, utilizada en una revisión, contiene sentencias que ayudan al revisor a buscar errores. Cada sentencia está escrita para buscar cierta clase de defectos. Previo al desarrollo de una *checklist*, una organización debería identificar los tipos de defectos típicos cometidos en proyectos pasados, construir un esquema de clasificación para esos defectos y decidir categorías de impacto o severidad para los defectos [Burnstein, 2006].

Al utilizar las revisiones en software se obtienen importantes beneficios: se disminuye en costo y se mejora en calidad y productividad [Kan, 2002]. Por su

parte, Burnstein identifica varios beneficios. Primeramente, se obtiene software de alta calidad. En cuanto a la gestión, mejora en la productividad, mejora el control del proceso (hay mejor apego al cronograma del proyecto) y mayor conciencia de los problemas de calidad. Así mismo, resulta una herramienta educativa para los jóvenes practicantes y una oportunidad de identificar software reusable. Finalmente, reduce costo de mantenimiento, logra mayor satisfacción del cliente, promueve un plan de verificación más eficiente y una actitud más profesional dentro del equipo de desarrollo [Burnstein, 2006].

El diseño generalmente es revisado durante una o más etapas del ciclo de desarrollo. Es útil revisar la arquitectura de alto nivel al comienzo y luego revisar el diseño detallado. En todos los niveles del diseño, es importante verificar que el diseño es consistente con los requerimientos y que estos son cubiertos en su totalidad. Algunos elementos que deberían verificarse durante una revisión de diseño son: descripción de la técnica de diseño utilizada, evaluación de alternativas de diseño, calidad del modelo de arquitectura de alto nivel, descripción de las interfaces de los módulos, calidad de la interfaz de usuario, calidad de facilidades de ayuda para el usuario, identificación de secuencias de operación y criterios de ejecución, cobertura de los requerimientos funcionales, cobertura de todos los requerimientos de calidad, reusabilidad de componentes de diseño y verificabilidad [Burnstein, 2006].

Las revisiones de código son herramientas que sirven para encontrar defectos, así como también para evaluar la calidad del código. Algunas organizaciones exigen que las revisiones de código se realicen previo a la compilación y otras requieren que se haga con el código compilado. Las checklist para revisión de código pueden ser generales o específicas para un lenguaje de programación [Burnstein, 2006].

Tal como se mencionó, el proceso de revisión grupal de software incluye determinadas etapas en las que la revisión es llevada a cabo por un individuo, por ejemplo: el paso de chequeo individual dentro de un proceso de inspección de software [Fagan, 1999] o la etapa de preparación dentro de una revisión técnica [IEEE, 2008]. La revisión individual puede ser realizada por el autor del artefacto o por un par. Como se verá en la siguiente sección, en el PSP se llevan a cabo revisiones personales dirigidas por *checklist*. Además de los beneficios para el proceso de software, las revisiones personales tienen beneficios para el individuo, tales como la clarificación del pensamiento y la visualización del rendimiento propio con la posterior mejora del desempeño [Cohen, 2013; Burnstein, 2006].

2.6 PERSONAL SOFTWARE PROCESS (PSP)

El PSP es un proceso que incluye prácticas disciplinadas para la gestión y mejora de desempeño personal de los ingenieros de software. Entre esas prácticas se destacan algunas tareas vinculadas al aseguramiento de la calidad de los productos, como las revisiones personales de diseño detallado y código. Estas son realizadas por el autor del artefacto, quien revisa el código o modelo de diseño utilizando una lista de verificación (*checklist*) en busca de defectos [Pomeroy-Huff et al., 2009].

El diseño del PSP está basado en los siguientes principios de planificación y calidad [Humphrey, 2000]:

- cada ingeniero es diferente; para ser más efectivos, los ingenieros deben planificar su trabajo y deben basar su planificación en sus propios datos personales
- para mejorar constantemente su desempeño, los ingenieros deben usar procesos medibles y bien definidos
- para producir productos de alta calidad, los ingenieros deben sentir responsabilidad personal por la calidad de sus productos
- el costo de encontrar y corregir defectos, durante un proceso, es menor temprano que tarde
- es más eficiente prevenir defectos que buscarlos y corregirlos
- la manera correcta de realizar un trabajo es siempre la más rápida y la más barata

El proceso básico de PSP consta de tres fases: Planificación (en la que se genera un plan para realizar el trabajo); Desarrollo (donde se lleva a cabo el trabajo); Fase posterior a la finalización (se compara rendimiento real contra el plan, se registran los datos del proceso, se genera un reporte y documenta cualquier idea para mejorar el proceso). Dentro de la fase de Desarrollo se tienen las siguientes actividades: definir los requerimientos, diseñar el programa, revisar el diseño y corregir todos los defectos, escribir el programa, revisar el código y corregir todos los defectos, compilar y corregir todos los defectos, y finalmente, probar el programa y corregir todos los defectos [Pomeroy-Huff et al., 2009].

Las revisiones personales en el PSP resultan efectivas y eficientes para mejorar la calidad de un producto y la productividad individual [Pomeroy-Huff et al., 2009]. Varios trabajos presentan datos de las mejoras que introducen el

uso de las prácticas del PSP [Hayes y Over, 1997; Elminir et al., 2009; Khan, 2012; Ferguson et al., 1997; Paulk, 2010, 2006; Grazioli y Nichols, 2012; Vallespir y Nichols, 2016]. Como algunos ejemplos de buenos resultados destacamos: una mejora de la calidad del producto, medida como la cantidad de defectos que se encuentran en el *testing* unitario; y un incremento en los defectos encontrados antes de la compilación del código [Hayes y Over, 1997].

En el trabajo de Hayes y Over [1997] se encuentra que, al introducir las revisiones personales, la reducción de la densidad de defectos es de un factor de 3,7 en la fase de compilación y de 2,5 en la fase de *testing* unitario. Por su parte, Paulk [2006] encontró que la densidad de defectos en el *testing* unitario disminuye un 75 % luego de aplicar todas las prácticas del proceso PSP. Luego, Paulk [2010] confirmó esos hallazgos al encontrar que la calidad mejoraba en un 79 % observando las mismas etapas del PSP que en su trabajo anterior. El estudio de Elminir et al. [2009] encuentra una mejora estadística significativa en la estimación de tamaño, en la estimación de esfuerzo y en la calidad del producto. En particular, la densidad de defectos en la fase de compilación y de *testing* unitario decrece significativamente al introducir revisiones de código y diseño. Ferguson et al. [1997] analizan estudios en varias compañías donde se ven buenos resultados en el desempeño de los ingenieros que utilizan PSP, así como en la disminución de la cantidad de defectos. Por su parte, Rombach et al. [2008] presentan un estudio empírico orientado a confirmar la hipótesis de que el desarrollo de software disciplinado a nivel individual (representado por el PSP) es beneficioso para los desarrolladores. En particular, se demuestra que con la introducción de revisiones de diseño y de código la densidad de defectos en los programas, antes de la compilación y fase de *testing*, disminuye significativamente. Además, se agrega que una reducción en el total de defectos se traduce directamente en reducción de la cantidad de retrabajo para la organización que participa en desarrollo de software. Finalmente, en el estudio de Vallespir y Nichols [2016] se discuten resultados destacando que la revisión de diseño detecta un 50 % de los defectos y la revisión de código un 60 % de los defectos, sugiriendo que combinando las revisiones de diseño y de código junto al *testing* unitario se detecta un 80 % de los defectos.

2.7 REVISIONES INDIVIDUALES FUERA DEL PSP SOBRE CÓDIGO Y DISEÑO DETALLADO

Las revisiones individuales son aquellas revisiones llevadas a cabo por un individuo y que pueden ser realizadas tanto por el autor del artefacto (revisión personal) como por un par. Las revisiones individuales pueden realizarse en

ámbitos informales como revisiones de pares o revisiones de escritorio. También, se pueden encontrar como parte de la etapa de preparación individual en los procesos de inspección o de revisiones grupales.

El objeto de estudio de este trabajo son las revisiones individuales aplicadas a artefactos de diseño detallado y de código. Siendo, a su vez, utilizadas en marcos de trabajo donde no se aplica el Personal Software Process.

ANTECEDENTES: ESTUDIOS DE MAPEO SISTEMÁTICO

Este capítulo presenta la definición de estudios de mapeo sistemático y trabajos previos. Se divide en dos secciones. En la primera se introduce la ingeniería de software basada en evidencias y se describen aspectos importantes de los estudios de mapeo sistemático de la literatura, así como otras definiciones relacionadas. En la segunda sección se presentan antecedentes de la ejecución de mapeos sistemáticos en ingeniería de software.

3.1 ASPECTOS RELEVANTES DE LOS ESTUDIOS DE MAPEO SISTEMÁTICO

El paradigma basado en evidencia tiene su origen y auge en la medicina clínica a partir de la década del noventa. Luego, otras disciplinas, entre ellas la psiquiatría, sociología y educación, han adoptado enfoques similares. Recientemente, la ingeniería de software también ha incorporado la práctica de este paradigma. La ingeniería de software basada en evidencias (ISBE) tiene como objetivo proporcionar los medios para integrar la mejor evidencia actual de la investigación con la experiencia práctica y los valores humanos en el proceso de toma de decisiones sobre el desarrollo y mantenimiento de software [Kitchenham et al., 2004].

Según Kitchenham et al. [2004], existen cinco pasos necesarios para llevar a la práctica la ingeniería de software basada en evidencias. Ellos son:

1. convertir la necesidad de información en una pregunta a ser respondida
2. localizar la mejor evidencia para dar respuesta a esa pregunta
3. valorar críticamente la validez, el impacto y la aplicabilidad (utilidad en la práctica del desarrollo de software)
4. integrar la evaluación crítica con nuestra experiencia en ingeniería de software y con los valores y circunstancias de nuestros grupos de interés
5. evaluar la eficacia y eficiencia en la ejecución de los pasos 1-4 y buscar formas de mejorar dicha ejecución.

Las revisiones sistemáticas de literatura (RSL) se utilizan como metodologías para la agregación de la evidencia empírica. Una RSL es un estudio que,

mediante un procedimiento riguroso, busca evaluar e interpretar toda investigación disponible relevante con el fin de responder una pregunta de investigación. El proceso está dirigido por una pregunta de investigación específica que puede ser respondida mediante investigación empírica [Kitchenham y Charters, 2007]. A su vez, las RSL pueden ser subclasificadas de acuerdo a si los datos están sintetizados de forma cualitativa o cuantitativa. Esa forma determina cómo se realizará la extracción de datos y cómo se organizará la síntesis. Un ejemplo de función de una RSL, en ingeniería de software, es establecer si ciertas técnicas o prácticas de trabajo son mejores que otras, y bajo qué condiciones es eso cierto [Kitchenham et al., 2015].

Por su parte, los estudios de mapeo, también conocidos como estudios de alcance, son un tipo de revisión sistemática de literatura, pero a diferencia de las RSL su propósito es encontrar y clasificar estudios primarios dentro de un tópico específico. Asimismo, este tipo de estudios identifica huecos en investigaciones actuales, lo cual puede sugerir determinados trabajos posteriores. Por lo tanto, proporcionan un marco de referencia para desarrollar adecuadamente las actividades de investigaciones futuras. Están definidos para proveer una visión amplia de un área de interés, establecer si existe evidencia de investigación en un tema y proveer indicadores de cantidad de la evidencia [Budgen et al., 2008].

Los mapeos sistemáticos y las revisiones sistemáticas de literatura utilizan un mismo marco metodológico, el cual se compone de tres fases: planificación, ejecución y reporte. Sin embargo, debido a sus características puede ser conveniente realizar una RSL o un estudio de mapeo según el caso. Por ejemplo, si durante el examen inicial de un dominio, previo a la puesta en marcha de una revisión sistemática, se descubre que existe muy poca evidencia o que el tema resulta demasiado amplio, entonces un estudio de mapeo sistemático puede ser más apropiado que una revisión sistemática. En un mapeo sistemático la pregunta de investigación es más general que en una RSL. Esta está relacionada con tendencias de investigación, con los subtópicos abordados, con los métodos empíricos utilizados y con los subtópicos abarcados por suficientes estudios empíricos para realizar una revisión sistemática más detallada. El proceso de búsqueda de un estudio de mapeo está definido por el área del tópico, mientras que en las RSL está definido por la pregunta de investigación. Por último, los resultados en un mapeo presentan un conjunto de estudios relacionados con un tópico y la cantidad de estudios en distintas categorías, a la vez que en una RSL los resultados contestan una pregunta de investigación específica [Kitchenham y Charters, 2007; Kitchenham et al., 2010].

Aunque en el extremo, los estudios de mapeo y las revisiones sistemáticas tienen objetivos bastante diferentes, en algunos casos el límite puede ser difuso o incluso existir una superposición. Por ejemplo, algunas RSL incluyen un esquema de clasificación para organizar la evidencia relevante junto con un análisis más detallado dentro de cada categoría. En el mismo sentido, la literatura ha encontrado estudios que han sido clasificados como RSL cuando la correcta clasificación podría haber sido un estudio de mapeo según los criterios definidos por las principales guías. En el cuadro 1 se resumen las diferencias entre RSL y mapeo sistemático [Kitchenham et al., 2015, 2010].

	RSL	Estudio de mapeo
Objetivo	Busca contestar una pregunta específica	Busca recolectar y clasificar estudios dentro de un tópico
Pregunta de investigación	Específica, relacionada con los resultados de los estudios empíricos	Amplia, relacionada con tendencias de investigación
Proceso de búsqueda	Definido por la pregunta de investigación	Definido por el área temática
Estrategia de búsqueda	Extremadamente estricta	No es muy estricta
Evaluación de calidad	Esencial para asegurar la calidad de la evidencia	Es opcional
Extracción de datos	Específica para contestar la pregunta de investigación	Busca clasificar los estudios
Etapas de análisis	Profunda y puede incluir meta-análisis	Tiende a resumir la información que contesta las preguntas
Resultados	Respuestas a preguntas de investigación específicas y calificadores	Conjunto de estudios relacionados con el tópico y su distribución en varias categorías

Cuadro 1: Diferencias entre RSL y estudio de mapeo

El uso de los estudios de mapeo sistemático y de las RSL provee un enfoque imparcial, objetivo y sistemático para responder una o varias preguntas de investigación mediante la búsqueda de todos los resultados relevantes obtenidos a partir de estudios empíricos primarios [Budgen et al., 2008]. Se entiende por estudio primario aquel en el cual se hacen mediciones directamente sobre los objetos de interés, ya sea a través de encuestas, experimentos, estudios de caso, etc. [Budgen, 2007]. Por lo tanto, en los estudios empíricos primarios se utilizan métodos que pertenecen a la ingeniería de software empírica, por lo que, antes de continuar con las prácticas de investigación basada en evidencia, es necesario introducir conceptos de la ingeniería de software empírica.

La ingeniería de software empírica se basa en la experimentación para corresponder hechos con las especulaciones, suposiciones y creencias sobre la cons-

trucción de software [Juristo y Moreno, 2013]. Existen dos enfoques diferentes en investigación empírica: el enfoque cualitativo y el cuantitativo. El enfoque cualitativo está relacionado con estudiar objetos en su contexto natural y con interpretar un fenómeno a partir de las explicaciones de las personas. Por su parte, el enfoque cuantitativo se corresponde principalmente con la cuantificación de una relación entre dos o más grupos. El objetivo es identificar una relación de causa-efecto [Wohlin et al., 2012].

Hay tres tipos principales de técnicas o estrategias para la investigación empírica: las encuestas, los estudios de casos y los experimentos. Las encuestas son, en general, realizadas en retrospectiva. Por ejemplo, cuando una herramienta o técnica ha sido utilizada por un tiempo. En cuanto a la recolección de datos, los medios principales son los cuestionarios o entrevistas, los cuales son realizados sobre una muestra representativa de la población a ser estudiada. Los resultados de la encuesta son analizados y luego son generalizados al resto de la población. Por su parte, los estudios de casos son estudios observacionales que se utilizan para monitorear proyectos, actividades o tareas. Normalmente su objetivo es el seguimiento de un atributo específico o el establecimiento de relaciones entre diferentes atributos. Por otro lado, el experimento es una investigación formal que, generalmente, es realizada en un ambiente de laboratorio de forma de disponer de un alto nivel de control. El objetivo de un experimento es manipular una o más variables y controlar el resto [Wohlin et al., 2012].

Retornando al paradigma de ISBE, se puede clasificar el mapeo sistemático, así como a las RSL, como tipos de estudio secundario. Un estudio secundario es aquel que no genera conocimiento de mediciones directas, sino que analiza un conjunto de estudios primarios y por lo general trata de agregar los resultados de esos estudios, con el fin de proporcionar una evidencia más fuerte acerca de un fenómeno particular [Budgen, 2007].

Por su parte, existe otro tipo de revisión que complementa las revisiones sistemáticas de literatura: las revisiones terciarias. En un dominio donde ya existe un número de revisiones sistemáticas, se puede llevar a cabo una revisión terciaria, entendida como una revisión sistemática de revisiones sistemáticas, con el fin de responder preguntas de investigación más amplias. La revisión terciaria es un estudio terciario, ya que consiste en un estudio secundario que utiliza como entrada las salidas de otros estudios secundarios [Kitchenham y Charters, 2007; Budgen, 2007].

Una de las grandes ventajas de los mapeos sistemáticos y de las revisiones sistemáticas de literatura es que permiten recolectar y sintetizar evidencia de distintas fuentes. A diferencia de las revisiones de literatura convencionales, los mapeos y las revisiones sistemáticas comienzan con la definición de un

protocolo, el cual especifica la pregunta de investigación a ser contestada y los métodos que se usarán para llevar a cabo el mapeo. Otra característica que los distingue de las revisiones convencionales es que están basados en una estrategia de búsqueda definida, con el objetivo de detectar tanta literatura relevante como sea posible. Además, documentan la estrategia de búsqueda y de esa manera los lectores pueden evaluar la rigurosidad, la integridad y la repetibilidad del proceso. Para evaluar cada potencial estudio primario se requiere de criterios explícitos de inclusión y exclusión. Asimismo, se especifica la información a ser obtenida de cada estudio primario, incluyendo, en el caso en que se realice, el criterio de calidad con el que será evaluado cada uno [Kitchenham y Charters, 2007].

Como ya se comentó, un mapeo sistemático se conduce a través de un proceso que tiene tres fases: planificación, ejecución del mapeo y reporte de los resultados. Cada fase tiene un conjunto de etapas y actividades asociadas. Varias de estas etapas son iterativas, es decir que se refinan a medida que se desarrolla el protocolo. [Kitchenham y Charters, 2007; Biolchini et al., 2005; Budgen et al., 2008]. En el cuadro 2 se muestran las fases mencionadas junto con sus respectivas etapas.

Fase	Etapas
Planificación	Identificación de la necesidad del mapeo Encargo del mapeo (opcional) Especificación de la(s) pregunta(s) de investigación Desarrollo de un protocolo Evaluación del protocolo (opcional)
Realización	Identificación de los artículos Selección de los estudios primarios Evaluación de la calidad del estudio (opcional) Extracción de datos y monitoreo Síntesis de los datos
Información	Especificación de los mecanismos de diseminación Formateo del informe principal Evaluación del informe (opcional)

Cuadro 2: Fases de un mapeo sistemático

La primera fase es la de planificación de la revisión. Las actividades más importantes previas a la revisión incluyen definir las preguntas de investigación a ser abordadas por el mapeo sistemático y definir un protocolo que establezca

los procedimientos básicos de la revisión. La necesidad de un estudio de mapeo sistemático deriva de la exigencia de los investigadores para resumir la información que existe acerca de un determinado fenómeno, de manera exhaustiva e imparcial. Esta exigencia puede surgir de la intención de extraer conclusiones más generales que las conclusiones obtenidas en los estudios individuales, o puede ser motivada como preludeo para fomentar actividades de investigación mediante la identificación de huecos o mediante la identificación de concentración de estudios primarios en un tema en particular.

El objetivo principal de un estudio de mapeo sistemático es proveer un panorama general de un área de investigación e identificar la cantidad y tipos de trabajos de investigación, así como los resultados disponibles en ellos. Comúnmente, se pretende realizar un mapeo de las frecuencias de publicaciones a lo largo del tiempo para visualizar tendencias. Un objetivo secundario es identificar los foros en donde se publican las investigaciones en el área. Estos objetivos se reflejan en las preguntas de investigación, las cuales deberán ser contestadas por los estudios primarios identificados a través de los elementos extraídos de ellos [Petersen et al., 2008].

La pregunta puede estructurarse considerando distintas variables, tales como la población, intervención, comparación, resultados y contexto. En experimentos de ingeniería de software la población puede constituir un rol específico (*tester*, gerente), una categoría de ingeniero de software (novato, con mucha experiencia), un área de aplicación o un grupo industrial (empresas de telecomunicaciones, pequeñas empresas). La intervención refiere a la metodología, herramienta o tecnología que se aplica. La comparación está vinculada con la metodología, herramienta o tecnología con la que se compara la intervención. Los resultados están relacionados con factores de importancia para los practicantes, como pueden ser reducción de costo, reducción de tiempo de salida al mercado, etc. Por último, el contexto refiere a donde tiene lugar la comparación (academia, industria), a qué participantes forman parte del estudio (estudiantes, profesionales) y al tipo de tareas que son realizadas (escala pequeña, gran escala).

Asimismo, un protocolo de revisión especifica los métodos que serán usados para llevar a cabo un mapeo sistemático específico. Los componentes de un protocolo incluyen todos los elementos de la revisión, más información de planificación. Ellos son:

- Introducción, antecedentes (Background) y justificación de la investigación.
- Las preguntas de investigación que el mapeo intenta responder.

- La estrategia que será usada para buscar los estudios primarios, incluyendo los términos de búsqueda y recursos donde se realizará la búsqueda. Los recursos incluyen librerías digitales, revistas específicas y actas de congresos.
- El criterio de selección de estudios que se usa para determinar cuáles estudios son incluidos o excluidos del mapeo sistemático.
- Los procedimientos de selección de estudios. El protocolo debe describir cómo se aplicará el criterio de selección, por ejemplo, cuántos asesores evaluarán cada estudio primario y cómo se resolverán los desacuerdos entre los evaluadores.
- *Checklists* y procedimientos de la evaluación de calidad de los estudios (En caso que se realice).
- Estrategia de extracción de datos, en las que se define cómo se obtendrá la información requerida de cada estudio primario.
- Síntesis de los datos extraídos. Es decir, definición de la estrategia de síntesis.
- Estrategia de diseminación.
- Cronograma del estudio de mapeo.

Debido a que el protocolo es un elemento crítico de cualquier revisión o mapeo sistemático, los investigadores deben acordar un procedimiento para evaluarlo.

La segunda fase del proceso trata de la ejecución o realización de la revisión o mapeo. Los estudios primarios pueden ser identificados mediante cadenas de búsquedas en las bases de datos científicas o mediante la consulta manual de los recursos relevantes. Las búsquedas deben aplicarse sobre la misma metadato (por ejemplo: título, resumen) para todos los artículos en todas las fuentes. De todas maneras, la cadena de búsqueda debe ser adaptada para ejecutarse en las distintas bibliotecas digitales.

El proceso de realización de un estudio de mapeo sistemático debe ser tan transparente y repetible como sea posible. La revisión debe ser documentada con el suficiente detalle para que los lectores puedan evaluar qué tan minuciosa fue la búsqueda. Así, la búsqueda debe ser documentada tal como ocurrió y los cambios deben anotarse y justificarse; los resultados de la búsqueda deben ser guardados para un posible reanálisis.

La revisión se lleva a cabo siguiendo y adaptando los lineamientos que fueron señalados en el protocolo durante la fase de planificación. Es decir, es preciso respetar los criterios de selección de estudios y de evaluación de calidad, aplicando la estrategia de extracción de información definida y realizando la síntesis de los datos de la forma pautada. Por esto, a continuación se describen dichas actividades.

Los criterios de selección de estudios tienen como objetivo identificar aquellos estudios primarios que proveen evidencia directa acerca de la pregunta de investigación. Los criterios de inclusión y exclusión deben estar basados en la pregunta de investigación. A su vez, el proceso de selección de estudios tiene varias etapas. Inicialmente, el criterio de selección debería ser interpretado libremente. Es decir, a menos que un artículo identificado por la búsqueda electrónica pueda ser claramente descartado debido a título o resumen, debería mantenerse una copia del texto completo. No obstante, se sugiere, además, revisar las conclusiones. El siguiente paso consiste en aplicar el criterio de inclusión/exclusión basado en cuestiones prácticas como lenguaje, fuente, autores, participantes, diseño de la investigación, método, fecha de publicación, etc.

En los mapeos sistemáticos la evaluación de la calidad de los estudios primarios no es obligatoria. En caso de realizarse, deben considerarse ciertos aspectos tales como las características a evaluar y la forma de medirlas.

Las características medidas por la evaluación de la calidad abarcan aquellos aspectos vinculados con el planteo del experimento, la forma en que se llevó a cabo y los resultados alcanzados, por ejemplo:

- objetivos (son claros y están establecidos; existe una motivación; se plantearon preguntas de investigación; se plantean hipótesis);
- contexto de la investigación correctamente explicado y apropiado;
- diseño del experimento, justificado y descrito;
- recolección de datos;
- procedimientos de análisis de datos;
- conclusiones.

Al medir dichas características, en general, se definen criterios de puntajes y se consideran categorías de puntajes totales.

La evaluación de la calidad presenta algunas limitaciones. A menudo el informe de los estudios primarios es pobre, por lo que puede resultar muy difícil determinar la forma de evaluar un criterio de calidad. Además, es tentador suponer que si algo no se encuentra reportado, es porque no se hizo. No obstante,

los investigadores deberían intentar obtener más información por parte de los autores del estudio.

Una de las tareas principales de un estudio de mapeo sistemático consiste en obtener un conjunto de estudios relacionados con un determinado tópico y clasificar dichos estudios en categorías definidas [Kitchenham et al., 2010]. La actividad de categorización puede requerir el uso de diferentes esquemas. El esquema, a su vez, debe reflejar características específicas de los estudios primarios.

Para seleccionar las facetas que constituirán el esquema de clasificación se suele utilizar un esquema existente. Por ejemplo, se puede clasificar los estudios encontrados en términos del método de investigación empleado por cada estudio primario, mediante un conjunto de categorías como experimento, quasi-experimento, encuesta, observacional y estudio de caso.

Otra manera de construir el esquema es derivar las categorías para una determinada característica a partir del análisis del conjunto de estudios primarios e identificar patrones que permitan una agrupación [Kitchenham et al., 2015]. Por ejemplo, en Petersen et al. [2008], se utiliza la construcción de un esquema de clasificación a partir de los términos clave de los resúmenes de los artículos hallados.

Luego de haber aplicado los criterios de selección y haber obtenido el conjunto de estudios primarios relevantes se debe disponer de una estrategia de extracción de datos. El objetivo de la etapa de extracción de datos es diseñar formularios para registrar con precisión la información obtenida de los estudios primarios. Los formularios de extracción de datos deben estar diseñados para recolectar toda la información necesaria que permita afrontar las preguntas del mapeo y los criterios de calidad. Además de incluir la respuesta a la pregunta del mapeo sistemático y el criterio de evaluación de calidad, el formulario de extracción de datos deberá proveer información básica incluyendo nombre del revisor, fecha de la extracción de datos, título, autores, revista, detalles de la publicación y lugar para notas adicionales.

Por otro lado, la síntesis de la información abarca la recolección y resumen de los resultados de los estudios primarios incluidos. Si bien la síntesis podría ser descriptiva (no cuantitativa), esta puede complementarse con un resumen cuantitativo. La información a incluir como síntesis cuantitativa abarca la agrupación por pregunta de investigación, el número de los artículos por año, la tendencia en el tiempo y la agrupación por categorías.

En cuanto a la fase de presentación del informe de la revisión, esta fase final de un mapeo sistemático comprende, por un lado, la redacción de los

resultados del mapeo, y por otro, implica la disseminación de los resultados a partes potencialmente interesadas.

Como información adicional, esta tesis incluye en el Apéndice B el reporte técnico "*Conceptos de ingeniería de software basada en evidencias*". Dicho RT presenta la ingeniería de software basada en evidencias y un resumen de los métodos más utilizados dentro de ella, como son las revisiones sistemáticas de literatura, los estudios de mapeo sistemático y las revisiones terciarias.

3.2 TRABAJOS PREVIOS DE ESTUDIOS DE MAPEO SISTEMÁTICO EN INGENIERÍA DE SOFTWARE

Tal como fue mencionado, las técnicas de revisiones y mapeos sistemáticos surgen en otras disciplinas, como la medicina, y han sido adoptadas por la ingeniería de software. Si bien el uso de esta herramienta carece de vasta experiencia, es posible encontrar varios ejemplos de estudios de mapeo dentro del contexto de la ingeniería de software [Dybå y Dingsøy, 2008; Fernández-Sáez et al., 2013; Mujtaba et al., 2008].

Por otra parte, existen algunos estudios que analizan el uso de mapeos sistemáticos en ingeniería de software. Dos ejemplos de ello son, por un lado, el trabajo de Budgen donde se examina la experiencia y resultados de seis mapeos sistemáticos [Budgen et al., 2008]. Por otro lado, el trabajo de Kitchenham también realiza un análisis de características de los mapeos en ingeniería de software desde el punto de vista de ventajas, desventajas y características necesarias para cumplir con sus expectativas [Kitchenham et al., 2010].

Acerca del primero, se pueden considerar algunos aportes de dicho trabajo. El estudio menciona que el uso de estas técnicas por parte de estudiantes puede ser conveniente, dado que la mayoría de los proyectos de estudiantes en ingeniería de software, tanto de pregrado como de postgrado, ya incluyen la revisión de literatura. Además, los autores reportan buenas experiencias motivando estudiantes a realizar las revisiones de forma sistemática. Sin embargo, es preciso reconocer algunas precauciones. Un mapeo sistemático consume mucho tiempo, es decir, se extiende mucho más allá del tiempo que normalmente se utiliza en la lectura del estado del arte. En compensación, un estudio de mapeo puede proveer una oportunidad de publicación por sí mismo. Por su parte, la clasificación presenta una tarea desafiante para los estudiantes. A su vez, las guías advierten que la extracción de datos debe ser realizada usando dos o más analistas independientes, o por lo menos un analista y un revisor; lo cual no resulta práctico para los proyectos de estudiantes. Incluso investi-

gadores experimentados pueden diferir sustancialmente cuando clasifican un trabajo.

Asimismo, los autores destacan en ese trabajo dos problemas recurrentes en el conjunto de estudios analizados. Uno de estos problemas, es acerca de cómo clasificar y evaluar los estudios primarios. Esto constituye un desafío para aquellos investigadores sin experiencia así como para aquellos más experimentados pero que no están familiarizados con un campo particular. Así, mientras que un mapeo sistemático puede proporcionar un excelente punto de partida para un proyecto estudiantil, es solo probable que lo haga en la medida en que el problema de la clasificación sea razonablemente manejable.

Por otro lado, el segundo tema recurrente es la pobre calidad de los informes de los estudios empíricos en ingeniería de software, tanto en artículos como en resúmenes. Los autores sugieren basarse en guías y experiencias anteriores para mejorar los trabajos, por ejemplo, utilizar resúmenes estructurados, los cuales incluso ayudarían a la extracción de datos.

Los autores concluyen que los mapeos sistemáticos son particularmente útiles para tener una mejor comprensión de la evidencia disponible dentro de un dominio. Además, resultan útiles para identificar las áreas donde hay información suficiente para que una RSL sea eficaz, así como aquellas áreas donde son necesarios más estudios primarios. En cuanto a la educación, los autores entienden que los mapeos sistemáticos proporcionan un valioso punto de partida para los estudiantes de doctorado que necesitan organizar y entender el trabajo de investigación en un dominio específico. A largo plazo, a medida que más estudios de mapeo se publiquen, podrían llegar a cambiar la forma en que se investiga en ingeniería de software.

Dentro del análisis realizado por el segundo trabajo, podemos destacar algunos conceptos relativos a las ventajas, desventajas y al valor de los estudios de mapeo en investigación en ingeniería de software. En cuanto a las ventajas, se señala que el uso de estudios de mapeo ahorra tiempo de los trabajos posteriores, mientras que provee una comprensión de la literatura. En el caso de una extensión de un estudio de mapeo, el protocolo construido puede ser reutilizado. Si los mismos investigadores llevan a cabo el estudio, tendrán experiencia en el procedimiento; además, se pueden estudiar tendencias a lo largo del tiempo; y se dispone de un conjunto de estudios conocidos que puede servir para validar cadenas de búsqueda utilizadas en búsquedas automatizadas. También, el estudio de mapeo puede identificar *clusters* de estudios de investigación donde es adecuado realizar estudios más detallados, así como identificar la necesidad de más estudios primarios. Por último, los propios investigadores obtienen un muy buen panorama de la literatura.

Como potenciales desventajas se destaca que si el proceso de búsqueda utilizado por el estudio de mapeo original fue restringido, es probable que existan limitaciones que tendrán un impacto negativo en actividades de investigación posteriores. Un problema particular que puede ocurrir en esta situación es no encontrar estudios primarios, ya sea debido a limitaciones en la cadena de búsqueda o por falta de propagar la búsqueda hacia estudios adicionales. Además, el esquema de clasificación puede ser demasiado simplista o usarse incorrectamente. Por otro lado, si la actividad de investigación es una extensión de un estudio de mapeo anterior, es importante considerar los cambios en el proceso que podrían reducir la capacidad comparación entre el mapeo inicial con los siguientes estudios de mapeos. Asimismo, los estudios de mapeo pueden estimar por debajo el número de estudios primarios que utilizarán las RSL posteriores. Esto ocurre cuando se presentan múltiples estudios primarios en un solo artículo y es de particular importancia si un estudio de mapeo se utiliza como un medio para ayudar a la estimación de recursos para futuras RSL. Es igualmente importante constatar que la política para tratar los trabajos duplicados del mismo estudio es apropiada. Finalmente, si se retrasan las actividades posteriores, incluso un estudio de mapeo de alta calidad deberá ser actualizado antes de iniciar el seguimiento de la investigación.

Por último, [Kitchenham et al. \[2010\]](#) contestan dos preguntas. La primera indaga los aspectos que hacen a un mapeo adecuado para respaldar futuras investigaciones. Como respuesta enuncia que un estudio de mapeo no puede dar soporte fácilmente a otras actividades de investigación, excepto que sea llevada a cabo por los investigadores originales, a menos que se citen todas las referencias y se reporte toda la información de clasificación para cada estudio. El estudio de mapeo debe ser de alta calidad. Es decir, debe emplearse un proceso de búsqueda rigurosa incluyendo búsquedas automatizadas, búsquedas manuales de fuentes críticas, búsqueda por propagación a partir de referencias de estudios primarios y la comunicación directa con los investigadores relevantes y grupos de investigación. Además, debe tener un sistema de clasificación fiable y bien definido. Si se desconoce si el estudio es de alta calidad o se ha llevado a cabo en un rango de tiempo restringido, debe ser revisado críticamente antes considerarlo adecuado para dar soporte a una futura investigación. Cuando no es adecuado para una futura investigación, puede servir como base para realizar un estudio de mapeo más detallado.

La segunda presunta busca responder acerca de la forma en que los estudios de mapeo sistemático de alta calidad contribuyen a futuras investigaciones. Se señala que sirven como línea base contra la que pueden rastrearse las tendencias de investigación a lo largo del tiempo; como justificación para futuros es-

tudios primarios cuando hay pocos (o ninguno) estudios primarios pertinentes; como forma de identificar literatura relevante para la sección de trabajos relacionados de un estudio primario; como un recurso de educación y como un conjunto de referencias conocidas que otro mapeo o RSL puede utilizar para validar sus propias búsquedas.

REVISIONES INDIVIDUALES DE SOFTWARE FUERA DEL PSP: MAPEO SISTEMÁTICO

En este capítulo se presentan las características generales del proceso de mapeo sistemático ejecutado en este trabajo y los resultados obtenidos. El objetivo principal es identificar aspectos del uso de las revisiones individuales de software sobre artefactos de código y diseño detallado, en marcos donde no se utiliza el PSP.

En las siguientes secciones se describen la fase de planificación, la fase de ejecución y, por último, se presentan los resultados.

4.1 PLANIFICACIÓN DEL MAPEO SISTEMÁTICO

En la fase planificación se identificó la necesidad de realizar el estudio, se especificaron las preguntas de investigación y se construyó el protocolo del mapeo. En cuanto a la necesidad de realizar el estudio, se destacan dos aspectos. El primero, es el conocido aporte de las revisiones de software dentro del proceso de desarrollo de software, en cuanto a obtención de productos de software de calidad. En segundo lugar, los buenos resultados encontrados dentro del PSP para las revisiones personales de software. Aún así, existen otros estudios que identifican una baja adopción de técnicas de revisión en la industria del software. Por esto, se encuentra necesario reunir información existente acerca del uso de las revisiones individuales de software, en particular donde no se aplique el PSP. Finalmente, si se considera que no se conoce la investigación existente en este tópico y que se pretende establecer un punto de referencia para futuros trabajos de investigación, se decide conducir un estudio de mapeo como estrategia para lograr reunir la información relevante de forma sistemática.

Para la construcción del protocolo de mapeo sistemático se siguió la guía de [Kitchenham y Charters \[2007\]](#) y se consideró el proceso propuesto por [Petersen et al. \[2008\]](#). Por un lado, [Kitchenham y Charters \[2007\]](#) proveen una completa guía de alto nivel acerca de la realización de revisiones sistemáticas en ingeniería de software, a partir de las experiencias del uso de revisiones sistemáticas en medicina. Por su parte, [Petersen et al. \[2008\]](#), quienes, además, toman como referencia esta guía, discuten las diferencias entre revisiones sistemáticas y

estudios de mapeo; al mismo tiempo que presentan un proceso detallado de mapeo sistemático para ingeniería de software.

En cuanto al proceso de [Petersen et al. \[2008\]](#), este consta de 5 pasos, donde cada paso genera una salida. Los pasos y las correspondientes salidas se describen en el cuadro 3. A diferencia del proceso mencionado, en el presente trabajo no se construye un esquema de clasificación a partir de palabras claves de los resúmenes de los estudios primarios sino que los estudios son clasificados según aspectos de interés los cuales se señalan dentro de la descripción de la estrategia de extracción de datos del mapeo.

Paso	Definición de las preguntas de investigación	Ejecución de la búsqueda	Análisis de los artículos	<i>Keywording</i> a partir de resúmenes	Extracción de datos y proceso de mapeo
Salida	Alcance de la revisión	Todos los estudios primarios	Estudios primarios relevantes	Esquema de clasificación	Mapeo sistemático

Cuadro 3: Pasos y salidas del proceso de mapeo

En el protocolo de un mapeo sistemático se especifican los objetivos, las preguntas de investigación, estrategia de búsqueda, los criterios de inclusión y exclusión, la extracción de la información y método de síntesis de los datos.

Se llevó a cabo una etapa de evaluación del protocolo por parte de un experto en revisiones de la literatura, para determinar si el protocolo fue definido de forma adecuada y permite realizar una ejecución correcta. Adicionalmente, se realizó una revisión del protocolo entre pares aplicando una lista de verificación.

Por otro lado, el protocolo fue calibrado mediante el análisis de un conjunto de los primeros artículos devueltos por búsquedas en distintas fuentes. Se observó si el conjunto de artículos obtenido era coherente con lo esperado por el protocolo y en dirección a las preguntas de investigación.

El protocolo final de este estudio de mapeo sistemático se incluye en el Apéndice A.

En la guía de [Kitchenham y Charters \[2007\]](#), se hace referencia a recomendaciones acerca de la estructura de la pregunta de investigación. Por ejemplo, el criterio PICOC tiene 5 puntos de vista: Población, Intervención, Comparación, Resultados y Contexto. En este estudio se toma en cuenta dicho criterio y los términos son interpretados de la siguiente manera:

- Población: Artefactos de software (código y diseño detallado)

- Intervención: Revisión individual de software
- Comparación: No se compara contra otras intervenciones
- Resultados: No hay restricción, ya que se quiere obtener toda la información disponible y, en particular, experiencias del uso de la intervención y resultados de efectividad o costo.
- Contexto: No hay restricciones (Academia o industria)

4.1.1 Preguntas de investigación

Las preguntas de investigación se definen de acuerdo con las variables de respuesta a ser analizadas. Estas variables de respuesta son aquellas relacionadas con la evidencia del uso de las revisiones individuales de software. En particular, pueden describirse de la siguiente manera: motivos que llevan a una organización a usar las técnicas de revisiones individuales; motivos por los que las revisiones individuales no sean adoptadas; evidencia acerca de efectividad y costo de ejecutar las revisiones individuales; e identificación de factores que tengan impacto en el rendimiento de las revisiones individuales de software. Por lo que las preguntas de investigación definidas son las enumeradas a continuación:

RQ.1. ¿Por qué motivos se decide realizar revisiones individuales de código y diseño detallado?

Justificación: Se desea conocer los factores que influyen en un organización para decidir utilizar revisiones individuales de software sobre código y diseño detallado.

RQ.2. ¿Cuáles son los motivos por los que las revisiones individuales sobre código y diseño detallado no son adoptadas en la industria del software?

Justificación: Se desea encontrar los motivos por los que una organización decide no utilizar revisiones individuales de software sobre código y diseño detallado.

RQ.3. ¿Existe evidencia empírica acerca de la efectividad y/o del costo de las revisiones individuales de código y diseño detallado? Un ejemplo de medida de efectividad es el porcentaje de defectos detectados sobre defectos totales. Un ejemplo de costo es el tiempo empleado en llevar a cabo la revisión por cada mil líneas de código (hs/Kloc).

Justificación: Se quiere saber qué tan efectivas y costosas son las revisiones individuales de software sobre código y diseño detallado. Y conocer si existen

resultados comparables e incluso mejores que otras técnicas, en aquellos casos que sea factible la comparación.

RQ.4. ¿Qué factores influyen en la efectividad y el costo de aplicar revisiones individuales sobre código y diseño detallado?

Justificación: Es deseable conocer qué factores pueden influir en los resultados de efectividad y costo de la aplicación de las revisiones individuales. (Algunos factores posibles podrían ser: experiencia, programa bajo revisión, diseño detallado bajo revisión, motivación, contexto).

Cabe aclarar que si bien estas preguntas definidas pueden resultar muy específicas para dirigir un estudio de mapeo sistemático, en la ejecución del proceso no se aplica la rigurosidad de análisis que requiere una RSL. Estas preguntas se utilizaron para enmarcar el contexto de interés. De todas maneras, debido a que algunas de ellas refieren a resultados se debió leer en profundidad los estudios, pero sin realizar una agregación de los resultados.

4.1.2 *Estrategia y criterios de búsqueda*

Con el fin de llevar a cabo la búsqueda de los estudios primarios fue utilizada una estrategia de búsqueda constituida por una cadena de búsqueda y los recursos electrónicos en donde realizar la misma.

Para calibrar el protocolo y para realizar las pruebas se utilizaron varias bases de datos electrónicas, tales como ACM Digital, IEEE Xplore, SCOPUS y SpringerLink. De todas maneras, como este trabajo no busca ser exhaustivo sino desarrollar un protocolo que pueda aplicarse en cualquier buscador digital, la base de datos electrónica seleccionada para la extracción y análisis de información es IEEE Xplore. De esta forma se evalúa el método de mapeo sistemático en sí mismo y el protocolo de este mapeo en particular. Se decidió utilizar esta única biblioteca digital para la conducción del mapeo, debido a que, junto con ACM, cubren las revistas y actas de conferencias más importantes en la ingeniería de software y, por otro lado, IEEE es la más utilizada dentro de las revisiones en ingeniería de software [Kitchenham et al., 2015]. Además, se consideró minimizar el esfuerzo y tiempo requerido para llevar a cabo el estudio, y el hecho de que esta investigación constituye un trabajo inicial donde se busca también explorar el mapeo sistemático como metodología de investigación.

Teniendo en cuenta las preguntas de investigación formuladas anteriormente, se definieron las palabras claves y sus respectivos sinónimos y palabras alternativas, considerando las siguientes características:

- ¿Quién aplica el método, es decir quién hace la revisión?

- ¿A qué artefacto se le aplica la revisión?
- Técnica o herramienta relacionada con la utilización de revisiones individuales.
- Producto desarrollado: Software

En el cuadro 4 se presentan los términos identificados mediante la descomposición en aspectos.

Quién	Qué artefacto	Técnica	Producto desarrollado
Personal	Code	Review	Software
Individual	Design	Inspection	
Author	Program	CheckList	
Developer	Diagram	Defect Detection	
Single	Document	Reading Technique	
Reviewer	Artefact	Code Reading	
		Error detection	
		Code evaluation	
		Find fault	
		Fault detection	
		Detecting defect	
		Design Defect	
		Find defect	
		Examination	

Cuadro 4: Términos a partir de descomposición en aspectos

De manera de poder realizar búsquedas automáticas, se compone la siguiente cadena de búsqueda, obtenida a partir de los términos de la descomposición en aspectos.

(personal OR individual OR author OR developer OR single OR reviewer) AND (review OR inspection OR checklist OR "defect detection" OR "reading technique" OR "code reading" OR "error detection" OR "code evaluation" OR "find fault" OR "fault detection" OR "detecting defect" OR "design defect" OR "find defect" OR examination) AND (code OR design OR program OR diagram

OR document OR artefact) AND software

Cabe mencionar que la cadena debe adaptarse a la sintaxis y a las reglas permitidas de cada motor de búsqueda utilizado.

La cadena de búsqueda se ejecutó sobre título, resumen y palabras clave y no sobre el texto completo del artículo. El período consultado fue de estudios publicados desde el primer año disponible hasta el 19 de setiembre de 2014.

4.1.3 *Criterio de selección de estudios*

Los términos y criterios de selección de estudios tienen como objetivo identificar aquellos estudios primarios relevantes dentro del conjunto total de estudios obtenidos al ejecutar la búsqueda.

La selección se aplica según los criterios de inclusión y exclusión, revisando el título y el resumen de cada artículo. En caso de duda, se lee el artículo en su totalidad. Luego, se detectan duplicados y como resultado se obtienen los estudios primarios que servirán para responder las preguntas de investigación planteadas.

Criterios de Inclusión:

- IC1 - El artículo presenta algún tipo de evaluación o análisis acerca de técnicas de revisión, procesos o herramientas relacionados con las revisiones individuales de código o diseño detallado de software

Criterios de exclusión:

- EC1 - El estudio se enfoca en la ejecución de revisiones o inspecciones pero no tiene datos de la fase de revisión o inspección individual
- EC2 - El trabajo presenta experiencias o resultados de revisiones individuales pero aplicadas como parte del PSP
- EC3 - El trabajo presenta un título duplicado
- EC4 - El trabajo realiza una revisión sistemática de literatura
- EC5 - El artículo no ha sido publicado
- EC6 - El trabajo solo está disponible en formato de resumen o presentación *PowerPoint*

- EC7 - El idioma del artículo no es inglés

De manera de maximizar el cubrimiento del mapeo y para evitar agregar tareas que requieren un mayor esfuerzo, no se realizó evaluación de la calidad de los artículos.

4.1.4 Estrategia de extracción de datos y esquema de clasificación

Se define un formulario de extracción de datos, utilizado para extraer toda la información relevante de acuerdo con los objetivos de este trabajo. Cada estudio fue identificado mediante un número correlativo. Además de extraer el resumen y las conclusiones, los datos que se extrajeron de cada artículo son:

- Identificador de artículo
- Título del artículo
- Tipo de artículo (Medio de publicación: revista, conferencia, libro)
- Autores
- Institución principal
- Año
- Foro
- Fuente (Recurso electrónico consultado)
- Contexto (Academia, Industria)
- Artefacto (Diseño, Código: los artículos identificados pueden tratarse de revisiones aplicadas sobre documentos de diseño o sobre código fuente.)

A su vez, se define un esquema de clasificación de los artículos. Este esquema está compuesto por los siguientes aspectos:

- Tipo de investigación
 - De acuerdo a estrategia empírica (Experimento, Estudio de Caso, Encuesta)
 - De acuerdo a clasificación propuesta por Wieringa [16]
- Tipo de técnica utilizada en la revisión (Checklist-based Reading, Usage-based reading, etc.)

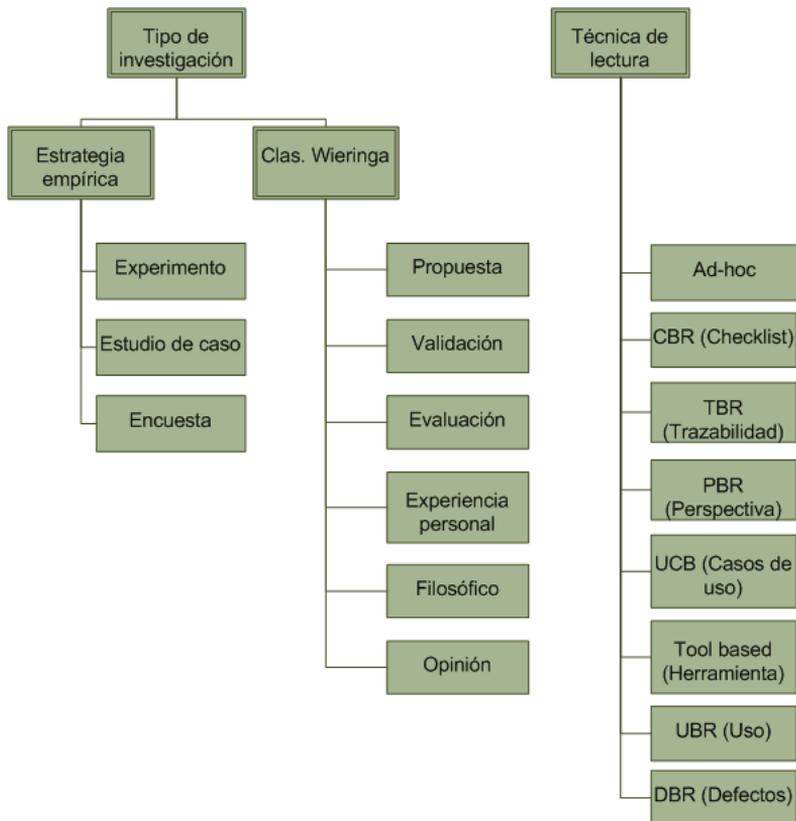


Figura 1: Esquema de clasificación

En la figura 1, se presenta una descripción del esquema de clasificación.

Tipo de investigación

Identifica el tipo de estudio que han llevado a cabo los autores. Para clasificar los trabajos mediante este aspecto se aplican dos enfoques. Por un lado, a partir del tipo o estrategia de investigación presentada en el propio artículo, se asigna una de las categorías definidas en el cuadro 5. Por otro lado, se considera la clasificación propuesta por [Wieringa et al. \[2006\]](#).

Si bien se considera la estrategia del estudio empírico como primer enfoque de clasificación, cabe mencionar que existen varias maneras de clasificar un tipo de investigación [[Freimut et al., 2002](#); [Wohlin et al., 2012](#)]. Una de esas maneras refiere a un enfoque amplio de la investigación empírica, a partir del cual se identifican dos tipos: cualitativo y cuantitativo. Las investigaciones cuyo enfoque es cualitativo generan datos compuestos principalmente por texto, gráficas e imágenes y parten del estudio de la naturaleza del objeto y de la interpretación de un fenómeno desde la percepción de las personas. El enfoque cuantitativo, en cambio, busca cuantificar una relación entre dos o más grupos, y obtiene valores numéricos como datos del estudio.

Una segunda forma de clasificar estudios empíricos considera el objetivo del estudio y de acuerdo con él los estudios empíricos pueden dividirse en exploratorio, descriptivo y explicativo. Un estudio exploratorio debe ser un trabajo donde se estudien objetos dentro de parámetros naturales y donde los resultados surgen de observaciones, con el fin de descubrir lo que sucede. Por otro lado, los estudios descriptivos apuntan a retratar un perfil exacto de eventos, organizaciones o situaciones. Finalmente, un estudio explicativo cuantifica la relación entre dos o más grupos, con el objetivo de identificar relaciones de causa-efecto [[Wohlin et al., 2012](#); [Freimut et al., 2002](#)].

Otra clasificación posible surge a partir de la estrategia del estudio. Las tres estrategias principales son: experimento, estudio de caso y encuesta. Además de esta clasificación existen otras más refinadas, como por ejemplo, la de [Zelkowitz y Wallace \[1998\]](#), en la que se identifican doce modelos experimentales para validación de nuevas tecnologías, y la de [Kitchenham y Pickard \[1998\]](#), que identifica nueve tipos de estudio diferentes.

En el cuadro 5 se presentan las definiciones de estas estrategias de acuerdo con [Freimut et al. \[2002\]](#) y [Wohlin et al. \[2012\]](#).

En cuanto a la clasificación y tomando en cuenta la característica de los estudios recolectados, en este trabajo se considerarán las categorías (experimento, estudio de caso y encuesta) desde el punto de vista de estrategia empírica re-

Estrategia	Descripción
Experimento	Una investigación formal y detallada que es llevada a cabo en condiciones controladas. El objetivo es manipular una o más variables y controlar el resto. El proceso para llevar a cabo un experimento está formado por varias fases: definición, planificación, operación, análisis e interpretación y presentación.
Estudio de Caso	Una investigación en detalle de un caso o de un número de casos relacionados. Se basa en la observación de una actividad o proyecto durante su curso. Se utiliza para monitorear proyectos, o actividades y para investigar entidades o fenómenos en un período específico.
Encuesta	Una investigación amplia donde la información se recoge de una forma estandarizada a partir de un grupo de personas o proyectos. Es un estudio retrospectivo de las relaciones y los resultados de una situación. Puede realizarse cuando una técnica, o herramienta, ya ha sido utilizada o antes de que ésta sea introducida. Es realizada sobre una muestra representativa y luego los resultados son generalizados al resto de la población. Provee un gran número de variables para estudiar, lo que posibilita construir una variedad de modelos.

Cuadro 5: Definición de estrategias empíricas

ción mencionada. Uno de los principales motivos por el que se selecciona este enfoque tiene que ver con la reducida cantidad de artículos que arroja el mapeo y que se adecua al número de categorías. Además, esta clasificación se adapta a los estudios vinculados con las ciencias de la computación e ingeniería de software y, particularmente, con los encontrados en esta revisión.

En este trabajo se toman algunas decisiones para incluir los artículos en estas categorías definidas. En algunos casos, el artículo define claramente el método empírico llevado a cabo y, a su vez, se describe el diseño del estudio de forma completa. En casos en que el trabajo no define explícitamente el tipo de investigación, se realiza un análisis para adjudicar al estudio la categoría correspondiente, tomando en cuenta la definición considerada en el cuadro 5.

Dentro de los artículos se presentan estudios que si bien tienen características de experimento, no alcanzan a conformar un experimento controlado. Por ejemplo, la elección de los sujetos no resulta aleatoria. Estos estudios son defini-

dos como cuasi-experimentos. No obstante, a efectos de este trabajo se agrupan bajo la categoría Experimento.

Para la segunda clasificación del aspecto Tipo de investigación se elige la propuesta de [Wieringa et al. \[2006\]](#). Se decide utilizarla debido a su facilidad y en acuerdo a la sugerencia del trabajo de [Petersen et al. \[2008\]](#), donde se describe que las categorías definidas son fáciles de interpretar y pueden utilizarse para clasificar sin necesidad de evaluar cada artículo al detalle. Por ejemplo, la opción *Evaluation Research* puede ser excluida fácilmente si no hay cooperación de la industria o no se menciona una técnica ya implementada en el mundo real. La categoría *Validation Research* es fácil de identificar al chequear si el artículo presenta hipótesis, utiliza resumen estadístico y describe los componentes principales del diseño de un experimento. El esquema, además, permite identificar, tanto estudios empíricos como trabajos no empíricos. En el cuadro 6 se presenta dicha clasificación.

Tipo de técnica utilizada

La revisión de artefactos de software se puede realizar aplicando diferentes tipos de técnicas de lectura. Una técnica de lectura es un procedimiento paso a paso implementado por un revisor o inspector que permite una inspección bien definida y sistemática de un artefacto, una retroalimentación y una mejora [[Shull et al., 2001](#)]. A partir de este aspecto, se clasifican los trabajos según la técnica aplicada para la lectura de los artefactos bajo revisión. Algunas de estas técnicas son: *checklist-based reading*, *horizontal reading*, *vertical reading* o *usage-based reading*. Las definiciones y características de las técnicas de lectura identificadas en los artículos seleccionados, son las siguientes:

- Ad-Hoc. Ninguna técnica es especificada. El revisor emplea el mecanismo que considera adecuado según su experiencia e intuición para determinar cómo buscar defectos [[Laitenberger y DeBaud, 2000](#)].
- Checklist-based reading. La técnica de lectura aplicada está basada en una o varias listas de verificación que en general son un conjunto de preguntas que tienen la intención de guiar al revisor durante la revisión. Una *checklist* se diseña a partir de datos históricos de defectos aunque puede utilizarse una lista general ya existente. [[Laitenberger y DeBaud, 2000](#)]. Se considera una de las técnicas de revisión más utilizadas y fue descrita inicialmente por [Fagan \[1999\]](#).
- Perspective-based reading. La revisión es realizada desde un enfoque basado en las perspectivas de diferentes interesados. Por ejemplo, desde el

Categoría	Descripción
Validación de investigación	La técnica investigada es novedosa y aún no ha sido implementada en la práctica. Posibles métodos pueden ser: experimentos, simulaciones, prototipación, análisis matemático, etc.
Evaluación de Investigación	La técnica fue implementada en la práctica y una evaluación de la técnica es llevada a cabo. Se muestra cómo fue implementada la técnica en la práctica y cuáles son las consecuencias de la implementación en términos de ventajas y desventajas.
Propuesta de solución	Se propone la solución a un problema. La solución puede ser original o una extensión significativa de una técnica existente. Los potenciales beneficios y la aplicabilidad de la solución se muestra mediante un ejemplo pequeño o una argumentación sólida.
Trabajo filosófico	Estos trabajos esbozan una nueva manera de ver las cosas, mediante la estructuración del campo en forma taxonómica o un marco conceptual.
Artículo de opinión	Estos trabajos expresan la opinión personal de alguien acerca de si una cierta técnica es buena o mala, o acerca de cómo se deben realizar las cosas. No se basan en trabajos relacionados o en metodologías de investigación.
Artículo de experiencia	Trabajos de experiencia cuentan qué y cómo algo ha sido realizado. Debe ser la experiencia personal del autor. En general la evidencia presentada es de carácter anecdótico.

Cuadro 6: Clasificación propuesta por Wieringa

punto de vista de un *tester*, de un desarrollador o de un usuario. PBR pertenece al conjunto de técnicas de lectura basadas en escenario. La técnica se apoya en escenarios que aplican a cada perspectiva [Basili et al., 1996].

- **Horizontal Reading.** Pertenece a la familia de técnicas de lectura para diseño orientado a objetos, llamada Traceability-Based Reading (TBR). Refiere a la técnica de lectura que es utilizada para leer documentos construidos en la misma fase del ciclo de desarrollo. El hecho más importante es revisar la trazabilidad entre documentos de diseño. Apunta a la identificación de defectos por ambigüedad e inconsistencia [Travassos et al., 1999].
- **Vertical Reading.** Al igual que la lectura horizontal, pertenece a la familia de técnicas de lectura para diseño orientado a objetos, llamada Traceability-Based Reading (TBR). Refiere a la técnica de lectura que es utilizada para leer documentos construidos en diferentes fases del ciclo de desarrollo. Por ejemplo, los artefactos de requerimientos son comparados con los artefactos de diseño. La característica más importante es la trazabilidad entre las fases. Se busca la identificación de defectos por omisión o incorrectitud [Travassos et al., 1999].
- **Use Case Reading.** Es una técnica diseñada para inspeccionar sistemas desarrollados de forma orientada a objetos, tomando en cuenta la interacción dinámica de objetos colaborativos. El objetivo es asegurar que cada objeto responde correctamente de acuerdo con su rol en el escenario ejecutado. Esto es, los métodos correctos son invocados y los cambios de estado son consistentes y correctos. Los inspectores crean múltiples escenarios a partir de los casos de uso del sistema. Ciertos métodos pueden llegar a no ser inspeccionados a partir de los escenarios creados [Dunsmore et al., 2003].
- **Usage-based Reading.** Es una técnica de la familia Scenario-based Reading. El foco de la revisión es de acuerdo al uso esperado del sistema. UBR intenta guiar a los inspectores en buscar los defectos que puedan tener un mayor impacto negativo en la usabilidad del software. Los casos de uso son priorizados de acuerdo con la manera en que el sistema va a ser utilizado [Thelin et al., 2003].
- **Specification-based.** La idea fundamental del enfoque es verificar que todos los requerimientos funcionales definidos en una especificación formal están implementados correctamente. La técnica se basa en encontrar conjuntos de sentencias y condiciones, en los programas, que implementen

correctamente las funciones requeridas. Una descripción detallada del método puede encontrarse en el trabajo de Nagoya et al. [2004].

Es pertinente agregar que para realizar la clasificación de los estudios, se incluirán dentro de la categoría Ad-Hoc también aquellos estudios que no identifican la técnica de lectura llevada a cabo. Además, se agrega una categoría para asociar aquellos artículos cuyo proceso de revisión está fuertemente basado en el uso de una herramienta. Un resumen de la clasificación considerada en este trabajo se muestra en el cuadro 7.

Categoría	Descripción
Ad-Hoc (AHR)	Se trata de aquellas revisiones donde no se espera que se aplique una técnica específica o no está mencionada de forma explícita.
Checklists (CBR)	La revisión es guiada por listas de verificación.
Perspective-based reading (PBR)	La técnica de lectura aplicada está basada en el enfoque basado en perspectivas.
Horizontal reading	Se revisan artefactos de diseño OO, en busca de defectos de inconsistencia o ambigüedad. La trazabilidad es el aspecto que dirige la revisión.
Vertical reading	Se revisan artefactos de diseño OO, en busca de asegurar correctitud y completitud. Se comparan los artefactos de diseño con otros artefactos como requerimientos o código. La trazabilidad es el aspecto que dirige la revisión.
Use case reading	Se revisan artefactos de OO recorriendo escenarios creados a partir de los casos de uso del sistema.
Usage-based reading (UBR)	En estos artículos se realizan revisiones de software aplicando un enfoque basado en el punto de vista del uso del sistema y priorizando casos de uso.
Specification-based review	La revisión se basa en utilizar la especificación formal de los requerimientos funcionales.
Tool-based review	Una herramienta es utilizada como soporte para llevar a cabo la revisión. Es decir, la técnica de lectura se aplica en un entorno basado en herramienta

Cuadro 7: Clasificación de técnicas de lectura

4.1.5 Síntesis de los datos extraídos

En la etapa de síntesis de la información extraída se resumen los datos recolectados de los estudios primarios. Estos datos se presentan a modo de resumen cuantitativo a partir de los datos extraídos de cada artículo y según la categorización de los artículos en las clasificaciones definidas. Además se realiza un análisis de forma de poder responder las preguntas de investigación.

A partir de la información extraída se busca presentar distintas distribuciones de los artículos y formas de visualizar los datos relevantes, tales como evolución de la investigación en el tema a través de los años, medios de publicación, contexto e instituciones participantes de cada investigación, artefacto revisado, etc. A su vez, se decide realizar una categorización de los artículos según el esquema de clasificación definido; por tipo de investigación, por estrategias empíricas y técnicas de lectura utilizadas. Por último, se deben presentar los resultados del mapeo en lo que respecta a las preguntas de investigación, clasificando artículos y respuestas por pregunta.

4.2 EJECUCIÓN DEL MAPEO SISTEMÁTICO

Aquí se presenta la búsqueda realizada en la biblioteca digital IEEEExplore y el proceso de selección de los artículos obtenidos.

Para la búsqueda en IEEEExplore se utiliza una cadena de búsqueda adaptada, la cual se aplica primero sobre el campo *resumen* y luego sobre el campo *título*, para posteriormente combinar los resultados.

Al realizar la búsqueda se obtienen como resultado 998 artículos. Se procesan los 998 resultados, con un primer análisis de selección por título y resumen, aplicando los criterios de inclusión y exclusión. Luego del proceso inicial de selección quedan 17 artículos factibles de ser analizados. Posteriormente, se procesa el texto completo aplicando nuevamente los criterios de inclusión y exclusión. Después del análisis de texto completo se seleccionan 13 artículos, a partir de los cuales se realiza la extracción de datos y la síntesis de información.

Si bien hay muchos trabajos que se ocupan de los procesos de revisión e inspección de software, son pocos o casi inexistentes los que presentan estudios centrados en revisiones individuales. La característica de los trabajos seleccionados es de estudios que evalúan técnicas de revisión de software, llevado a cabo, en su mayoría, mediante procesos clásicos, grupales, pero que dentro del análisis de datos presentan evidencia de datos que pertenecen a las actividades individuales.

Luego de realizar un análisis más profundo, durante el proceso de extracción de datos y clasificación de los trabajos, se resuelve descartar un trabajo más. El motivo es que si bien, se incluyen datos de revisión individual, no se identifican claramente los datos generados de la revisión sobre diseño y código, sino que los datos están agregados a las revisiones sobre documentos de requerimientos. De esta manera, son 12 artículos sobre los que se presentan los datos extraídos.

4.3 RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados de la ejecución del mapeo y se cuenta cómo se llevó a cabo la búsqueda y la selección de los estudios. Además, se presentan los datos extraídos, la síntesis de la información según los datos extraídos y el esquema de clasificación. Por último, se presentan los resultados del mapeo mediante un análisis de las respuestas a las preguntas de investigación.

4.3.1 *Extracción y análisis de datos*

Como se mencionó en la sección anterior, los artículos seleccionados fueron 12. Para la extracción de los datos relevantes de cada artículo se utiliza una estrategia basada en la definición de un formulario a completar para cada artículo y un esquema de clasificación, ambos definidos en la estrategia de extracción de datos.

De forma de referenciar y clasificar los estudios de una manera más simple, se identifica cada artículo con un número. En el cuadro 8 se presenta un resumen de los artículos seleccionados que contiene identificador, título y referencia bibliográfica.

Toda la información recabada fue consolidada en una planilla electrónica en la que cada columna se corresponde con cada uno de los datos a extraer definidos en el protocolo del mapeo. Además, esta planilla fue utilizada para trabajar sobre la información de los artículos, por ejemplo, para generar la representación de los resúmenes cuantitativos.

Por otro lado, la estrategia de síntesis de los datos utilizada abarca distintas formas de visualizar información general de los artículos, clasificación según los esquemas definidos y mapeo de respuestas con las preguntas.

El cuadro 9 presenta la evolución de los artículos publicado a través de los años. Es posible apreciar, aunque con intermitencias, cómo a lo largo de los años se cuenta con pocas publicaciones pero de forma sostenida.

Id.	Título
[1]	An experimental comparison of checklist-Based Reading and Perspective-Based Reading for UML Design Document Inspection [Sabaliauskaite et al., 2002]
[2]	An Analysis System for Multi-document Software Review Using Reviewers Eye Movements [Uwano et al., 2008]
[3]	An internally replicated quasi-experimental comparison of checklist and perspective based reading of code documents [Laitenberger et al., 2001]
[4]	Reviewing Software Diagrams: A cognitive study [Hungerford et al., 2004]
[5]	Studying the effects of code inspection and structural testing on software quality [Laitenberger, 1998]
[6]	Validating the defect detection performance advantage of group designs for software reviews: report of a replicated experiment [Land et al., 1997]
[7]	The Significance of Participant Experience when Evaluating Software Inspection Techniques [McMeekin et al., 2009]
[8]	An investigation of the approach to specification-based program review through case studies [Nagoya et al., 2004]
[9]	Can peer code reviews be exploited for later information needs? [Sutherland y Venolia, 2009]
[10]	Expectations, outcomes, and challenges of modern code review [Bacchelli y Bird, 2013]
[11]	Quantitative modeling of software reviews in an industrial setting [Laitenberger et al., 1999]
[12]	An Empirical Study on a Specification-Based Program Review Approach [Nagoya et al., 2006]

Cuadro 8: Estudios seleccionados

Año	Cantidad de estudios
1997	1
1998	1
1999	1
2001	1
2002	1
2004	2
2006	1
2008	1
2009	2
2013	1

Cuadro 9: Textos seleccionados por año

En el cuadro 10, se presenta un resumen de los medios de publicación de los trabajos. Los datos muestran que la mayoría de los trabajos tuvieron como medio de publicación una conferencia. Los lugares de publicación de los artículos se distribuyen tal como se muestra en el cuadro 11.

Medio	Cantidad
Conferencia	10
Revista	2

Cuadro 10: Medio de publicación de los trabajos

A partir de los cuadros 12 y 13 puede apreciarse la distribución de los trabajos según al contexto de la investigación y las instituciones participantes. En el caso de las instituciones, se toma en cuenta el hecho de que algunos trabajos fueron realizados por más de una institución.

Por otro lado, en el cuadro 14 puede apreciarse la distribución de los trabajos según el artefacto sobre el que se aplica la revisión de software. Encontramos que en la mayoría de trabajos el artefacto es el código. Sin embargo, existen otros trabajos que no fueron seleccionados para esta investigación, donde el artefacto es el diseño, pero no a nivel de diseño detallado.

Lugar de publicación	Estudios
Australian Software Engineering Conference	2
IEEE Transactions on Software Engineering	2
International Conference on Software Engineering	2
International Symposium on Empirical Software Engineering	1
International Conference on Software Engineering Advances	1
International Symposium on Software Reliability Engineering	1
IEEE International Conference on Engineering Complex Computer Systems	1
International Software Metrics Symposium	1
International Conference on Dependability of Computer Systems	1

Cuadro 11: Lugares de publicación de los trabajos

Contexto	Cantidad de estudios
Academia	8
Industria	4

Cuadro 12: Contexto de la investigación

Institución	Estudios
Fraunhofer Inst. for Exp. Software Eng.	3
Fac. of Comput. and Inf. Sci., Hosei Univ., Tokyo	2
Microsoft	2
Dept. of Informatics and Math., Osaka Univ., Japan	1
Grad. Sch. of Inf. Sci., Nara Inst. of Sci. and Technol., Ikoma, Japan.	1
Bosch Telecom GmbH	1
Dept. of Manage. Inf. Syst., Wisconsin Univ., Oshkosh, WI, USA	1
New South Wales Univ., Sydney, NSW, Australia	1
Digital Ecosyst. and Bus. Intell. Inst., Curtin Univ. of Technol., Bentley, WA, Australia	1
Univ. of Saskatchewan, Saskatoon, SK	1
Fac. of Inf., Univ. of Lugano, Lugano, Switzerland	1
Lucent - Product Realization Center for Optical Networking	1

Cuadro 13: Instituciones de las investigaciones

Artefacto	Cantidad de estudios
Diseño	4
Código	9

Cuadro 14: Artefacto sobre el que se aplica la revisión de software

A continuación, se presenta la categorización realizada con los estudios seleccionados según el esquema de clasificación definido.

Clasificación según tipo de investigación

Tal como se definió anteriormente, se considerarán dos clasificaciones. Las categorías de la primera clasificación son Experimento, Estudio de Caso y Encuesta. La segunda clasificación está basada en las categorías propuestas por [Wieringa et al. \[2006\]](#), detalladas en el cuadro 6. Con el fin de agrupar los artículos según estas clasificaciones, se realiza primero un breve análisis de sus características principales.

El artículo [1] presenta un experimento que tiene como objetivo principal comparar la efectividad y el costo de la inspección al utilizar CBR y PBR. En él se plantean tres hipótesis. En la primera se asume que los sujetos consumen más tiempo en la inspección si utilizan CBR que si aplican PBR. En la segunda, se asume que el costo por defecto de los individuos que usan PBR es mayor que el costo por defectos de los individuos que usan CBR. En la tercera, se asume que existe una diferencia en la efectividad de los individuos que usan PBR comparada con la efectividad de aquellos que usan CBR como técnica de inspección. Por otro lado, se toma un diseño de *Un factor con dos niveles* (ver [Juristo y Moreno \[2013\]](#)), donde las intervenciones comprenden las técnicas de lectura aplicadas (PBR y CBR) y los sujetos utilizan solo un tratamiento sobre un objeto experimental. Las técnicas de lectura son las variables independientes manipuladas; participaron cincuenta y nueve estudiantes, mientras que se utilizaron documentos con diagramas UML de dos sistemas de software como objetos del experimento.

En el artículo [4] se presenta un experimento donde se realiza un análisis de protocolo verbal como forma de obtener evidencia de los procesos cognitivos (identificación de patrones de búsqueda) empleados al aplicar una revisión de defectos. Las preguntas de investigación buscan responder, por un lado, qué técnicas cognitivas de búsqueda son usadas por individuos con experiencia al inspeccionar múltiples artefactos de software, incluyendo diagramas y, por otro, cuál es la relación entre diferentes técnicas cognitivas de búsqueda usadas en la inspección de múltiples diagramas de software y la efectividad de la detección de defectos. Participaron doce individuos profesionales de distintas empresas de software que deben decir en voz alta las decisiones que van tomando mientras aplican una técnica cualquiera en busca de defectos en diagramas de software.

El estudio [5] tiene como objetivo principal determinar la manera en que cada técnica de detección de defectos, inspección y *testing* estructural, por separado y en conjunto, contribuyen a la calidad del software. Como hipótesis de trabajo se asume que la inspección y el *testing* estructural se complementan bien en sus capacidades de detección de defectos. La variable independiente manipulada es la técnica de detección de defectos para individuos y equipos. Veinte estudiantes participaron como sujetos y el objeto experimental es un módulo de código escrito en C.

En el estudio [6] se replica un experimento anterior y se compara el rendimiento en un proceso de revisión de grupos interactivos con el rendimiento individual, en el contexto de revisiones técnicas de software. La variable independiente es la revisión de diseño y los sujetos se asignan a dos tratamientos: revisión individual y revisión en grupo interactivo. Se utilizaron ciento un estudiantes como individuos del experimento y los objetos de revisión consistieron en porciones de código y documentos de diseño detallado.

Por su parte, el estudio [7] compara, por un lado, las técnicas de inspección UBR, UCR y CBR y, por otro, el impacto del nivel de experiencia del inspector en la eficacia de la inspección. El experimento controla dos variables independientes: la técnica de lectura implementada por cada participante (UBR, UCR, CBR) y el nivel de experiencia de los participantes (estudiantes y profesionales). Participaron treinta y seis estudiantes y veintiséis profesionales. La asignación de técnicas se hace de forma aleatoria. A su vez, los artefactos presentados a los individuos incluyen una especificación en lenguaje natural, una especificación de clases, un diagrama de clases del sistema, una porción de código java y el acceso a la API de Java.

Por su parte, el artículo [3] presenta un cuasi-experimento con dos réplicas, donde, por ejemplo, no se asignan de forma aleatoria los individuos en los grupos. Se compara la técnica de lectura de PBR para detección de defectos en código con el enfoque CBR. Las hipótesis experimentales son, en primer lugar, que para equipos la efectividad de PBR es mayor que la efectividad de CBR. En segundo lugar, que el costo de detección de defectos para equipos es menor al usar PBR que al utilizar CBR. En tercer lugar, el costo de las reuniones es menor en equipos con PBR que con CBR. Por último, el costo total es menor con PBR que al usar CBR. En cada ejecución del experimento se utilizaron veinte profesionales, por lo que, en total, participaron sesenta sujetos. Como objeto de revisión se dispone de código C, junto con un documento con la especificación de las operaciones.

Como ejemplo de estudio de caso, el artículo [2] presenta una evaluación acerca de la utilidad de un sistema de evaluación de revisión multi-documento.

Dicho sistema graba para su análisis los movimientos de ojos de los revisores y operaciones de teclado y mouse. Se busca verificar que el sistema satisfaga los siguientes requerimientos: la detección de cambio de documento, el reconocimiento del movimiento de los ojos entre líneas, la posibilidad de que los revisores tomen notas acerca de los defectos detectados y el soporte para análisis de datos. En este estudio se le solicita a los participantes que encuentren defectos en un documento de diseño en base a la lectura de un conjunto de otros documentos. Estos documentos incluyen la especificación de requerimientos, el documento de diseño, un archivo con datos y una *checklist* para revisión de documentos de diseño. A partir del registro de los movimientos de los ojos de los participantes se espera realizar una observación de los patrones de lectura y analizar cuantitativamente la relación entre los patrones y el rendimiento de la revisión.

Por su parte, en el artículo [8] se propone un nuevo enfoque de revisión rigurosa de programas basada en su especificación formal. El lenguaje de especificación que utiliza este trabajo es SOFL (*structured object-oriented formal language*). En este trabajo se conducen dos estudios de caso para examinar la efectividad y las debilidades del método de revisión propuesto. Se consideran dos programas diferentes, implementados a partir de la misma especificación formal. El primer estudio de caso examina qué tan efectivo es el método de revisión para detectar defectos en el programa implementado por el propio revisor. Por otro lado, el segundo estudio de caso está diseñado de tal forma que permite examinar la efectividad del método de revisión, pero bajo la condición de que el revisor sea diferente al programador.

En cuanto al estudio [10], se aplica una combinación de estrategias de investigación para explorar características de las revisiones de código informales y basadas en herramientas. En particular, se combina la observación con las entrevistas y encuestas. Este tipo de estudio puede ser considerado como un estudio de caso, ya que abarca la observación de datos de proyectos ya ejecutados dentro de una compañía específica en la industria. Además, estudia los fenómenos relacionados con la técnica, tales como la motivación, desafíos y resultados. En el contexto de este estudio se entiende por revisiones modernas aquellas revisiones informales, donde se utiliza una herramienta como soporte. Las preguntas de investigación que se formulan indagan en las motivaciones y expectativas de las revisiones de código modernas; en las posibles diferencias entre gerentes, desarrolladores y *testers*; en los resultados actuales de las revisiones de código modernas; en si estos resultados concuerdan con las expectativas y, por último, en los principales desafíos experimentados, relativos a las expectativas y los resultados al realizar revisiones de código modernas. El

método de investigación llevado a cabo persigue un enfoque mixto, cualitativo y cuantitativo. Los datos son recolectados de varias maneras, como por ejemplo, análisis de los datos de un estudio previo acerca del uso de la revisión de código basada en herramienta por parte de los distintos grupos de la compañía; observación de la ejecución de una revisión por parte de los sujetos; entrevistas con los diferentes desarrolladores y categorización de la información surgida de las entrevistas. Luego, se realiza un diagrama de actividad para organizar estas categorías definidas y, por último, se lleva a cabo una encuesta para validar los conceptos surgidos a partir de las etapas mencionadas.

Por su parte, el estudio [11] postula un modelo teórico en el que se especifican los factores que impactan en el grado de éxito de las revisiones de software en una determinada compañía. De acuerdo con este estudio, el éxito de la revisión se define como la cantidad de defectos encontrados durante la revisión. Los factores considerados son el esfuerzo de preparación y el tamaño del documento inspeccionado. El análisis se realiza a partir de más de 300 revisiones realizadas en esta empresa sobre especificaciones de requerimientos, diseño y código. Asimismo, se utiliza el método de análisis de camino (*path analysis*) como forma de verificar el modelo teórico planteado. Podemos considerarlo un estudio de caso, puesto que se investigan un conjunto de casos ya realizados donde se recolecta y analiza la información para investigar ciertos fenómenos para ese conjunto y periodo estudiado.

En el artículo [12] se describe un estudio empírico en el que se lleva a cabo un proceso de revisión sobre un sistema de software particular. Tiene como objetivo investigar la efectividad y eficiencia de un enfoque de revisión basado en la especificación formal, presentado en otro estudio. Tomando como caso un sistema de ATM, se realiza un proceso de revisión clásico que incluye las actividades de planificación, detección, recolección y corrección. A partir de la ejecución de todo el proceso de revisión, se obtienen datos de efectividad y de esfuerzo requerido. Además, el estudio provee un proceso efectivo para utilizar el enfoque de revisión rigurosa basada en especificación formal, a la vez que presenta una herramienta de soporte al proceso de revisión.

En cuanto al artículo [9], utiliza la encuesta como método y realiza una investigación de las prácticas de equipos de productos de software en una empresa. Analiza las experiencias de revisiones entre pares llevadas a cabo en dicha compañía. Se parte de la hipótesis de que el conocimiento intercambiado durante la revisión de código puede ser de gran valor para los ingenieros para entender o modificar el código. El estudio consiste en un cuestionario preliminar vía correo, una encuesta y el análisis de la transcripción de los correos de revisiones realizadas utilizando el correo. La encuesta fue diseñada a partir de las respues-

tas del cuestionario preliminar y los participantes la contestaron a través de un formulario web. Allí fueron consultados acerca de las herramientas de comunicación que utilizan al conducir revisiones de código y acerca de aspectos de la revisión que fueron ineficientes o difíciles.

A partir de las características recién mencionadas, los artículos se pueden clasificar de acuerdo con las siguientes categorías establecidas (Experimento, Estudio de Caso y Encuesta).

Estrategia	Artículos
Experimento	[1],[3],[4],[5],[6],[7]
Estudio de Caso	[2],[8],[10],[11],[12]
Encuesta	[9],[10]

Cuadro 15: Clasificación de estrategias empíricas

El cuadro 15 recoge la distribución de los estudios según la estrategia empírica. Dentro de la categoría Experimento se agrupan tanto los experimentos controlados, como los cuasi-experimentos (experimentos que no alcanzan totalmente la formalidad experimental esperada para un estudio controlado). Los artículos [1], [4], [5], [6] y [7] son experimentos controlados. Definen claramente los componentes del diseño del experimento; se realizan en un ambiente controlado y asignan a los sujetos de forma aleatoria. Por su parte, el artículo [3] es un cuasi-experimento. Respecto al estudio de caso, cinco de los artículos trabajan tomando el estudio de caso como estrategia de investigación. Vale aclarar que el trabajo [10] combina distintas estrategias de investigación, por lo que se clasifica como estudio de caso y como encuesta, junto con el trabajo [9] que aplica a la última categoría.

En el cuadro 16, se presenta la distribución de los estudios a través de las categorías detalladas en 6.

Como se ve en la tabla 16, los artículos [2] y [8] pertenecen a la categoría de *Validación de la investigación* debido a que presentan ideas novedosas. En la categoría siguiente, *Evaluación de la investigación*, se encuentran el resto de los estudios ya que todos ellos realizan evaluaciones de técnicas que ya han sido implementadas en la práctica.

Clasificación según técnica de lectura

La distribución de los artículos seleccionados según la clasificación por técnica de lectura queda ilustrada en el cuadro 17.

Tipo de investigación	Artículos
Propuesta de solución	
Validación de la investigación	[2],[8]
Evaluación de la investigación	[1],[3],[4],[5],[6] [7],[9],[10],[11],[12]
Experiencia personal	
Trabajo filosófico	
Artículo de opinión	

Cuadro 16: Clasificación de tipos de investigación

Técnica de lectura	Artículos
Ad-Hoc (AHR)	[5],[9],[11]
Checklists (CBR)	[1],[2],[3],[6],[7],[11]
Perspective-based reading (PBR)	[1],[3]
Horizontal reading	[4]
Vertical reading	[4]
Use case reading	[7]
Usage-based reading (UBR)	[7]
Specification-based review	[8],[12]
Tool-based review	[10],[12]

Cuadro 17: Clasificación de técnicas de lectura

Considerando la técnica Ad-Hoc, en los artículos [5], [9] y [11] no se especifica una técnica de lectura en particular llevada a cabo por los sujetos para realizar la revisión de los artefactos. En particular, en el artículo [11] si bien se tiene indicado en el diseño del ciclo de desarrollo el uso de checklist para revisiones, se aclara que los revisores en general tienen mucha experiencia y no utilizan una técnica de lectura determinada.

La técnica de lectura basada en checklist, por su parte, se presenta en la mayoría de los artículos. En particular, en los estudios [1], [3] y [7] se compara la eficiencia de esta técnica con otros enfoques como PBR, UCR y UBR. Por ejemplo en [1], tomando como artefactos documentos de diseño, se encuentra que PBR requiere menos tiempo de inspección que CBR, pero CBR presenta menor costo por defecto. Por otro lado, el estudio [3], toma como artefacto el código y encuentra que PBR tiene menor costo por defecto que CBR. El trabajo [7] no encuentra diferencia significativa en efectividad entre las técnicas CBR, UCR y UBR sobre código. En [2] se utiliza una checklist genérica para revisión de documentos de diseño. En el artículo [6], se adaptó una checklist que consistía en un guía y ejemplos de clasificación de defectos, de forma de hacerla apta para revisión de código. En cuanto al artículo [11], es agrupado en esta categoría debido a que en el proceso de desarrollo indicado se tiene previsto el uso de una *checklist* para revisiones. A pesar de ello, como se mencionó anteriormente, se especifica que los revisores no utilizan una técnica de lectura en particular.

Con respecto a las técnicas de lectura horizontal y vertical, en el trabajo [4] se identifican las técnicas de lectura que tienen foco en la trazabilidad como son la lectura horizontal y la vertical. Durante el experimento, la mayoría de los sujetos utiliza lectura vertical. En cuanto a la lectura horizontal, se detallan 4 tipos de patrones de búsqueda identificados dentro de esa técnica.

En los estudios [8] y [12], se presenta y aplica la técnica de revisión basada en especificación. Primeramente es presentado el enfoque y sus características, y luego es aplicado dentro de un proceso definido, introduciendo además una herramienta de soporte.

En cuanto a la técnica *tool-based review*, tanto en el estudio [10] como en el [12] se utiliza una herramienta como soporte al proceso de revisión. En [10] se trata de una herramienta utilizada por Microsoft, llamada *CodeFlow*.

4.3.2 Respuestas a las preguntas de investigación

En esta sección se describe y analiza la información de los artículos que dan respuesta a las preguntas de investigación.

El cuadro 18 resume la relación entre preguntas y los artículos que contienen respuestas a cada una de ellas.

Pregunta	Artículos
RQ.1. ¿Por qué motivos se decide realizar revisiones individuales de código y diseño detallado?	[2],[3],[9],[10],[11]
RQ.2. ¿Por qué motivos las revisiones individuales sobre código y diseño detallado no son adoptadas en la industria del software?	[12]
RQ.3. ¿Qué evidencia existe acerca de la efectividad y/o del costo de las revisiones individuales de código y diseño detallado?	[1],[2],[3],[4],[5],[6],[7],[8]
RQ.4. ¿Qué factores influyen en la efectividad y el costo de aplicar revisiones individuales de código y diseño detallado?	[1],[2],[3],[4],[5],[11],[12]

Cuadro 18: Artículos por preguntas

Motivos para el uso de revisiones individuales

Cinco artículos reportan información relevante para responder la pregunta RQ1; las respuestas se ilustran en el cuadro 19. Se realizó una categorización por términos claves de las respuestas. En dicho cuadro se muestran estos conceptos representativos de las respuestas a esta pregunta de investigación. Por otro lado, es pertinente agregar que si bien los artículos [2], [3] y [12] se refieren a revisiones generales, se consideran los conceptos incluidos en esos artículos con el fin de enriquecer el conjunto de respuestas.

En [2] se plantea que la necesidad de las revisiones se debe a que puede realizarse en etapas tempranas sin haber desarrollado el sistema, por lo que reduce el costo de retrabajo, especialmente en el contexto de proyectos de gran escala donde la detección y corrección consumen muchos recursos.

El artículo [3] sostiene que las revisiones son requeridas ante altos requerimientos de calidad exigidos, debido a la característica del software de telecomunicaciones desarrollado por la empresa. Además, en el artículo se menciona que muchos defectos son encontrados en el nivel de *testing* de integración e identifican que la mayoría son insertados al codificar. Por esto, la baja calidad actual hace necesario optar por técnicas de revisión para mejorar. El experimento en sí mismo forma parte de una entrenamiento de inspecciones en la

Respuestas	Artículo
Proyectos donde se consumen enormes recursos al detectar y corregir defectos	[2]
Posibilidad de realizarse en etapas tempranas	[2]
Reducción de costos	[2],[3]
Altos requerimientos de calidad	[3]
Mejorar calidad actual	[3],[10]
Transferencia de conocimiento	[9],[10]
Soluciones alternativas	[10]
Cohesión de equipo	[10]
Exigencia del Proceso de desarrollo seguido	[11]

Cuadro 19: Artículos por respuestas a la pregunta RQ1

empresa, que tiene como cometido aplicar inspecciones en etapas tempranas y, así, reducir el costo de detección y corrección de los efectos.

En el estudio [10] se recolectan datos de diferentes fuentes: análisis de estudios previos, observaciones y entrevistas con desarrolladores, datos de entrevista, datos de comentarios de revisión de código y encuestas a gerentes y programadores. Según los desarrolladores entrevistados, las mayores motivaciones son: encontrar defectos, mejorar el código, buscar soluciones alternativas, transferir conocimiento, aumentar conciencia de equipo, mejorar el proceso de desarrollo y compartir la propiedad del código. A partir de los resultados, se puede identificar que las revisiones de código proveen un amplio espectro de beneficios para los equipos de software como, por ejemplo, transferencia de conocimiento, conciencia de equipo y mejora de soluciones a problemas.

En [9] se realiza una encuesta donde se identifica la transferencia de conocimiento como uno de los motivos de adopción de las revisiones.

Por último, en el entorno del trabajo [11] se utiliza un proceso de desarrollo que requiere realizar revisiones en distintas etapas del ciclo de desarrollo.

Luego de analizar la respuesta que dan estos artículos a la primera pregunta, podemos identificar que los principales motivos para realizar revisiones individuales son los siguientes:

- Pueden realizarse en etapas tempranas, lo que reduce el costo de retrabajo. (Esto es importante en contextos donde la detección y corrección exigen muchos recursos, especialmente en proyectos de gran tamaño).

- Son útiles ante altos requerimientos de calidad, por ejemplo, en contextos de software crítico.
- Reducen el costo y mejoran la calidad actual.
- Facilitan la transferencia de conocimiento, la conciencia de equipo y la búsqueda de soluciones alternativas.
- Son exigidas por el proceso de desarrollo seguido por la organización.

Las respuestas resumidas a la pregunta RQ2, que indaga sobre los motivos de la no adopción de las revisiones individuales, se muestran en el cuadro 20. Como se ve, solamente el artículo [12] presenta información que puede alinearse con esta pregunta. Allí se menciona que las técnicas existentes aún enfrentan desafíos en la industria, debido a que hay pocos enfoques sistemáticos o guías precisas para aplicar las técnicas en la práctica. Asimismo, agrega que muchas técnicas de revisión requieren confianza de los revisores en sus propias habilidades y experiencia.

Respuesta	Artículos
Falta de procesos sistemáticos.	[12]
Falta de guías precisas.	[12]

Cuadro 20: Artículos por respuestas a la pregunta RQ2

Efectividad y costo

Con respecto a la pregunta RQ3, podemos subdividir las respuestas de acuerdo con efectividad y costo. En cuanto a efectividad, en el cuadro 21 se muestran las respuestas de cada artículo indicando el artefacto revisado.

El trabajo [1] realiza un análisis de las técnicas CBR y PBR en las revisiones de diseño. Se recolectan resultados individuales y grupales. Para los resultados individuales se comparan efectividad en la detección de defectos, tiempo transcurrido en inspección y costo por defecto. Los resultados alcanzados acerca de la efectividad indican que la diferencia entre los valores de la efectividad al aplicar cada una de las técnicas no es significativa. La checklist utilizada fue desarrollada por los investigadores a partir de otras genéricas, que consiste en una estructura de dos componentes: descripción en donde buscar defectos y lista de preguntas para ayudar a encontrar defectos. Para la técnica PBR se tomaron en cuenta las perspectivas de usuario, diseñador e implementador. En

Efectividad (Prom. defectos encontrados sobre el total)	Artefacto	Artículos
PBR 69,1 % - CBR 70,2 %	Diseño	[1]
CBR 60,56 %	Diseño	[2]
TBR 10,93 %	Diseño	[4]
Ad-hoc 38 %	Código	[5]
CBR 19,1 %	Código	[6]
Estudiantes: 24 % (UBR), 23 % (CBR), 12 % (UCR) Profesionales: 37 % (UBR), 40 % (CBR), 31 % (UCR)	Código	[7]
Autor 60 % - Par 80,7 %	Código	[8]

Cuadro 21: Artículos por respuestas a la pregunta RQ3 - Efectividad

total se inyectaron 15 defectos, repartidos entre los distintos diagramas de diseño.

En el estudio [2] se lleva a cabo un estudio de caso donde se analizan factores que impactan en la performance de la detección de defectos cuando se tienen revisiones multi-documentos. Aplicando la técnica de CBR, a partir de una lista genérica, se obtiene una efectividad en la revisión de diagramas de diseño del 60.56 % sobre un total de 9 defectos inyectados.

El artículo [4] tiene como objetivo capturar y analizar patrones de búsqueda efectivos utilizados por desarrolladores experimentados al realizar revisión de diagramas de software. Los individuos utilizan lectura vertical y lectura horizontal. Se observa que el tipo de técnica utilizada tiene impacto en la efectividad. La efectividad registrada es considerablemente menor al resto de los estudios: el valor 10.93 % no incluye los falsos positivos y es en base a una cantidad de 29 defectos inyectados. En cuanto a la baja efectividad, se discute acerca de que en ese estudio se dispone de un máximo de dos horas para toda la revisión y se hace referencia a otros estudios que han mostrado que la cantidad de defectos detectados para revisiones de 2 horas varía entre 0 y 8.

En cuanto al código, puede verse que los artículos [5], [6] y [7] son los que presentan datos de efectividad. En [5] se estudia la interacción de la inspección de código y el *testing* estructural. A partir del experimento se obtienen datos de la fase de inspección individual. La efectividad identificada es de 38 % sobre un total de 13 defectos.

En el estudio [6], si bien se estudia la *performance* desde el punto de vista de diseño de los equipos o grupos de revisión, se generan datos individuales sobre la efectividad en revisión de código utilizando una *checklist*. En un total de 25 defectos el promedio de efectividad encontrado es 19.1 %.

Por su parte, el estudio [7] presenta datos para la comparación de las técnicas UBR, CBR y UCR y, además, compara los resultados de estudiantes con profesionales. Se trabaja sobre código inyectado con 14 defectos y se utiliza una *checklist* adaptada a partir de dos *checklists* utilizadas en otros estudios. Los resultados no muestran diferencias significativas entre técnicas, pero sí se encuentran diferencias significativas entre los resultados de estudiantes con los de los profesionales.

Finalmente, el artículo [8] plantea un enfoque basado en especificación formal. En ese trabajo se realizaron dos casos de estudio. En el primero el autor revisa su propio código y en el segundo el código es revisado por un par. Se encuentra que la efectividad es considerablemente mayor cuando la revisión del código implementado es realizada por una persona distinta al autor. En ambos casos se parte de la misma especificación formal.

Así, como se acaba de mostrar, las respuestas a la pregunta RQ3 relativas a la efectividad permiten ver los resultados de [1] y [2]. Cuando el artefacto es diseño, los resultados son buenos e incluso mejores que otras técnicas de testing funcional o estructural. Algunos resultados de esas otras técnicas se pueden ver en distintos trabajos [Hetzel, 1976; Myers, 1978; Wood et al., 1997; Basili y Selby, 1987]. Por el contrario, el caso de [4] es más bajo aunque, como se mencionó previamente, esto puede estar condicionado a que se tiene más cantidad de defectos en un tiempo acotado. En cuanto a los estudios en los que el artefacto es código, los resultados se reparten en resultados similares a otros estudios y algunos inferiores. Es decir, los datos de sujetos profesionales de los trabajos [5] y [7] están dentro del mismo nivel de efectividad registrado por otros trabajos [Hetzel, 1976; Myers, 1978; Wood et al., 1997]. En cambio, los datos de individuos no profesionales de [6] y [7] presentan valores por debajo de los promedios vistos en esos otros trabajos.

Es conveniente agregar algunas observaciones acerca de los bajos valores de efectividad en los estudios [4], [6] y [7]. El estudio [4] determina un tiempo máximo de dos horas para la revisión. En el siguiente caso, en el estudio [6] el tiempo de inspección individual se encuentra acotado a una hora aproximadamente. En el estudio [7], en cambio, cuando los individuos son estudiantes no se dispuso de un tope de tiempo para realizar la revisión. De todas maneras, se menciona que al estudiante que dedicó más tiempo, la revisión le demandó setenta y ocho minutos. Es decir, bastante menos de dos horas. A su vez, este in-

dividuo fue considerado un *outlier*. El escaso tiempo permitido o utilizado para las revisiones individuales puede ser una de las causas para la baja efectividad en la detección de defectos.

Se agrega, además, el hecho de la considerable cantidad de defectos inyectados para poder detectar en el tiempo disponible, como es el caso de [4] y [6]. En este sentido, existen varios trabajos en la literatura que refuerzan estas ideas. Por ejemplo, se entiende que en la revisión individual existe una relación entre el tiempo dedicado a la revisión y la cantidad de defectos detectados [Porter et al., 1998]. En dicho trabajo, se observa que la cantidad de defectos detectados varía de cero a ocho defectos en un esfuerzo de dos horas de revisión. Por otra parte, otro estudio empírico obtuvo resultados que indican un promedio de detección de cuatro defectos por hora durante las primeras ocho horas [Biffi y Halling, 2003].

Considerando las características de las *checklists* utilizadas, puede cuestionarse la validez de la comparación de diferentes técnicas de lectura con CBR, a nivel de revisión individual, debido a que se han basado en *checklists* genéricas, extraídas de la literatura o adaptadas al contexto a partir de otros estudios empíricos. Incluso, se toma como base una *checklist* genérica que fue pensada para documentos de requerimientos o diagramas de diseño de alto nivel y luego se utiliza la misma *checklist* para revisión de código. Por otro lado, algunos autores enfatizan que la *checklist* debe basarse en datos históricos de la organización y de los individuos que forman parte de la construcción de los artefactos bajo revisión. Es decir, estos autores apuntan que para construir una *checklist* que sea adecuada para las necesidades de los programadores de una organización, es necesario, primero, adaptar ciertos elementos de la lista utilizada como referencia. Pero además, deben construirse *checklists* personales enfocadas en las debilidades de cada programador. Este método incluso sirve para mejorar el rendimiento personal de los programadores. Tomando en cuenta que cada individuo es diferente, personalizar la *checklist* le permitirá al profesional descubrir aquellos defectos que acostumbra inyectar y, por otro lado, conocer sus errores y así evitar inyectar los mismos defectos. Continuando con el ciclo, la *checklist* debe ser nuevamente actualizada para enfocarse en los tipos de defectos remanentes [Humphrey, 2000; Hayes y Over, 1997; Vallespir y Nichols, 2016].

En lo que tiene que ver con el costo, en el cuadro 22 se muestran las respuestas de cada artículo a la pregunta RQ3.

Si bien en el artículo [1] se encuentra que al utilizar PBR se requiere menos tiempo de inspección, también se observa que el costo por defecto es menor al aplicar CBR. En cuanto a la primera afirmación, se argumenta que se debe a que, al aplicar CBR, es necesario analizar todos los diagramas. Sin embargo,

Respuesta	Artefacto	Artículos
PBR 10,2 min/def - CBR 6,2 min/def	Diseño	[1]
CBR 4,59 min/def	Diseño	[2]
36,36 min/def	Diseño	[4]
Quasi-Exp: CBR 115,2 PBR 145,6 minutos (total) Rep (1): CBR 118,8 PBR 132,5 minutos (total) Rep (2): CBR 168 PBR 150,5 minutos (total)	Código	[3]
Autor 600 min - Par 1200 min	Código	[8]

Cuadro 22: Artículos por respuestas a la pregunta RQ3 - Costo

al aplicar PBR, depende del rol o perspectiva en ejecución. Por ejemplo, si las perspectiva es de usuario, se tendrá un conjunto de diagramas o documentos de diseño relevantes, que posiblemente no serán el total de los documentos. El costo por defecto en PBR sería mayor, puesto que esta técnica implica más tareas para detectar defectos que leer la lista de preguntas.

En cuanto al costo de revisión sobre diseño, aplicando CBR, en el estudio [2] se generan datos que muestran que se requieren 4,59 minutos de esfuerzo por cada defecto.

Como último caso de revisión de diseño, el trabajo [4] presenta datos donde se ve que el esfuerzo promedio de la revisión es de 110 minutos o en función de los defectos identificados 36,36 minutos por defecto. Los individuos que realizaron las revisiones no fueron provistos de guías ni pautas de enfoques particulares para llevar a cabo la revisión.

Por otro lado, el artículo [3] genera datos de costo aplicando CBR y PBR en tres experimentos (un experimento y dos réplicas), para revisión de código. En este estudio se divide el código a revisar en varios módulos y en cada experimento un grupo de individuos aplica CBR en un módulo y PBR a otro módulo distinto. En los tres experimentos se obtienen diferencias en el esfuerzo requerido; en los dos primeros el PBR requiere mayor esfuerzo. Sin embargo solamente en el primer caso encuentra que la diferencia es significativa a nivel estadístico.

Por último, en el artículo [8], que compara el rendimiento según la relación del revisor con el artefacto, el costo es casi la mitad cuando quien revisa el artefacto es el propio autor.

En suma, a partir de las respuestas a la pregunta RQ₃ concernientes al costo y tomando en cuenta el costo por defecto, se puede ver que la técnica de lectura impacta en este punto y CBR es una mejor opción en esos términos.

Acerca de los resultados relativos a la situación de que el revisor sea o no el autor del artefacto, habría que preguntarse si pueden generalizarse para tipos de técnicas de lectura que no sea el basado en especificación formal, para lo que se podría realizar análisis específicos en tal contexto. De todos modos, parece razonable que el costo sea menor cuando el autor revisa su propio trabajo, ya que el revisor está familiarizado con el artefacto. Y que, por otro lado, la efectividad se observe superior cuando se revisa el artefacto desarrollado por otro individuo, debido a que el revisor tiende a ser más crítico y riguroso en ese caso.

En general, si bien el esfuerzo requerido por estas revisiones puede resultar considerable, estas permiten identificar defectos de forma temprana e incluso evitar la inyección de defectos. Ello sugiere que el costo de las revisiones sería conveniente comparado al costo de detección tardía y retrabajo. Otros autores discuten estos puntos en el mismo sentido [Boehm y Basili, 2005; Glass, 2002; Wong, 2006].

Factores que influyen en el rendimiento de las revisiones individuales

Finalmente, las respuestas a la pregunta RQ₄ se muestran en el cuadro 23. Como se puede ver allí, las respuestas que se hallan en los artículos abarcan distintos factores, como la técnica de lectura utilizada, la lectura de documentos de especificación de requerimientos durante la revisión, la técnica cognitiva o patrón de búsqueda, la experiencia en revisiones de software por parte de los individuos, el revisor como autor del artefacto y, por último, el uso de una herramienta de soporte a la revisión.

En particular, los artículos [1] y [3] concluyen, luego de realizar experimentos, que existe una diferencia significativa en los resultados de la revisión de software, que depende de la técnica utilizada para leer los documentos. En estos dos casos, se llegan a conclusiones opuestas. En [1], donde el artefacto es diseño, se encuentra que el costo por defecto es menor al aplicar CBR que PBR. Por otro lado, en [3], donde el artefacto es código, se concluye que aplicando PBR el costo por defecto es menor que al utilizar CBR.

En el estudio [2], se observa que el desempeño mejora cuando se consulta durante más tiempo el documento de especificación de requerimientos mientras se revisan los documentos de diseño. Específicamente, para encontrar defectos

Respuesta	Artículos
Técnica de lectura	[1], [3]
Lectura de documento de requerimientos	[2]
Técnica cognitiva de búsqueda de defectos	[4]
Experiencia del revisor	[7]
Autor del artefacto	[8]
Esfuerzo	[11]
Uso de herramienta	[12]

Cuadro 23: Artículos por respuestas a la pregunta RQ4

del tipo omisión de requerimientos es necesario leer durante más tiempo el documento de requerimientos.

En el artículo [4] no se distingue una estrecha relación entre la experiencia del revisor en trabajar con diagramas y la efectividad de identificación de defectos, si bien es positivo que el revisor tenga experiencia o que tenga entrenamiento formal en diagramas de diseño antes de la aplicación de la revisión. Por otro lado, se encuentra que sí tiene impacto en la efectividad la técnica cognitiva de búsqueda de defectos utilizada al revisar los diagramas. Aquellos patrones que dan la posibilidad de construir relaciones entre diagramas y formular un modelo mental de la tarea permiten alcanzar una alta efectividad.

En el trabajo [7], la experiencia se identifica como un factor que influye en la efectividad. Se observa que los resultados obtenidos por profesionales son mejores en términos de efectividad que los obtenidos por estudiantes.

En el artículo [8] se compara el rendimiento de la revisión cuando el autor del artefacto es el revisor y cuando quien revisa es una persona diferente al autor. A partir de dicha comparación, se encuentra que si el revisor es quien implementó el programa, tanto la efectividad como el costo son menores. Se reflexiona que el costo puede resultar menor debido a la familiaridad con el código y con la construcción del programa. Acerca de la efectividad, se sostiene que el revisor tiende a pensar que su propio programa es correcto en casi todos los aspectos. Este exceso de confianza puede provocar que el revisor no haya prestado atención a detalles complejos del programa.

El estudio [11] presenta un modelo teórico de relación entre la cantidad de defectos detectados, el tamaño del artefacto y el esfuerzo de revisión. Allí se encuentra que la cantidad de defectos detectados está determinada por el esfuerzo en detectar defectos más que por el tamaño del artefacto.

Finalmente, en el trabajo [12] se observa que el uso de una herramienta de soporte tiene un impacto positivo en el costo (esfuerzo por defecto). En particular, esta herramienta fue construida específicamente para dar soporte al proceso de revisión basada en especificación formal, permitiendo comparar especificaciones en lenguaje de especificación formal con programas escritos en Java.

Como es indicado por el cuadro y la descripción de las respuestas de cada artículo a la pregunta RQ4, estas respuestas identifican que los factores con mayor impacto positivo sobre el rendimiento de las revisiones individuales de software son: la técnica de lectura utilizada, el uso de documentos adicionales como el documento de requerimientos, el patrón de búsqueda utilizado para encontrar defectos, la experiencia, la relación entre el revisor y el artefacto (siendo positivo para la efectividad cuando alguien distinto al autor es quien realiza la revisión y siendo positivo para el costo cuando el autor es quien revisa el artefacto), el esfuerzo empleado para detectar defectos y el uso de una herramienta de soporte.

4.3.3 *Discusión*

En síntesis, podemos reunir los datos obtenidos y enunciar los siguientes comentarios. Primero, la motivación principal para el uso de las revisiones de software es la mejora de calidad y la reducción de costos de detección y remoción de defectos. Sin embargo, existen muchos otros beneficios del uso de estas técnicas, que abarcan el desarrollo de la cohesión de equipo, la búsqueda de soluciones alternativas y la transferencia de conocimiento.

Por otro lado, la supuesta falta de guías específicas o procesos sistemáticos para los enfoques de revisión pueden desalentar la incorporación de estas prácticas. En cuanto a este punto, en el trabajo de Glass [2002] se analizan algunas razones por las que las inspecciones no se utilizan de forma masiva. Entre ellas se destaca la falta de motivación debido a la escasez de grandes interesados en su uso y a la falta de surgimiento de cosas nuevas acerca de las inspecciones; el hecho de que son vistas como algo que forma parte de la parte trasera del ciclo de desarrollo; el hecho de que a pesar de ser efectivas, las inspecciones requieren mucho trabajo mentalmente intenso que puede resultar extenuante.

A partir de los resultados de efectividad y costo podemos señalar que estos sugieren que las revisiones de diseño detallado y de código presentan valores similares e incluso mejores que otras técnicas de detección de defectos como el *testing* tradicional, sobre todo cuando el artefacto revisado es diseño. A propósito, es pertinente mencionar otro trabajo que realiza un análisis de estudios empíricos acerca de métodos de detección de defectos [Runeson et al., 2006].

Allí no se llega a una conclusión clara, aunque se identifican algunos conceptos que tenderían a ser verdaderos. Dentro de esos conceptos, aquellos que se refieren a diseño y código muestran que las inspecciones son más eficientes y más efectivas que el *testing* funcional para defectos en diseño. Sin embargo, en cuanto a defectos en código, el *testing* funcional y estructural están mejor considerados en la mayoría de los estudios analizados.

Por otro lado, algunos de los hallazgos y reflexiones se encuentran en consonancia con los encontrados en otras investigaciones dentro del PSP. Los datos arrojados por este estudio muestran que los valores más altos de efectividad para la revisión de diseño se encuentran entre 60 % y 70 % y para código entre 40 % y 60 %. Estos datos coinciden con los datos expuestos en Vallespir y Nichols [2016], donde se señala que dentro del PSP la detección de defectos de la revisión de diseño es del 50 % y la revisión de código del 60 %. En ese trabajo también se sugiere combinando las técnicas de revisión con el *testing* unitario se pueden detectar el 80 % de los defectos. Lo mismo señala el trabajo de Ferguson et al. [1997] donde se hallan datos de detección de defectos del 80 % al aplicar el proceso de PSP en algunos proyectos bajo estudio en la industria. Esto es confirmado por Paulk [2010] al encontrar que la calidad del producto mejora en un 79 % al utilizar las prácticas del PSP.

Por otra parte, Kemerer y Paulk [2009] postulan que los procesos disciplinados que siguen prácticas recomendadas pueden mejorar el desempeño. Este estudio empírico verifica que si se dedica tiempo suficiente de preparación para revisores e inspectores, se puede producir un mejor desempeño. En términos numéricos, los datos que se manejan son acordes a los mejores resultados encontrados para revisión de código en los artículos seleccionados en nuestro trabajo, donde la tasa de detección se encuentra en el entorno del 60 %. En ese estudio también se hace referencia a que los valores de efectividad para diferentes técnicas de revisión varía entre 30 y 90 por ciento. Los datos arrojados por nuestro trabajo se encuentran en su mayoría dentro de ese rango.

Finalmente, los factores como la experiencia, el conocimiento del artefacto a ser revisado, la técnica de lectura utilizada, los patrones de búsqueda empleados y el esfuerzo dedicado a la revisión aparecen como los principales elementos influyentes en el éxito de la revisión. Esto da cuenta de que la cercanía con el dominio del problema, el artefacto bajo revisión y la experiencia previa en la práctica de revisiones de software es determinante en el buen resultado de la función principal de la revisión, que consiste en encontrar defectos de forma temprana. De la misma manera, la forma en que es llevada a cabo la revisión desde el punto de vista cognitivo, así como la técnica sistemática, son componentes relevantes de la *performance* de la revisión.

4.4 AMENAZAS A LA VALIDEZ

En esta sección se analiza la validez de la metodología utilizada en esta investigación.

En este trabajo, se presentan las amenazas a la validez tomando los tipos usuales que se utilizan en estudios primarios, tal como se sugiere en [Kitchenham et al. \[2015\]](#). Estos son: validez de constructo, validez interna, validez de conclusión y validez externa. Vale mencionar que quizás otras formas de estructurar las amenazas a la validez de una revisión sistemática o estudio de mapeo sean más adecuadas, tal como se puede ver en las experiencias de estudios secundarios que se encuentran en la literatura [[Jorgensen y Shepperd, 2007](#); [Petersen et al., 2015](#)]. No obstante, en este trabajo se opta por una propuesta clásica de presentación del análisis de la validez del estudio.

En términos de validez de constructo, una limitación relacionada con el proceso de mapeo sistemático tiene que ver con la formulación de preguntas de investigación amplias. Como es sabido, el objetivo principal de un mapeo sistemático está orientado a categorizar los artículos seleccionados y a identificar estudios representativos y no tanto a realizar la agrupación de evidencia empírica. Para obtener las respuestas a una pregunta y, a su vez, analizar la validez de los resultados empíricos, se deberían considerar preguntas de investigación más específicas, por ejemplo: ¿Cuál es la efectividad al aplicar revisiones individuales sobre código?; ¿Cuál es la técnica de revisión individual de código más eficiente? Asimismo, deberían ser abordadas con técnicas más comunes en el contexto de una revisión sistemática, por ejemplo, para combinar evidencia.

En cuanto a los criterios de selección de estudios, se decidió incluir estudios cuyo foco de investigación es otro si, de todas maneras, aportan información que permita responder las preguntas de investigación acerca de revisiones individuales. Este es el caso de los estudios que abordan procesos de revisiones o inspecciones clásicas, pero incluyen datos de desempeño de la etapa de preparación individual. Esto podría generar un sesgo en los resultados. Se optó asumir esta amenaza considerando la poca cantidad de estudios específicos en revisiones individuales.

Un aspecto que afecta al conjunto de estudios devueltos por la búsqueda es la construcción de la cadena de búsqueda. Para mitigar esta amenaza, la cadena de búsqueda se elaboró de forma iterativa, a partir de varias ejecuciones en distintos buscadores. De todas formas, ya finalizada la búsqueda y la selección de estudios, al momento de escribir la tesis se ha identificado un trabajo que estaba disponible en IEEE Xplore y resultaba adecuado para abordar la pregunta acerca de la no adopción [[Ciolkowski et al., 2003](#)]. Una vez identificado,

este trabajo fue incorporado como una referencia más de aquellos estudios que señalan la baja adopción de las revisiones de software en la industria. Ese trabajo podría haberse identificado si dentro del aspecto correspondiente a quien realiza la intervención sobre el artefacto se hubiera incluido el término "*practitioner*" y, si además, no se hubiera restringido el aspecto correspondiente al artefacto bajo revisión. Esto da cuenta de la importancia y dificultad de la tarea de encontrar el equilibrio entre especificidad y amplitud de la cadena de búsqueda.

En el contexto de la validez interna, como posible amenaza, se puede argumentar que este mapeo sistemático fue realizado únicamente por un individuo, pese a que la literatura aconseja que deben intervenir varios revisores. Debido a esto, podría existir un sesgo en los resultados. Como acción para disminuir esta amenaza, se construyó un protocolo de investigación, donde la cadena de búsqueda fue desarrollada de forma iterativa, ajustándose a medida que se realizaban búsquedas preliminares. A su vez, se realizó una revisión entre pares sobre el protocolo. Por otra parte, en lo que tiene que ver con la clasificación, el uso ambiguo de ciertos conceptos, como por ejemplo, aquellos que refieren al tipo de estudio realizado como experimento y estudio de caso constituye una amenaza a la correcta categorización de los estudios de acuerdo con el esquema de clasificación. Para afrontar esta amenaza, se realizó la lectura completa de los artículos con el fin de determinar de qué forma los autores empleaban los distintos conceptos.

Por otro lado, si bien se dispone de un proceso definido de selección, no es posible garantizar que se seleccionaron todos los estudios primarios relevantes. Con el fin de mitigar esta amenaza, se hizo un seguimiento de referencias de primer nivel. Es decir, se revisaron aquellos artículos que estaban en las referencias de los estudios relevantes seleccionados. Por otra parte, al haber utilizado buscadores automáticos, es posible que algunos estudios recientes no hayan sido indexados en la biblioteca electrónica. Otro factor que podría cuestionar la validez interna de este mapeo es que se realizó en un solo recurso digital. Por ello, sería necesario expandir este mapeo a otros buscadores como Scopus, ACM o Springer.

En cuanto a la validez de las conclusiones, las amenazas están relacionadas con la síntesis de la información y con la adecuación de esa síntesis como soporte a las conclusiones. Una herramienta que mitigó este tipo de amenaza fue el diseño de un formulario de extracción de datos, el cual garantiza la consistencia de los datos extraídos. Este formulario, además, fue nuevamente analizado y ajustado luego de comenzada la extracción. Por último, luego de extraerse to-

da la información, se realizó una nueva revisión de los artículos y de los datos extraídos para corregir o completar la información del formulario.

Acerca de la validez externa, hay que considerar que, dada la poca cantidad de artículos, no se pueden trasladar los resultados como afirmaciones que apliquen, por ejemplo, para el resto de la industria. De todas maneras, tal como se ha señalado en otros estudios de revisiones sistemáticas, las características del proceso sistemático llevado a cabo permiten que los resultados del estudio de mapeo puedan servir como referencia e inicio de próximas investigaciones.

A pesar de las amenazas expuestas, se debe recordar que el presente trabajo surge como un primer acercamiento al campo de estudio y, en ese contexto, es útil como punto de partida para otras investigaciones. Asimismo, a partir de las limitaciones identificadas, es posible reproducir el trabajo de una forma mejorada y, así, profundizar en los resultados obtenidos.

Luego del análisis de estas amenazas, cabe reflexionar acerca de la validez de los estudios de mapeos sistemáticos. A pesar de que algunas amenazas puedan limitar el alcance de un estudio de mapeo, estos problemas pueden ser abordados en etapas tempranas y así mitigar su impacto. De esa forma es posible conducir estudios de mapeo o revisiones sistemáticas de utilidad.

En general, creo que la investigación basada en evidencia otorga un aporte muy importante a la investigación en la ingeniería de software. Es claro que este enfoque no es ajeno a las limitaciones que surgen de las características de los estudios primarios de nuestra área, tales como problemas en la calidad de los reportes, en los resúmenes, en la taxonomía utilizada, etc. De todas maneras, la conducción de revisiones sistemáticas puede generar una motivación adicional para mejorar la forma en que se reporta la investigación en ingeniería de software.

Por su parte, los investigadores que realizan estudios secundarios deben tener en cuenta, durante la planificación y ejecución del estudio, las características de los estudios primarios para poder controlar y mitigar las amenazas relacionadas con ellas. La construcción del protocolo tiene una importancia muy grande para el éxito del mapeo o revisión. Por ello, es esencial que el proceso de desarrollo del protocolo sea iterativo e incluya una constante evaluación. Por otro lado, la documentación de la planificación, de las decisiones tomadas y de la conducción de la revisión sistemática es un elemento fundamental para que estos estudios puedan auditarse, repetirse y extenderse por parte de otros investigadores.

En particular, a mi juicio, los estudios de mapeo sistemático son una herramienta rigurosa y ordenada para identificar y categorizar el estado de la investigación en un contexto de interés. A su vez, el control sobre las posibles

amenazas desde la etapa de planificación es lo que potencia su validez. Este tipo de estudio resulta ser un buen comienzo en la investigación basada en evidencias así como también aporta para el aprendizaje de la investigación en sí misma.

4.5 CONCLUSIONES

En esta sección se resume la información más relevante de acuerdo con los resultados obtenidos.

Respecto de las características generales, la ejecución del estudio de mapeo sistemático arrojó 12 estudios finales. La información extraída de los artículos fue distribuida en distintas clasificaciones. Acerca del origen, la mayoría de los estudios provienen de conferencias. A su vez, la mayor parte de los estudios toman el código como artefacto objeto de la revisión. En cuanto al contexto, hay una supremacía en estudios provenientes de la academia.

La categorización de los estudios muestra que uno de los intereses de las investigaciones ha sido comparar las distintas técnicas de lectura que se pueden aplicar en las revisiones de software. En ese sentido, las técnicas son comparadas en general tomando como referencia la técnica CBR, considerada como una de las más extendidas. Dentro de las técnicas que se busca validar, se encuentra PBR. Esta técnica ha surgido para revisar requerimientos, aunque también se ha comenzado a aplicar en revisión de diseño y código.

Las respuestas a las preguntas de investigación pueden resumirse de la siguiente manera. En primer lugar, las respuestas a la primera pregunta, que refiere a los motivos del uso, señalan que las principales causas son la exigencia de altos niveles de calidad o la mejora de la calidad actual, la reducción de costos de detección y la corrección de defectos. A su vez, se utiliza como una forma de transferencia de conocimiento.

La segunda pregunta refiere a los motivos del no uso. Si bien se detallan algunos conceptos, tales como falta de procesos sistemáticos y guías precisas, no se han encontrado claras respuestas. Por esto, parece necesario indagar más en esta pregunta, por ejemplo, mediante otros estudios, a través de un mapeo que explore las guías y enfoques sistemáticos existentes o de una encuesta en la industria que consulte por la experiencia de los individuos en revisiones de software.

Respecto de la tercera pregunta, que indaga en la efectividad y el costo al aplicar revisiones individuales, se aprecian mejores resultados de efectividad cuando el artefacto es diseño. De todas maneras no es posible realizar una comparación de los resultados, entre diseño y código, ya que los experimentos son

diferentes. Por otro lado, la mayoría de las comparaciones entre técnicas de lectura no arrojan diferencias significativas. No obstante, sí se aprecian diferencias significativas cuando hay niveles diferentes de experiencia entre los sujetos.

Finalmente, la cuarta pregunta aborda los factores que impactan en el rendimiento de las revisiones individuales. Los factores abarcan la forma en que se realiza la revisión, la experiencia de los individuos en realizar este tipo de actividades y la relación entre el revisor y el artefacto bajo revisión.

A partir del análisis de los 12 artículos, es posible concluir que existe cierta escasez de investigación en revisiones individuales de software sobre código y diseño detallado. Si bien hay estudios que abordan cuestiones relativas a estas técnicas, el mapeo sistemático refleja un vacío en la investigación empírica llevada a cabo hasta el momento. En el mismo sentido, el hecho de que la mayoría de los estudios provengan de conferencias podría sugerir falta de madurez en el área de investigación.

A partir de estas apreciaciones se sugiere, por ejemplo, la necesidad de realizar más estudios empíricos que se enfoquen en la revisiones a nivel individual, con el fin de mejorar la fortaleza de los hallazgos. Además, se debería considerar, junto con la técnica utilizada, la interacción con otros factores, tales como experiencia, motivación, etc. Por otra parte, en términos de revisiones personales versus revisiones donde el revisor no es el autor, se sugiere, mediante nuevos estudios, analizar en más profundidad acerca de los resultados bajo estas dos condiciones en el contexto de distintas técnicas de lectura. De esa manera, por ejemplo, se podrá constatar bajo qué técnica de lectura se obtiene una mejor efectividad cuando el revisor es el propio programador.

En resumen, este estudio crea un antecedente de trabajo mediante un acercamiento a una metodología de investigación basada en evidencia, que sirve como referencia para el grupo de investigación y, además, como punto de partida para futuras investigaciones.

DIFICULTADES AL APLICAR MAPEOS SISTEMÁTICOS EN INGENIERÍA DE SOFTWARE

En este capítulo se exponen aspectos vinculados con las dificultades que pueden encontrarse al utilizar mapeos sistemáticos en el contexto de la ingeniería de software. Es de destacar que varios de los problemas afrontados en el presente trabajo coinciden con los identificados por otros autores en la literatura. Además, se incluyen lecciones aprendidas que se encuentran en las guías que se usaron para realizar este mapeo y cuestiones a tener en cuenta en la realización de revisiones y mapeos sistemáticos.

5.1 DIFICULTADES DEL MAPEO SISTEMÁTICO REALIZADO

Dentro de los obstáculos más significativos se pueden mencionar los problemas que surgen al utilizar las herramientas de búsqueda que proveen las bibliotecas digitales consultadas. Si bien el mapeo sistemático de este trabajo se ejecutó solo sobre IEEE Xplore, en la etapa de calibración del protocolo del mapeo se realizaron búsquedas en los siguiente recursos digitales: SpringerLink, Scopus, IEEE Xplore y ACM digital.

La primera dificultad encontrada fue el proceso de adaptación de la cadena de búsqueda para cada buscador. Esto puede resultar problemático cuando se tienen numerosos términos en la cadena de búsqueda, como es el caso de este estudio. A su vez, el hecho de disponer de demasiados términos nos acerca a las limitaciones de los motores de búsqueda, por lo que en algunos casos se debió partir la ejecución y luego se hizo la unión de los resultados. También, antes de ejecutar las búsquedas en cada biblioteca digital se debió tomar conocimiento de las reglas y características de cada una, lo que supuso un esfuerzo adicional.

En cuanto a los motores de búsqueda en particular, *ACM digital* tiene una sección *Advanced Search* que ayuda a construir consultas complejas. Sin embargo, en el momento de ejecución de la búsqueda, el mecanismo del formulario de búsqueda avanzada no era adecuado para ejecutar una cadena de búsqueda que tuviera composiciones AND de ORs en título y resumen, ya que no era posible crear un término compuesto por términos formados por operadores. Por ejemplo, era posible indicar que se quiere buscar en el título haciendo coincidir cualquiera de ciertas palabras o frases (aplicando el operador OR). Resultaba

así un primer término, pero no permitía combinar ese primer término con otro término mediante el operador AND (Puede verse la pantalla en la figura 2). Por esto, para realizar una búsqueda con una cadena de búsqueda de esas características, se debía identificar primero la sintaxis interna de una búsqueda cualquiera y luego, a través del cuadro de búsqueda común, escribir la cadena de búsqueda adaptada según la sintaxis de este buscador. Cabe mencionar que en la última versión consultada de la búsqueda avanzada de ACM sí es posible realizar una búsqueda de ANDs y ORs aplicadas a título.

The screenshot shows the ACM advanced search interface with the following sections and fields:

- Words or Phrases:** Find [any field] with . Options: all of this text (and), any of this text (or), none of this text (not).
- Names:** Find [any field] with names . Options: using all, any, none of the names.
- Keywords:** Find author's keywords . Options: using all, any, none of the keywords.
- Affiliations:** Find company or school . Options: using all, any, none of the affiliations.
- Publication:** Find publication . Options: using all, any, none of the names. Find publisher . Options: using any, none of the names. Published since [year] Published before [year]. In publication types: Journal Proceeding Transaction Magazine Newsletter.
- Conference:** Find sponsor names . Options: using all, any, none of the names. Find year (yyyy) . Options: using any, none of the years. Find location . Options: using any, none of the locations.
- Identification codes:** Find ISBN/ISSN . Find DOI .

Figura 2: Pantalla de búsqueda avanzada de ACM - setiembre 2014

Por otro lado, *SpringerLink* no permite buscar solamente por título y resumen, sino que busca a texto completo. Como consecuencia, se obtuvieron muchos resultados a partir de la cadena de búsqueda. Fue necesario, entonces, refinar la cadena para evitar combinaciones excesivas de palabras. La herramienta dispone de una sección de búsqueda avanzada donde, entre otras facilidades, se logra buscar por una lista de términos contenidos en el título, pero no es posible aplicar el mismo criterio al campo de resumen. De todas formas, dependiendo de los recursos de publicación que se busquen, puede ser una alternativa buscar

directamente en *Scopus*, pues esta base de datos indexa los principales *journals* de *Springer*.

Acerca de la biblioteca de *IEEE Xplore*, cuando la sintaxis de la búsqueda es incorrecta, por ejemplo, porque no se pone un operador, el motor de búsqueda no devuelve un mensaje de error, sino que devuelve un mensaje indicando que no se encontró ningún resultado, lo cual dificulta identificar la causa por la que no se obtienen resultados. De todas formas, la ayuda en línea del buscador señala cómo se tienen que utilizar los operadores y presenta varios ejemplos. Además, se advierte que se pueden utilizar como máximo quince términos. Esta biblioteca presenta una funcionalidad de búsqueda avanzada mediante comandos (*Command Search*), donde se puede ingresar la cadena de búsqueda directamente. Para utilizar esta funcionalidad, por ejemplo, sobre los campos de resumen y título, se deben aplicar los prefijos respectivos (*Abstract* y *Document Title*) antes de cada término de la cadena de búsqueda (como se ilustra en el ejemplo de búsqueda sobre el campo de resumen en la figura 3); luego se deben unir los resultados de la búsqueda por resumen y por título.

Advanced Search Options

Advanced Keyword/Phrases **Command Search** Citation Search Preferences

ENTER KEYWORDS, PHRASES, OR A BOOLEAN EXPRESSION
Note: Use the drop down lists to generate the correct Operator and Data Field Codes.
This wizard will NOT build your expression. [View examples of how to write a boolean search string](#)

Search : Metadata Only Full Text & Metadata

Data Fields Operators

(((\"Abstract\":software) AND (\"Abstract\":personal OR
\"Abstract\":individual) AND (\"Abstract\":review OR \"Abstract\":inspection
OR \"Abstract\":examination) AND (\"Abstract\":code OR
\"Abstract\":program OR \"Abstract\":design)))

SEARCH GUIDELINES
Operators need to be in all caps
– i.e. AND/OR/NOT/NEAR.

Asterisk wildcards cannot be
used within quotes or with the
NEAR/ONEAR operators.

There is a maximum of 15
search terms.

Reset All **SEARCH**

Figura 3: Pantalla de búsqueda por comando de IEEE

En cuanto a *Scopus*, esta biblioteca podría usarse sin necesidad de buscar en el resto de las librerías indexadas. Para esto, deben identificarse las revistas y conferencias de interés principal y se debe verificar que todas las ediciones estén en *Scopus*. Por su parte, las ediciones faltantes deben ser recogidas por otros medios. Si bien *Scopus* indica cuáles son las publicaciones que están indexadas, debe leerse la lista de manera detallada para verificar para qué años está indexado cada conferencia o revista.

Acerca de los motores de búsqueda en ingeniería de software, la literatura señala dificultades relacionadas con ellos que coinciden con las descritas en este trabajo. Se indica que los motores de búsqueda no están diseñados para soportar revisiones o mapeos sistemáticos. Esto implica que los investigadores de ingeniería de software deben realizar búsquedas dependientes de los recursos electrónicos. Es decir, es necesario adaptar la cadena de búsqueda para cumplir con los requerimientos de las diferentes bases electrónicas [Kitchenham y Charters, 2007; Brereton et al., 2007]. Así, las restricciones y limitaciones de las bases de datos electrónicas hace que la búsqueda se dificulte [Imtiaz et al., 2013]. Incluso se señala que para una revisión completa, las facilidades de búsqueda de las bibliotecas digitales no son suficientes [Jorgensen y Shepperd, 2007].

Otro punto que conlleva dificultades al llevar a cabo el mapeo sistemático tiene que ver con el chequeo del protocolo desarrollado y con el desconocimiento de si este es adecuado para encontrar los estudios relevantes. Sin una etapa de calibración del protocolo, recién se descubrirá la eficacia del protocolo una vez ejecutadas todas las búsquedas. Para afrontar esta amenaza, se realizó una construcción iterativa del protocolo; se realizaron búsquedas en diferentes bibliotecas digitales; y se ajustaron la cadena, los criterios de selección y la estrategia de extracción de datos. Además, se realizaron revisiones del protocolo por parte de un experto, del tutor de la tesis y de un par.

En cuanto a la construcción del protocolo, la literatura señala que la realización de pruebas es esencial. Asimismo, se sostiene que se encontrarán errores en la recolección de datos y procedimientos de agrupación. Además, las pruebas permitirán ajustar la metodología seguida para atender las preguntas de investigación, incluyendo los formularios de extracción de datos y métodos de síntesis [Kitchenham y Charters, 2007]. Conducir pruebas de la cadena de búsqueda es beneficioso para asegurar que se accede a evidencia representativa. Por ejemplo, se verifica que se obtienen resultados conocidos, con un balance entre una cadena de búsqueda general y específica, o incluso se pueden generar nuevas cadenas a partir de la lectura de referencias incluidas en los estudios [Imtiaz et al., 2013].

En lo que hace al proceso de búsqueda, haber obtenido una gran cantidad de resultados candidatos en relación al tiempo y a los recursos disponibles (en este caso una sola persona) ha constituido un problema para el procesamiento de los estudios. El alto número de resultados obtenidos se debió a que la cadena de búsqueda fue construida para cubrir todos los términos posibles relacionados con la revisión individual de software. Una de las opciones para reducir la cantidad de resultados fue ajustar la cadena de búsqueda. Sin embargo, dado que es posible que los estudios específicos dentro del tópico del mapeo sean pocos, es necesario abarcar la mayor cantidad posible de artículos candidatos. Por esto, si bien se ajustó iterativamente la cadena de búsqueda, se intentó mantener una cadena abarcativa.

Como herramienta para mitigar la carga de trabajo, se utilizó como soporte una planilla electrónica generada a partir de la exportación de los resultados desde la biblioteca digital IEEE Xplore. Esto permitió visualizar de forma resumida los resultados y permitió también trabajar aplicando filtros en el mismo documento sobre los artículos relevantes seleccionados. Luego, el formulario de extracción de datos diseñado se instanció en la misma plantilla. De esa forma, se dispuso de una parte de los datos a extraer de forma automática. Como consecuencia, esta plantilla contiene la información histórica, los datos extraídos y la clasificación de los resultados.

Según [Kitchenham et al. \[2015\]](#), existen algunas opciones para enfrentar el problema que supone disponer de una gran cantidad de estudios candidatos cuando el proceso del mapeo sistemático es llevado a cabo por una sola persona. Primero, se sugiere reconsiderar las preguntas de investigación y revisar la cadena de búsqueda. Cuando el problema persiste, una opción es reclutar más personas para realizar la revisión de los artículos, aunque los nuevos revisores necesitarán tiempo para interiorizarse en el procedimiento de revisión planificado. La segunda opción considera el uso de una herramienta de minería de texto para identificar el conjunto de artículos con mayor posibilidad de ser relevantes para las preguntas de investigación y excluir aquellos con baja probabilidad de relevancia. En [Dieste et al. \[2008\]](#) también se menciona este problema, al señalar que se debe maximizar la cantidad de material relevante pero economizando el esfuerzo de descarte. Se sugiere aplicar una evaluación de calidad temprana para descartar estudios candidatos y así disminuir el conjunto de artículos que llegan a la etapa de extracción de datos. Algo a tener en cuenta sobre la evaluación temprana de calidad, es que, al no estar normalizada, la evaluación puede llegar a ser demasiado o poco exigente.

En la literatura también se identifican otros problemas relacionados con el proceso de búsqueda. Uno de los principales problemas surge al realizar una

búsqueda de un tópico que no suele ser incluido como tópico principal de investigación. Es decir, se refiere a casos en que el tópico buscado difícilmente sea mencionado en el título, resumen o palabras clave. La consulta de aspectos detallados de un proceso de investigación o de un tópico requiere buscar en el texto completo de los estudios de investigación, lo cual no es soportado por los servicios de indexación ni por todas las bibliotecas digitales [Kitchenham et al., 2015]. En el caso de este trabajo, se intentó que la cadena de búsqueda no fuera demasiado específica. De todas maneras, si bien los artículos contenían las palabras clave buscadas en el título o resumen, la mayoría de ellos se centraban en otros tópicos, tales como inspecciones tradicionales o revisiones grupales. Algunos de esos trabajos fueron incluidos debido a que igualmente aportaban datos a nivel de revisión individual.

En el contexto de extracción de datos, a pesar del diseño de un formulario de extracción para ordenar los datos relevados, la obtención de cierta información detallada resultó compleja, ya que se debió leer varias veces los estudios para identificar e interpretar dichos datos. En este orden, por ejemplo, ante la falta de uniformidad en el uso de los términos fue necesario mantener anotaciones de las decisiones acerca de considerar ciertos términos como iguales. De todos modos, esto no evitó tener que volver a leer los artículos en caso de dudas. En Dieste et al. [2008] se analizan estos mismos problemas vinculados al uso de plantillas de extracción de datos. En dicho trabajo se sugiere que sería útil disponer de un enlace entre la plantilla de extracción y cada estudio primario de forma de poder consultar la información utilizada durante la extracción de datos, así como también disponer de un glosario de términos que incluyera las decisiones tomadas.

Otro punto que resultó problemático fue la agrupación de los estudios en el esquema de clasificación elegido. Particularmente, en lo que refiere a la categorización por tipo de investigación empírica, se identificó el uso ambiguo de términos como experimento o estudio de caso y se debió leer completamente cada trabajo para confirmar el tipo de investigación llevada a cabo. En Petersen et al. [2008] se describe este tipo de problemas en la clasificación de estudios y se señala que el resumen puede omitir información importante, por lo que es conveniente realizar una lectura en profundidad para clasificar los estudios y así mejorar la validez de los resultados.

En lo que refiere a las iteraciones de preparación del protocolo antes de disponer de su última versión, dentro de las guías no se especifica de forma clara cuál es el punto de salida de dichas iteraciones. En este trabajo se realizaron sucesivas iteraciones de ajuste de la cadena de búsqueda, criterios de selección y extracción de datos. Se consideró detener las iteraciones luego de haber

realizado varias pruebas en las distintas bibliotecas digitales y constatar que los resultados no dejaban fuera artículos identificados como candidatos; que los criterios de inclusión y exclusión permitían llevar adelante la selección de estudios de forma correcta; y que el formulario de extracción de información abarcaba los datos a recolectar. Además, en el mismo sentido, si bien se señala que deben reportarse todas las desviaciones del protocolo, en las guías seguidas no se define por completo si en el reporte debe incluirse el protocolo inicial además del protocolo final (mencionando todas las variaciones y sus respectivas justificaciones) o si, por el contrario, basta con presentar el protocolo final y las decisiones relevantes tomadas.

Estas cuestiones son identificadas por otros trabajos, tal como es reunido en [Dieste et al. \[2008\]](#), donde primero se considera el desarrollo del protocolo como un punto crítico. A partir de esto, se menciona que es necesario establecer lineamientos tanto para el criterio de parada del estudio piloto, previo a desarrollar el protocolo final, como para el reporte de las modificaciones sufridas por el protocolo de investigación original.

Es necesario mencionar que la falta de conocimiento específico en el tópico bajo revisión o, más aún, la falta de experiencia en experimentación en ingeniería de software pueden provocar dificultades adicionales. En el presente trabajo la poca experiencia específica en experimentación ha generado un esfuerzo previo mayor, en pos de adquirir el conocimiento faltante para lograr un mejor desempeño en la realización de determinadas tareas. Algunas de ellas son el desarrollo de las preguntas de investigación, la definición de los criterios de selección, la extracción correcta de datos y la identificación de los tipos de investigación empírica utilizadas en los estudios.

Los problemas relacionados a la falta de conocimiento o experiencia también han sido señalados en [Dieste et al. \[2008\]](#). En dicho trabajo, se detallan algunas consecuencias de la falta de experiencia en experimentación, tales como mala definición de la pregunta de investigación, incorrecta aplicación de los criterios de selección, no poder identificar los elementos y características de los distintos tipos de estudios empíricos y no poder validar la calidad de dichos estudios. Por esto, se recomienda, por un lado, establecer requerimientos mínimos de conocimiento que debería tener el revisor y, por otro lado, definir un conjunto de roles con perfiles que se complementen para la realización efectiva de una revisión sistemática.

Como se acaba de mostrar, en esta sección se detallaron las dificultades más importantes afrontadas durante la ejecución de este estudio de mapeo sistemático, que coinciden con las reportadas por la literatura. Finalmente, se indicaron las recomendaciones y acciones realizadas ante estas dificultades.

5.2 OTROS PROBLEMAS RELACIONADOS CON LOS MAPEOS SISTEMÁTICOS

A propósito de los mapeos sistemáticos, la literatura destaca otros problemas junto con algunas recomendaciones que, a pesar de no haber sido identificados en este trabajo, merecen ser mencionados.

Primeramente, se debe evitar incluir múltiples publicaciones acerca de los mismos datos. Incluir duplicaciones generará sesgos importantes en los resultados. Para evitar esto, puede ser necesario tener que contactar a los autores para determinar si los reportes refieren al mismo estudio. En el caso de existir publicaciones duplicadas, se debería incluir la versión más completa [Kitchenham y Charters, 2007].

Acerca del procedimiento de búsqueda, la literatura señala que se debe seleccionar y justificar la estrategia de búsqueda que se consideró apropiada para la pregunta de investigación. Por ejemplo, se debe justificar en caso de que la búsqueda se restrinja a revistas y a conferencias específicas. Asimismo, es necesario buscar en diferentes recursos electrónicos, ya que en un solo recurso no será posible encontrar todos los estudios primarios [Kitchenham y Charters, 2007]. A su vez, en la estrategia de búsqueda se debe tener especial cuidado en la selección de palabras clave, sinónimos y términos alternativos. Como el material revisado depende de los términos utilizados para construir la cadena de búsqueda, el cubrimiento de estudios puede verse afectado si las palabras clave tienen demasiados sinónimos o términos alternativos desconocidos. Se recomienda identificar palabras clave y sinónimos mediante el análisis de artículos relevantes, así como también se sugiere excluir sinónimos relacionados [Imtiaz et al., 2013].

En el contexto del procedimiento de búsqueda, se identifica la combinación de búsqueda automática y manual como una forma de asegurar el cubrimiento de la evidencia relevante ya que, por un lado, la búsqueda que es solo automática requiere más esfuerzo para clasificar el material relevante a partir de una gran cantidad de artículos devueltos. Por otro lado, la búsqueda estrictamente manual requiere menos esfuerzo en la tarea de selección, pues se identifican artículos relevantes y se omiten artículos de baja calidad. Sin embargo, en lo global requiere demasiado esfuerzo y cabe la posibilidad de que se pierda evidencia relevante [Imtiaz et al., 2013].

A modo de garantizar la obtención de un conjunto relevante de evidencia empírica, se sugiere seguir una estrategia de dos fases. En primer término, se deberían analizar las referencias de los estudios para identificar nuevos trabajos. En segundo término, se deben contactar a los autores de los estudios seleccio-

nados con el fin de identificar nuevo material relevante que aún no haya sido publicado [Imtiaz et al., 2013].

En cuanto a la selección de estudios, un punto a tomar en cuenta es que las decisiones de inclusión pueden verse afectadas por el conocimiento de los autores, instituciones, revistas o año de publicación. Se sugiere remover esa información antes de tomar las decisiones de inclusión, aunque esto puede resultar costoso y no garantice la mejora de las revisiones [Kitchenham y Charters, 2007; Berlin et al., 1997]. Si bien en el presente trabajo esta información no se ocultó al momento de la selección de los artículos, no fue considerada para las decisiones de inclusión. En parte, porque es una primera aproximación al área de investigación y, en segundo lugar, porque dado el número reducido de artículos que podía arrojar el mapeo, la motivación de la selección era incluir todos los artículos relevantes según los criterios ya definidos.

Acerca de la extracción de datos, cuando hay un número importante de trabajos a revisar, puede ser de ayuda que uno de los revisores actúe como extractor de datos y otro como verificador. A su vez, los integrantes del equipo de revisión deben asegurarse de que comprenden el protocolo y el proceso de extracción de datos [Kitchenham y Charters, 2007; Brereton et al., 2007].

En lo que respecta a la síntesis de los datos, se identifican tres cuestiones. En primer lugar, las revisiones y los mapeos sistemáticos en ingeniería de software tienden a ser de naturaleza cualitativa, es decir descriptivos. En segundo lugar, aún cuando se recolecta información cuantitativa, puede no ser posible realizar meta-análisis sobre estudios en ingeniería de software, porque los protocolos de reporte de datos varía demasiado de un trabajo a otro. Por otro lado, si bien tabular los datos es un medio útil de agrupación, es necesario explicar cómo los datos agrupados responden las preguntas de investigación. Finalmente, al agregar evidencia se debe considerar el mecanismo apropiado para agrupar evidencia de diferentes tipos de estudios empíricos [Kitchenham y Charters, 2007; Brereton et al., 2007].

Con relación al reporte de una revisión o mapeo sistemático, el protocolo puede servir como una base para la construcción del reporte final, pero además este reporte debe reflejar las decisiones tomadas durante la planificación, ejecución y síntesis. Ante esa necesidad, resulta problemático disponer de todo el detalle para el informe final. Por ejemplo, cuando las decisiones son fruto de largos intercambios de correo o se encuentran distribuidas en notas de distintas reuniones. Por esto, se requiere mantener un registro formal y detallado de las decisiones tomadas durante el proceso de revisión, en especial las desviaciones del protocolo [Kitchenham y Charters, 2007; Brereton et al., 2007].

Algunas críticas hacia las guías tienen que ver con el hecho de que el proceso tiende a ser general. Por ejemplo, se señala que pese a que es preferible un proceso que permita flexibilidad, un enfoque demasiado abstracto puede generar la dependencia de la arbitrariedad de los revisores, lo cual probablemente favorezca los sesgos dentro del proceso [Dieste et al., 2008].

En cuanto a los antecedentes de ejecuciones de mapeos, existen trabajos que han analizado las experiencias de mapeos sistemáticos en ingeniería de software. Al comparar distintos mapeos han identificado algunos problemas. Por ejemplo, se observa que mapeos muy similares sobre el mismo tópico generan distintos resultados. En particular, ciertos mapeos con un mismo esquema de clasificación han categorizado los estudios de manera diferente. Además, algunos mapeos presentan preguntas de investigación muy específicas, lo cual no se corresponde con la amplitud de un mapeo sistemático [Wohlin et al., 2013].

En otro trabajo, se ha identificado cuáles son las guías seguidas para la realización de los mapeos sistemáticos. En cuanto a esto, se da cuenta de que en un mismo mapeo se utiliza una mezcla de varias guías. Esto muestra que las guías actuales no son suficientes y es necesario generar una guía que abarque el proceso completo de un estudio de mapeo sistemático. A su vez, se propone un método de evaluación de un proceso de mapeo. Dicho proceso consiste en una evaluación *rubric*, la cual presenta una lista de verificación de las actividades que deben realizarse en un mapeo sistemático y define niveles de cumplimiento [Petersen et al., 2015].

Como se acaba de ver, en esta sección se han señalado otros problemas encontrados en la literatura, así como recomendaciones para mitigar las distintas dificultades dentro del proceso de revisión o mapeo sistemático.

CONCLUSIONES Y TRABAJOS A FUTURO

En este capítulo se presentan las consideraciones finales y se dan pautas de trabajos a futuro.

6.1 CONCLUSIONES

En este trabajo se describió el proceso del estudio de mapeo sistemático inicial llevado a cabo, para conocer la evidencia que existe en revisiones individuales de software sobre código y diseño detallado por fuera del PSP. Se discutieron los resultados del estudio de mapeo y las amenazas a la validez. También, se presentó un análisis de las dificultades encontradas al aplicar esta metodología de investigación.

Si bien dentro del PSP distintos trabajos señalan que la introducción de las actividades de revisión de diseño detallado y de revisión de código mejora la calidad del software y logra una reducción del costo de retrabajo [Vallespir y Nichols, 2016; Kemerer y Paulk, 2009; Hayes y Over, 1997], estas prácticas no parecen ser utilizadas de forma masiva en la industria [Laitenberger et al., 2002; Macchi y Solari, 2013; Ciolkowski et al., 2003].

A partir de esta realidad, en esta investigación se exploraron los estudios existentes sobre revisiones individuales de software aplicadas a código y diseño detallado, por fuera del PSP, tanto a nivel académico como de trabajos que muestren la experiencia en la industria. En ese sentido, se plantearon las preguntas de investigación y se consideró un enfoque de investigación basada en evidencias. Las preguntas de investigación abordan los siguientes puntos: motivos que llevan a una organización a usar las técnicas de revisiones individuales; motivos por los que las revisiones individuales no sean adoptadas; evidencia acerca de efectividad y costo de ejecutar las revisiones individuales; e identificación de factores que tengan impacto en el rendimiento de las revisiones individuales de software.

Dado el objetivo de este trabajo, se realizó un primer estudio de mapeo sistemático en busca de evidencia relevante. El proceso de mapeo sistemático fue detallado en términos de las preguntas de investigación definidas, la cadena de búsqueda utilizada, los criterios de selección y la clasificación de los estudios.

El estudio de mapeo arrojó doce estudios primarios como evidencia relevante, como producto del proceso de selección aplicado a un primer conjunto de 998 trabajos. La cadena de búsqueda utilizada fue construida a partir de la descomposición en aspectos, tal como sugieren las guías. A su vez, se intentó mantener la cadena lo más amplia posible para que resultara abarcativa, al mismo tiempo que incluía los términos y sinónimos relevantes para la búsqueda. Asimismo, fue necesario adaptar la cadena a cada motor de búsqueda utilizado tanto en la etapa de calibración como en la etapa de ejecución. Dentro del primer conjunto de trabajos retornados por la búsqueda se identificaron distintos patrones por los que fueron descartados al aplicar los criterios de selección. Por ejemplo, muchos trabajos incluían el término *inspection* en el resumen, pero no trataban la inspección a nivel individual. Este término en particular, fue mantenido para poder abarcar aquellos estudios que incluían datos de la etapa de inspección individual, ya que de no contemplar esa práctica la evidencia hubiera sido aún inferior. Otros trabajos se enfocaban en revisiones de otros artefactos de especificación de requerimientos o diseño que no pueden ser considerados como diseño detallado.

En el sentido de mejorar la efectividad de las estrategias de búsqueda, promover que los resúmenes respeten una estructura predefinida puede ayudar a los motores de búsqueda a devolver resultados más precisos y a los investigadores durante el proceso de selección y de extracción de datos. Aún más, el desarrollo de herramientas de procesamiento de lenguaje natural aplicadas a las metodologías de revisión sistemática podrían servir como soporte para validar los criterios de selección y la información extraída.

La información extraída de los artículos se presentó mediante cuadros que enseñan la distribución de los estudios según año de publicación, medio de publicación, institución involucrada, contexto del estudio y artefacto bajo revisión. Además, se definió un esquema de clasificación compuesto por dos grandes clasificaciones de los trabajos, una de ellas referente al tipo de investigación y la otra con foco en el tipo de técnica de lectura aplicada. La clasificación de los datos extraídos permite realizar las siguientes apreciaciones:

- Medio de publicación: en su mayoría provienen de conferencias. Esto puede sugerir falta de madurez en el área de investigación
- Contexto del estudio: la mayoría de los estudios se encuentran en ambiente académico
- Artefacto bajo revisión: la mayor parte de los estudios consideran el código como artefacto

- Estrategias empíricas: las más utilizadas son los experimentos y los estudios de caso.
- Tipo de investigación: la mayoría de los estudios realizan evaluaciones de técnicas ya implementadas en la práctica.
- Técnicas de lectura: la técnica utilizada como referencia de comparación es CBR. Dentro del resto de las técnicas se identifican, por ejemplo, Ad-Hoc (ausencia de técnica o no se deja a elección del sujeto), PBR, basada en especificación y basada en herramienta.

A partir de los resultados obtenidos, en la búsqueda en IEEE Xplore, resulta llamativa la poca investigación que existe sobre las revisiones a nivel individual por fuera del PSP, si se toma en cuenta los excelentes resultados que han proporcionado dentro del PSP. Incluso, además de ser pocos en cantidad, se trata de investigaciones que difieren en el foco.

En lo que respecta a las preguntas de investigación, se realizó un análisis de las respuestas a cada pregunta a partir de los estudios primarios seleccionados. Las respuestas más importantes pueden resumirse de la siguiente manera:

¿Por qué motivos se decide realizar revisiones individuales de código y diseño detallado?

Las respuestas encontradas abarcan las siguientes razones: mejora de calidad, reducción de costos, proyectos de gran escala que insumirán enormes recursos para detectar y corregir defectos, ejecución en etapas tempranas, transferencia de conocimiento y exigencia de alta calidad o del proceso de desarrollo seguido.

Se puede concluir que los motivos están, en algunos casos, relacionados con la exigencia del marco de trabajo o de los requerimientos del proyecto. También, se decide conducir revisiones cuando existe una situación de baja calidad del software y se debe lograr una mejora en dicho sentido. Por otro lado, es esperable que la posibilidad de poder aplicar las revisiones en etapas tempranas sea un motivo encontrado en los estudios, debido a que esta característica provee un importante beneficio en términos de reducción de costos de detección y, sobre todo, de corrección. Por último, la utilidad de las revisiones de software también abarca la posibilidad de identificar oportunidades de mejora en los artefactos. Además, cuando se trabaja en equipos, se logra compartir conocimiento y fomentar la cohesión de grupo, y desarrollar el sentimiento de que, por ejemplo, el código pertenece a todos y no solo a un programador.

¿Cuáles son los motivos por los que las revisiones individuales sobre código y diseño detallado no son adoptadas en la industria del software?

Esta pregunta no ha podido ser abordada en profundidad, solo uno de los estudios menciona la falta de guías específicas o procesos sistemáticos.

Puede sugerirse que algunas razones para no adoptar las revisiones individuales de software surgen del desconocimiento de los importantes beneficios que conllevan. Sumado a esto, si en los casos en que se conducen revisiones, estas no son realizadas de una manera formal y sistemática pueden no obtenerse los mejores resultados. Cabría preguntarse si existe responsabilidad de la comunidad académica de ingeniería de software en una falta de transmisión a la industria, tanto de la importancia de las revisiones individuales como de las mejores prácticas para su conducción. Por último, podría suponerse que al igual que otras prácticas, la adopción de las revisiones individuales dentro de una organización depende en gran parte de un referente o grupo de personas que promuevan y lideren dicha adopción.

Resulta necesario, entonces, realizar algún trabajo con foco en esta pregunta. Por ejemplo, a través de un mapeo que explore las guías y enfoques sistemáticos actuales o de una encuesta en la industria que consulte por los motivos para no realizar revisiones individuales de código y diseño detallado.

¿Existe evidencia empírica acerca de la efectividad y/o del costo de las revisiones individuales de código y diseño detallado?

A partir de los resultados, se aprecian mejores resultados de efectividad cuando el artefacto es diseño. Por otra parte, si bien muchos estudios realizan comparaciones entre técnicas de lectura, la mayoría de esas comparaciones no arrojan diferencias significativas. En cuanto a las comparaciones entre sujetos con diferente nivel de experiencia sí se aprecian diferencias significativas.

Los valores de efectividad obtenidos, en general, se corresponden con los encontrados en otros trabajos. De todas maneras, tal como se discute en la presentación de los resultados del mapeo, hay elementos que generan una baja en la efectividad. Uno de esos elementos es que la checklist utilizada es genérica y, a pesar de que en algunos casos tiene una mínima adaptación al contexto o a defectos comunes publicados en la literatura, no se adapta a las características de la organización, equipo de desarrollo o individuo. Este punto es relevante, ya que las recomendaciones para mejorar la efectividad en revisiones basadas en *checklist* es mantener *checklists* en constante actualización y adecuadas a cada profesional. Otro de los elementos es que en algunos de los experimentos el tiempo para la revisión está acotado, lo que impacta en la cantidad de defectos que se pueden detectar.

Si bien, se han presentado resultados de efectividad y costo, no es el objetivo de este trabajo agrupar la evidencia, ya que en ese caso se deberían realizar

actividades de mayor análisis de los datos para determinar la validez de la evidencia y decidir de qué forma puede ser agrupada. Siendo esa tarea propia de un estudio de revisión sistemática de la literatura.

¿Qué factores influyen en la efectividad y el costo de aplicar revisiones individuales sobre código y diseño detallado?

A partir de esta pregunta se han recogido los siguientes factores. La forma en que se realiza la revisión (por ejemplo, técnica de lectura, lectura de otros documentos y uso de herramienta), la experiencia de los individuos en realizar este tipo de actividades y el hecho de que la revisión sea realizada por el propio autor o por un par.

Dada la característica de los estudios encontrados, donde en gran parte se realizan comparaciones de técnicas de lectura o evaluación de herramientas, es razonable esperar que esta pregunta sea atendida por casi todos los trabajos. Una vez identificados cuáles son algunos de los factores que impactan en el rendimiento, puede plantearse profundizar en estos factores con el objetivo de determinar las combinaciones más provechosas para las revisiones individuales, por ejemplo, comparar técnicas de lecturas cuando el revisor es el autor del artefacto. Algunos factores conocidos que no han sido mencionados por los estudios son las características de las *checklist* utilizadas, la motivación del individuo, el tipo de instrucciones seguidas para realizar las tareas o la complejidad del artefacto bajo revisión.

Si bien se ha podido realizar un análisis y discusión de las respuestas a las preguntas de investigación, la poca cantidad de estudios encontrados no permite generalizar estos resultados de forma concluyente. De todas maneras, tomando en cuenta que este es un primer estudio de mapeo de aproximación al tópico, se ha logrado el objetivo de identificar vacíos en la investigación en ingeniería de software y plantear posibles caminos de futuras investigaciones.

Acerca de la metodología, se identificaron las amenazas a la validez al estudio de mapeo realizado. Estas fueron detalladas en términos de validez de constructo, validez interna, validez de conclusión y validez externa. Las amenazas señaladas abarcan los siguientes puntos: el hecho de que el mapeo fue realizado por un solo revisor, la construcción de las preguntas de investigación y criterios de selección (al considerar trabajos que no tienen foco específico en revisiones individuales) y los pocos estudios primarios obtenidos. Para todas ellas se han realizado acciones de mitigación, salvo, por ejemplo, en considerar estudios con foco más general que ha sido asumida para poder acceder a la poca información existente. Dada las limitaciones enumeradas, resulta impor-

tante que este mapeo sea extendido a otros buscadores, incorporando mejoras que permitan reducir estas limitaciones.

Por otro lado, en el capítulo 5 se describieron las mayores dificultades encontradas en la ejecución del mapeo, así como lecciones aprendidas ubicadas en la literatura de referencia. Dentro de los problemas mencionados se destacan aquellos relacionados a las bibliotecas digitales; a la calibración del protocolo; a la cantidad de resultados iniciales; a la extracción de información y a la clasificación de los estudios. Parece existir un consenso en cuanto a las debilidades señaladas, así como en las propuestas para mitigar estas limitaciones.

En cuanto a la metodología, entiendo que, pese a presentar múltiples beneficios para la investigación basada en evidencia y a pesar de ser un proceso definido de forma sistemática, aún requiere madurar desde la práctica y continuar mejorando las guías de referencia a partir de la experiencia de los investigadores. Por ejemplo, con respecto a la construcción de la cadena de búsqueda podrían generarse pautas más específicas a partir de los tipos de preguntas o de los objetivos del mapeo sistemático. Por otra parte, es deseable una mejora en los buscadores digitales que faciliten su uso y den soporte a cadenas de búsqueda complejas. De la misma manera, la calidad de los estudios primarios tiene una incidencia en el desempeño de los mapeos y revisiones sistemáticas, lo cual sugiere una necesidad de mejora en el reporte y conducción de las investigaciones. Es importante recordar que los trabajos en ingeniería de software tienen características que difieren de aquellas disciplinas de donde surge el enfoque basado en evidencias. Por un lado, en lo que refiere a la taxonomía, se utilizan términos imprecisos o se crean nuevos términos de forma regular. En segundo lugar, los sujetos son activos y no simplemente reciben un tratamiento, muchas veces la evidencia depende de las características de los participantes. Finalmente, los artefactos también tienen su evolución y particularidades lo que cuestiona que la evidencia pueda ser comparada. Todas estas características deben ser consideradas y deberían incorporarse a las guías de referencia.

El estudio de mapeo sistemático realizado provee una visión general de la cantidad de estudios actuales, los tipos de investigación llevadas a cabo y los tipos de técnicas de lectura relacionadas con los estudios realizados, con el fin de identificar huecos de investigación y de establecer futuras investigaciones.

A su vez, mediante este trabajo de tesis se logra conducir un nuevo estudio de mapeo sistemático en ingeniería de software. A partir de esta experiencia se analizaron y discutieron aspectos de la metodología de investigación, reflexionando tanto en las características positivas como en las limitaciones o dificultades que presenta para el investigador.

Esta investigación brinda un marco de referencia para futuros estudios, al mismo tiempo que agrega información relevante para la comunidad sobre revisiones individuales de software.

6.2 TRABAJOS A FUTURO

En cuanto a posibles trabajos a futuro, es posible identificar algunas líneas de investigación. Por un lado, se visualiza una necesidad de investigación específica en lo que respecta a revisiones individuales o personales (esto es, sin estar incluidas dentro de inspecciones o revisiones convencionales) y por fuera de marcos donde son obligatorios como el PSP. Es decir, faltan estudios empíricos que analicen la aplicación de revisiones individuales de código y diseño detallado dentro del proceso de desarrollo.

Con respecto a los factores de éxito, existe espacio para continuar profundizando en el impacto de la experiencia, pero también en el impacto de otras habilidades blandas como la motivación tanto a partir del propio individuo como inducida por un líder.

A partir de los datos extraídos en este mapeo podrán plantearse otras revisiones sistemáticas de literatura, en la que se profundice en ciertos conceptos, como puede ser el rol de la experiencia en las revisiones de software o resultados concretos de cada técnica de lectura en la revisión de cada tipo de artefacto. En el mismo sentido, puede indagarse en cual técnica de lectura es más adecuada para que las revisiones personales resulten más efectivas.

Otros focos posibles de investigación pueden ser, por un lado, analizar qué características deben tener las *checklists* para adecuarse a los distintos tipos de revisión individual, así como la obtención de datos acerca de la forma más adecuada de complementar distintas técnicas de verificación dentro del proceso de desarrollo. Por otro lado, también puede resultar provechoso analizar los tipos de defectos detectados al aplicar las revisiones individuales relacionándolos con las características de la *checklist* utilizada.

Finalmente, más experimentos sobre las consecuencias de aplicar revisiones individuales dentro de ciclos de desarrollos convencionales servirían para generar mayor conocimiento de esta práctica y sus implicancias. Esto es, resta aún realizar estudios donde se incorporen progresivamente las prácticas de revisiones a nivel individual dentro de ciclos de desarrollo clásicos, de forma de poder estudiar los aportes de tales actividades en el ciclo completo.

BIBLIOGRAFÍA

- Bacchelli, Alberto y Bird, Christian. Expectations, outcomes, and challenges of modern code review. In *2013 international conference on software engineering*, pages 712–721. IEEE Press, 2013.
- Basili, Victor R y Selby, Richard W. Comparing the effectiveness of software testing strategies. *IEEE Transactions on Software Engineering*, (12):1278–1296, 1987.
- Basili, Victor R; Green, Scott; Laitenberger, Oliver; Lanubile, Filippo; Shull, Forrest; Sørumgård, Sivert, y Zelkowitz, Marvin V. The empirical investigation of perspective-based reading. *Empirical Software Engineering*, 1(2):133–164, 1996.
- Berlin, JA; Miles, CG; Crigiliano, MD; Conill, AM; Goldmann, DR; Horowitz, DA, y otros. Does blinding of readers affect the results of meta-analyses? results of a randomized trial. *Online J Curr Clin Trials*, 29, 1997.
- Biffi, Stefan y Halling, Michael. Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering*, 29(5):385–397, 2003.
- Biolchini, Jorge; Mian, P Gomes; Natali, A Candida Cruz, y Travassos, G Horta. Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, 679(05):45, 2005.
- Boehm, Barry y Basili, Victor R. Software defect reduction top 10 list. *Foundations of empirical software engineering: the legacy of Victor R. Basili*, 426, 2005.
- Brereton, Pearl; Kitchenham, Barbara A; Budgen, David; Turner, Mark, y Khalil, Mohamed. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4): 571–583, 2007.
- Budgen, David. Ase-eval: Software evaluation glossary-version 1.1, 2007. URL <http://community.dur.ac.uk/ebse/resources/ASE-EVAL-glossary.pdf>.
- Budgen, David; Turner, Mark; Brereton, Pearl, y Kitchenham, Barbara. Using Mapping Studies in Software Engineering. In *Psychology of Programming Interest Group 2008*, pages 195–204. Lancaster University, 2008.

- Burnstein, Ilene. *Practical software testing: a process-oriented approach*. Springer Science & Business Media, 2006.
- Ciolkowski, Marcus; Laitenberger, Oliver, y Biffl, Stefan. Software reviews: The state of the practice. *IEEE software*, (6):46–51, 2003.
- JasonCohen, S. T. *Best kept secrets of peer code review*. Smart Bear, 2013.
- Dieste, Oscar; Grimán, Anna, y López, Marta. Revisiones sistemáticas: Recomendaciones para un proceso adecuado a la ingeniería del software. In *XIII Jornadas de Ingeniería del Software y Bases de Datos*, pages 321–332, Oviedo, España, 2008. Universidad de Oviedo.
- Dunsmore, Alastair; Roper, Marc, y Wood, Murray. The development and evaluation of three diverse techniques for object-oriented code inspection. *IEEE Transactions on Software Engineering*, 29(8):677–686, 2003.
- Dybå, Tore y Dingsøy, Torgeir. Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9):833–859, 2008.
- Elminir, Hamdy K; Khereba, Eman A; Elsoud, Mohamed Abu, y El-Hennawy, Ibrahim. Application and evaluation of the personal software process. *International Journal of Basic and Applied Sciences IJBAS*, 9(10), 2009.
- Fagan, M. E. Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 38(2-3):258–287, June 1999. ISSN 0018-8670.
- Ferguson, Pat; Humphrey, Watts S; Khajenoori, Soheil; Macke, Susan, y Matvya, Annette. Results of applying the personal software process. *Computer*, 30(5): 24–31, 1997.
- Fernández-Sáez, Ana M; Genero, Marcela, y Chaudron, Michel RV. Empirical studies concerning the maintenance of uml diagrams and their use in the maintenance of code: A systematic mapping study. *Information and Software Technology*, 55(7):1119–1142, 2013.
- Freimut, Bernd; Punter, Teade; Biffl, Stefan, y Ciolkowski, Marcus. State-of-the-art in empirical studies. *Report: ViSEK/007/E, Fraunhofer Inst. of Experimental Software Engineering*, 2002.
- Glass, Robert L. *Facts and fallacies of software engineering*. Addison-Wesley Professional, 2002.

- Grazioli, F. y Nichols, W. A cross course analysis of product quality improvement with PSP. Technical report, 2012.
- Hayes, Will y Over, James W. The personal software process (PSP): An empirical study of the impact of PSP on individual engineers. Technical report, DTIC Document, 1997.
- Hetzl, William C. *An Experimental Analysis of Program Verification Methods*. PhD thesis, 1976. AAI7702047.
- Humphrey, Watts. The personal software process (PSP). Technical Report CMU/SEI-2000-TR-022, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2000.
- Humphrey, Watts S. *A Discipline for Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995. ISBN 0201847485.
- Humphrey, Watts S. White paper: The software quality profile. Technical report, Software Engineering Institute, October 2009. URL <http://www.sei.cmu.edu/library/assets/softwarequalityprofile.pdf>.
- Hungerford, Bruce C; Hevner, Alan R, y Collins, Rosann Webb. Reviewing software diagrams: A cognitive study. *IEEE Transactions on Software Engineering*, 30(2):82–96, 2004.
- IEEE. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, pages 1–84, Dec 1990.
- IEEE. IEEE standard for software reviews and audits. *IEEE STD 1028-2008*, pages 1–52, Aug 2008.
- Imtiaz, Salma; Bano, Muneera; Ikram, Naveed, y Niazi, Mahmood. A tertiary study: experiences of conducting systematic literature reviews in software engineering. In *17th International Conference on Evaluation and Assessment in Software Engineering*, pages 177–182. ACM, 2013.
- Jorgensen, Magne y Shepperd, Martin. A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, 33(1):33–53, 2007.
- Juristo, Natalia y Moreno, Ana M. *Basics of software engineering experimentation*. Springer Science & Business Media, 2013.
- Kan, Stephen H. *Metrics and models in software quality engineering*. Addison-Wesley Longman Publishing Co., Inc., 2002.

- Kemerer, Chris F y Paulk, Mark C. The impact of design and code reviews on software quality: An empirical study based on PSP data. *IEEE Transactions on Software Engineering*, 35(4):534–550, 2009.
- Khan, Abdul Kadir. Impact of personal software process on software quality. *IOSR Journal of Computer Engineering (IOSRJCE)*, 1(5):21–25, 2012.
- Kitchenham, Barbara y Charters, Stuart. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.
- Kitchenham, Barbara A; Dyba, Tore, y Jorgensen, Magne. Evidence-based software engineering. In *26th international conference on software engineering*, pages 273–281. IEEE Computer Society, 2004.
- Kitchenham, Barbara A; Budgen, David, y Brereton, P. The value of mapping studies: a participantobserver case study. In *14th international conference on Evaluation and Assessment in Software Engineering*, pages 25–33. British Computer Society, 2010.
- Kitchenham, Barbara Ann y Pickard, Lesley M. Evaluating software engineering methods and tools. *ACM SIGSOFT Software Engineering Notes*, 23(1): 24–26, 1998.
- Kitchenham, Barbara Ann; Budgen, David, y Brereton, Pearl. *Evidence-Based Software Engineering and Systematic Reviews*, volume 4. CRC Press, 2015.
- Laitenberger, Oliver. Studying the effects of code inspection and structural testing on software quality. In *ninth International Symposium on Software Reliability Engineering*, pages 237–246. IEEE, 1998.
- Laitenberger, Oliver y DeBaud, Jean-Marc. An encompassing life cycle centric survey of software inspection. *Journal of Systems and Software*, 50(1):5–31, 2000.
- Laitenberger, Oliver; Leszak, Marek; Stoll, Dieter, y El Emam, Khaled. Quantitative modeling of software reviews in an industrial setting. In *sixth International Software Metrics Symposium*, pages 312–322. IEEE, 1999.
- Laitenberger, Oliver; El Emam, Khaled, y Harbich, Thomas G. An internally replicated quasi-experimental comparison of checklist and perspective based reading of code documents. *IEEE Transactions on Software Engineering*, 27(5): 387–421, 2001.

- Laitenberger, Oliver; Vegas, Sira, y Ciolkowsky, M. The state of the practice of review and inspection technologies in germany. Technical Report ViSE-K/011/E, Virtuelles Software Eng. Kompetenzzentrum (ViSEK), 2002.
- Land, L Pek Wee; Jeffery, Ross, y Sauer, Chris. Validating the defect detection performance advantage of group designs for software reviews: Report of a replicated experiment. In *Australian Software Engineering Conference*, pages 17–26. IEEE, 1997.
- Macchi, Darío y Solari, Martín. Mapeo sistemático de la literatura sobre la adopción de inspecciones de software. In *Conferencia Latinoamericana de Informática (CLEI 2012)*, pages 1–8, 2012.
- Macchi, Darío y Solari, Martín. Diagnóstico del uso de técnicas de revisión de software en uruguay. In *XVI Congreso Iberoamericano en Ingeniería de Software (CIBSE 2013)*, page 14, 2013.
- McMeekin, David A; von Kinsky, Brian R; Robey, Michael, y Cooper, David JA. The significance of participant experience when evaluating software inspection techniques. In *Australian Software Engineering Conference*, pages 200–209. IEEE, 2009.
- Mujtaba, Shahid; Petersen, Kai; Feldt, Robert, y Mattsson, Michael. Software product line variability: A systematic mapping study, 2008.
- Myers, Glenford J. A controlled experiment in program testing and code walkthroughs/inspections. *Communications of the ACM*, 21(9):760–768, September 1978. ISSN 0001-0782.
- Nagoya, Fumiko; Liu, Shaoying, y Chen, Yuting. An investigation of the approach to specification-based program review through case studies. In *ninth IEEE International Conference on Engineering Complex Computer Systems*, pages 249–258. IEEE, 2004.
- Nagoya, Fumiko; Chen, Yuting, y Liu, Shaoying. An empirical study on a specification-based program review approach. In *International Conference on Dependability of Computer Systems*, pages 199–206. IEEE, 2006.
- Paulk, Mark C. Factors affecting personal software quality. *Institute for Software Research*, page 4, 2006.
- Paulk, Mark C. The impact of process discipline on personal software quality and productivity. *Software Quality Professional Magazine*, 12(2):15–19, 2010.

- Petersen, Kai; Feldt, Robert; Mujtaba, Shahid, y Mattsson, Michael. Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering*, volume 17. sn, 2008.
- Petersen, Kai; Vakkalanka, Sairam, y Kuzniarz, Ludwik. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.
- Pomeroy-Huff, Marsha; Cannon, Robert; Chick, Timothy A; Mullaney, Julia, y Nichols, William. The personal software process (PSP) body of knowledge, version 2.0. Technical report, DTIC Document, 2009.
- Porter, Adam; Siy, Harvey; Mockus, Audris, y Votta, Lawrence. Understanding the sources of variation in software inspections. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7(1):41–79, 1998.
- Pressman, Roger S. *Software engineering: a practitioner's approach*. McGraw-Hill, 8 edition, 2015.
- Rombach, Dieter; Münch, Jürgen; Ocampo, Alexis; Humphrey, Watts S, y Burton, Dan. Teaching disciplined software development. *Journal of Systems and Software*, 81(5):747–763, 2008.
- Runeson, Per; Andersson, Carina; Thelin, Thomas; Andrews, Anneliese, y Berling, Tomas. What do we know about defect detection methods?[software testing]. *Software, IEEE*, 23(3):82–90, 2006.
- Sabaliauskaite, Giedre; Matsukawa, Fumikazu; Kusumoto, Shinji, y Inoue, Katsuro. An experimental comparison of checklist-based reading and perspective-based reading for uml design document inspection. In *International Symposium on Empirical Software Engineering*, pages 148–157. IEEE, 2002.
- Shull, Forrest; Rus, Ioana, y Basili, Victor. Improving software inspections by using reading techniques. In *23rd International Conference on Software Engineering*, pages 726–727. IEEE Computer Society, 2001.
- Sommerville, Ian. *Software Engineering: (Update) (9th Edition) (International Computer Science)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2011. ISBN 0137035152.
- Sutherland, Andrew y Venolia, Gina. Can peer code reviews be exploited for later information needs? In *31st International Conference on Software Engineering - Companion Volume*, pages 259–262. IEEE, May 2009.

- Thelin, Thomas; Runeson, Per, y Wohlin, Claes. An experimental comparison of usage-based and checklist-based reading. *IEEE Transactions on Software Engineering*, 29(8):687–704, 2003.
- Travassos, Guilherme; Shull, Forrest; Fredericks, Michael, y Basili, Victor R. Detecting defects in object-oriented designs: Using reading techniques to increase software quality. In *14th ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*, volume 34, pages 47–56, New York, NY, USA, 1999. ACM.
- Uwano, Hidetake; Monden, Akito, y Matsumoto, Ken-ichi. Dresrem 2: An analysis system for multi-document software review using reviewers' eye movements. In *third International Conference on Software Engineering Advances, ICSEA'08*, pages 177–183. IEEE, 2008.
- Vallespir, Diego y Nichols, William. Quality is free, personal reviews improve software quality at no cost. *Software Quality Professional Magazine*, 18(2), 2016.
- Wieringa, Roel; Maiden, Neil; Mead, Nancy, y Rolland, Colette. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006.
- Wohlin, Claes; Runeson, Per; Höst, Martin; Ohlsson, Magnus C; Regnell, Björn, y Wesslén, Anders. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- Wohlin, Claes; Runeson, Per; Neto, Paulo Anselmo da Mota Silveira; Engström, Emelie; do Carmo Machado, Ivan, y De Almeida, Eduardo Santana. On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, 86(10):2594–2610, 2013.
- Wong, Yuk Kuen. *Modern Software Review: Techniques and Technologies*. IGI Global, Hershey, 2006.
- Wood, Murray; Roper, Marc; Brooks, Andrew, y Miller, James. Comparing and combining software defect detection techniques: A replicated empirical study. *ACM SIGSOFT Software Engineering Notes*, 22(6):262–277, Nov 1997. ISSN 0163-5948.
- Zelkowitz, Marvin V y Wallace, Dolores R. Experimental models for validating technology. *Computer*, 31(5):23–31, May 1998. ISSN 0018-9162.



APÉNDICE A: PROTOCOLO DEL MAPEO SISTEMÁTICO

En este apéndice se presenta el protocolo final del estudio de mapeo sistemático presentado en esta tesis.

A.1 MOTIVACIÓN Y ANTECEDENTES

A.1.1 *Equipo*

Por enmarcarse dentro de un trabajo de tesis de maestría, el equipo del estudio de mapeo sistemático está formado únicamente por el autor de esta tesis. De todas formas, se cuenta con la participación de un revisor experto y de un par para la revisión y evaluación del protocolo.

A.1.2 *Introducción y necesidad de un estudio de mapeo sistemático*

Las revisiones de software son procesos formales o informales, donde se examina un producto de software en busca de defectos o desviaciones con respecto a los estándares y especificaciones [IEEE, 2008].

Las revisiones de software se aplican a los distintos artefactos que se generan durante todo el ciclo de desarrollo de software. Pueden constituir documentos de requerimientos, modelos de diseño, código, entre otros. Asimismo, la utilización de las revisiones en software permite obtener importantes beneficios como la disminución en costo y la mejora en calidad y productividad [Kan, 2002].

Las revisiones de software a nivel individual son tenidas en cuenta por el Personal Software Process (PSP) [Humphrey, 1995]. Se ha observado que las revisiones personales resultan efectivas y eficientes para mejorar la calidad de un producto y la productividad individual [Pomeroy-Huff et al., 2009]. En ese sentido, existen numerosos estudios que presentan datos de las mejoras que introducen el uso de las prácticas del PSP [Hayes y Over, 1997; Elminir et al., 2009; Khan, 2012; Ferguson et al., 1997; Paulk, 2010, 2006; Grazioli y Nichols, 2012].

A pesar de los buenos resultados de la aplicación de las revisiones de software que, por ejemplo, se encuentran en el PSP, varios estudios sostienen que estas técnicas presentan una baja adopción en la industria del software [Lai-

tenberger et al., 2002; Macchi y Solari, 2012; Ciolkowski et al., 2003; Macchi y Solari, 2013].

Bajo esta perspectiva, un estudio de mapeo sistemático permitirá reunir la información existente acerca del uso de las revisiones individuales de software. En particular, en este trabajo interesa analizar la evidencia de las revisiones individuales en marcos donde no se aplique el PSP.

A.1.3 *Alcance del estudio*

Tal como se ha introducido, este mapeo sistemático abarca la evidencia acerca de las revisiones individuales de software aplicadas a código y diseño detallado, por fuera del marco del PSP. A su vez, pretende identificar cuestiones que no hayan sido investigadas, con el fin de proponer posibles caminos a seguir para futuros trabajos, cubriendo, de esa manera, los huecos en el tópico bajo estudio.

A.1.4 *Preguntas de investigación*

Para conocer la evidencia empírica de las revisiones individuales utilizadas fuera del marco del PSP, además de analizar el uso de dichas prácticas en la industria, se determina que las variables de respuesta a ser analizadas son: la utilización de la técnica; los motivos por los que la industria la utiliza o no; las métricas asociadas, tales como efectividad y costo; y los factores que influyen en el rendimiento de su uso. Debido a esto, se consideran las siguientes preguntas de investigación:

RQ.1. ¿Por qué motivos se decide realizar revisiones individuales de código y diseño detallado?

Justificación: Se desea conocer los factores que influyen en un organización para decidir utilizar revisiones individuales de software sobre código y diseño detallado.

RQ.2. ¿Cuáles son los motivos por los que las revisiones individuales sobre código y diseño detallado no son adoptadas en la industria del software?

Justificación: Se desea encontrar los motivos por los que una organización decide no utilizar revisiones individuales de software sobre código y diseño detallado.

RQ.3. ¿Existe evidencia empírica acerca de la efectividad y/o del costo de las revisiones individuales de código y diseño detallado? Un ejemplo de medida de efectividad es el porcentaje de defectos detectados sobre defectos totales. Un ejemplo de costo es el tiempo empleado en llevar a cabo la revisión por cada mil líneas de código (hs/Kloc).

Justificación: Se quiere saber qué tan efectivas y costosas son las revisiones individuales de software sobre código y diseño detallado. Y conocer si existen resultados comparables e incluso mejores que otras técnicas, en aquellos casos que sea factible la comparación.

RQ.4. ¿Qué factores influyen en la efectividad y el costo de aplicar revisiones individuales sobre código y diseño detallado?

Justificación: Es deseable conocer qué factores pueden influir en los resultados de efectividad y costo de la aplicación de las revisiones individuales. (Algunos factores posibles podrían ser: experiencia, programa bajo revisión, diseño detallado bajo revisión, motivación, contexto).

A.2 PROCESO

El proceso seguido para este estudio de mapeo sistemático se divide en tres fases: planificación, ejecución del mapeo y reporte de los resultados, y se basa en las guías [Kitchenham y Charters, 2007; Biolchini et al., 2005; Budgen et al., 2008]. La calibración del protocolo supone la iteración sobre varias de las etapas del proceso. En el cuadro 24 se muestran las fases mencionadas junto con sus respectivas etapas.

Fase	Etapas
Planificación	Identificación de la necesidad del mapeo Encargo del mapeo (opcional) Especificación de la(s) pregunta(s) de investigación Desarrollo de un protocolo Evaluación del protocolo (opcional)
Realización	Identificación de los artículos Selección de los estudios primarios Evaluación de la calidad del estudio (opcional) Extracción de datos y monitoreo Síntesis de los datos
Información	Especificación de los mecanismos de diseminación Formateo del informe principal Evaluación del informe (opcional)

Cuadro 24: Fases del estudio de mapeo sistemático

El proceso específico para este estudio de mapeo sistemático puede verse en la figura 4.

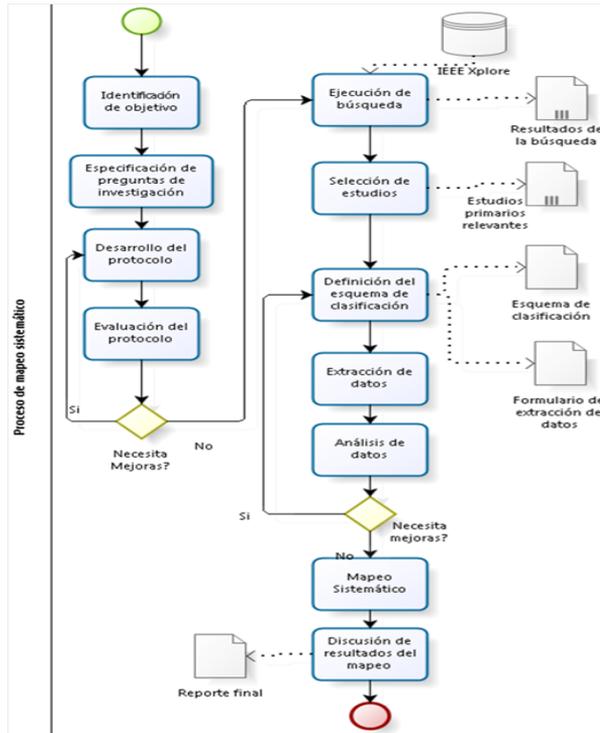


Figura 4: Proceso de mapeo sistemático

A.3 ESTRATEGIA DE BÚSQUEDA

Con el objetivo de llevar a cabo la búsqueda de los estudios primarios, se utiliza una estrategia de búsqueda a partir de la definición de una cadena de búsqueda a ser utilizada sobre determinados recursos electrónicos.

Además de la búsqueda automática, se plantea realizar una búsqueda manual sobre las referencias de primer nivel. Es decir, se aplica el criterio de selección a las referencias que se encuentran en los artículos seleccionados a partir de la búsqueda automática.

Teniendo en cuenta las preguntas de investigación formuladas anteriormente, se definen las palabras claves y sus respectivos sinónimos o palabras alternativas, considerando las siguientes características:

- ¿Quién aplica el método, es decir quién hace la revisión?
- ¿A qué artefacto se le aplica la revisión?
- ¿Cuál es la técnica o herramienta relacionada con la utilización de revisiones individuales?
- ¿Cuál es el producto desarrollado?

En el cuadro 25 se presentan los términos identificados mediante la descomposición en aspectos.

Quién	Qué artefacto	Técnica	Producto desarrollado
Personal	Code	Review	Software
Individual	Design	Inspection	
Author	Program	CheckList	
Developer	Diagram	Defect Detection	
Single	Document	Reading Technique	
Reviewer	Artefact	Code Reading	
		Error detection	
		Code evaluation	
		Find fault	
		Fault detection	
		Detecting defect	
		Design Defect	
		Find defect	
		Examination	

Cuadro 25: Términos a partir de descomposición en aspectos

De manera de realizar búsquedas automáticas, se compone la siguiente cadena de búsqueda, obtenida a partir de los términos de la descomposición en aspectos.

(personal OR individual OR author OR developer OR single OR reviewer) AND (review OR inspection OR checklist OR "defect detection" OR "reading

technique" OR "code reading" OR "error detection" OR "code evaluation" OR "find fault" OR "fault detection" OR "detecting defect" OR "design defect" OR "find defect" OR examination) AND (code OR design OR program OR diagram OR document OR artefact) AND software

La cadena debe adaptarse a la sintaxis y a las reglas permitidas por el motor de búsqueda utilizado.

En cuanto al recurso electrónico, la base de datos electrónica seleccionada para la extracción y análisis de información es IEEE Xplore.

Acerca del período de tiempo a ser cubierto, la consulta abarca desde el primer año disponible hasta el 19 de setiembre de 2014.

Por último, el proceso de búsqueda se evalúa durante la calibración del protocolo analizando un conjunto de los resultados devueltos.

A.4 CRITERIOS DE SELECCIÓN

A continuación se presentan los criterios de inclusión y exclusión para este estudio.

La selección se aplica según los criterios de inclusión y exclusión, mediante la revisión el título y el resumen de cada artículo. En caso de duda, se lee el artículo en su totalidad. Un estudio será seleccionado como un estudio primario relevante si satisface el criterio de inclusión y será descartado si satisface cualquiera de los criterios de exclusión.

A.4.1 *Criterios de Inclusión*

- IC₁ - El artículo presenta algún tipo de evaluación o análisis acerca de técnicas de revisión, procesos o herramientas relacionados con las revisiones individuales de código o diseño detallado de software

A.4.2 *Criterios de exclusión*

- EC₁ - El estudio se enfoca en la ejecución de revisiones o inspecciones pero no tiene datos de la fase de revisión o inspección individual
- EC₂ - El trabajo presenta experiencias o resultados de revisiones individuales pero aplicadas como parte del PSP
- EC₃ - El trabajo presenta un título duplicado

- EC4 - El trabajo realiza una revisión sistemática de literatura
- EC5 - El artículo no ha sido publicado
- EC6 - El trabajo solo está disponible en formato de resumen o presentación *PowerPoint*
- EC7 - El idioma del artículo no es inglés

A.5 EXTRACCIÓN DE DATOS

Se define un formulario de extracción de datos para recolectar toda la información relevante de acuerdo con los objetivos de este trabajo. Cada estudio se identifica mediante un número correlativo. Además de extraer el resumen y las conclusiones, los datos a extraer de cada artículo son:

- Título del artículo
- Tipo de artículo (Medio de publicación: revista, conferencia, libro)
- Autores
- Institución principal
- Año
- Foro
- Fuente (Recurso electrónico consultado)
- Contexto (Academia, Industria)
- Artefacto (Diseño, Código: los artículos identificados pueden tratarse de revisiones aplicadas sobre documentos de diseño o sobre código fuente.)

A su vez, se define un esquema de clasificación de los artículos. Este esquema está compuesto por los siguientes aspectos:

- Tipo de investigación
 - De acuerdo a estrategia empírica (Experimento, Estudio de Caso, Encuesta)
 - De acuerdo a clasificación propuesta por Wieringa [16]

- Tipo de técnica utilizada en la revisión (Checklist-based Reading, Usage-based reading, etc.)

Toda la información recabada es consolidada en una planilla electrónica en la que cada columna se corresponde con cada uno de los datos a extraer definidos en este protocolo.

La información es extraída en dos etapas: durante la aplicación de los criterios de selección con lectura de título y resumen y durante la aplicación de los criterios de selección con lectura de texto completo. Además, en caso de necesidad de un análisis más profundo se marca en el formulario para una posterior lectura completa, indicando la duda y la sección dentro del artículo, si correspondiera.

A.6 SÍNTESIS

Se presentan distintas distribuciones de los artículos, tales como evolución de la investigación en el tema a través de los años, medios de publicación, contexto e instituciones participantes de cada investigación y artefacto revisado.

Se decide realizar una categorización de los artículos según el esquema de clasificación definido: por tipo de investigación, por estrategias empíricas y por técnicas de lectura utilizadas.

A.7 LIMITACIONES DEL ESTUDIO

Este mapeo sistemático se realiza sobre un solo buscador digital, por lo que es altamente recomendable extenderlo a otros buscadores, tales como Scopus y ACM digital.

A.8 REPORTE

El reporte del estudio de mapeo sistemático se incluye como parte de la documentación del trabajo de tesis de maestría en ingeniería de software. La documentación presentada en dicha tesis también abarca la discusión acerca de la metodología de mapeos sistemáticos en la ingeniería de software.

A.9 EVALUACIÓN DEL PROTOCOLO

Se realiza la revisión del protocolo por parte de un experto en revisiones de la literatura, para determinar si el protocolo está definido de forma adecuada y si la ejecución se realiza correctamente.

A.10 AJUSTE DEL PROTOCOLO

El protocolo es calibrado mediante búsquedas en las fuentes utilizadas. Se analizan los primeros 5 artículos devueltos y se observa si el conjunto de artículos que se obtienen tienen coherencia con lo esperado por el protocolo y en dirección a las preguntas de investigación.

A.11 CRONOGRAMA

Finalmente, se presenta el tiempo estimado para las principales etapas del protocolo:

1. Planificación y construcción del protocolo: 20 días.
2. Evaluación del protocolo: 7 días.
3. Ajuste del protocolo: 30 días.
4. Ejecución del protocolo: 3 meses.

B

APÉNDICE B: CONCEPTOS DE INGENIERÍA DE SOFTWARE BASADA EN EVIDENCIAS

En este apéndice se presenta el reporte técnico del PEDECIBA 15-08. Se presentan conceptos de la ingeniería de software basada en evidencias. Además, se incluye una descripción de los métodos más utilizados dentro del paradigma, como son: las revisiones sistemáticas de la literatura, los estudios de mapeo sistemático y las revisiones terciarias.

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 15-08

**CONCEPTOS DE INGENIERIA DE
SOFTWARE BASADO EN EVIDENCIAS**

SEBASTIAN PIZARD
FERNANDO ACERENZA
VANESSA CASELLA
SILVANA MORENO
DIEGO VALLESPER

2015

Sebastian Pizard, Fernando Aceremza, Vanessa Casella,
Silvana Moreno, Diego Vallespir
ISSN 0797-6410
Tesis de Maestría en Informática
Reporte Técnico RT 1508
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República.
Montevideo, Uruguay, 2015

Conceptos de Ingeniería de Software Basada en Evidencias

Versión 1.0

Sebastián Pizard
Fernando Acerenza
Vanessa Casella
Silvana Moreno
Diego Vallespir

Reporte Técnico InCo/Pedeciba-2015 TR:15-08
Setiembre, 2015

Resumen

En este documento se presenta la definición de la Ingeniería de Software Basada en Evidencias y se describen sus objetivos. De forma más práctica, se incluye un resumen de los métodos más utilizados para la concreción de resultados, estos son: Revisiones Sistemáticas de Literatura, Estudios de Mapeo y Revisiones Terciarias.

Control de Versiones del Documento

Versión	# Reporte	Fecha	Cambios
Versión 1	Sin número	Setiembre 2015	Versión inicial

Índice general

1.Introducción.....	5
2.Ingeniería de Software Basada en Evidencias.....	5
3.Revisiones Sistemáticas de Literatura.....	6
3.1.Motivación.....	6
3.2.Definición y Etapas.....	6
3.3.Planificar la Revisión.....	7
3.3.1.Identificar la necesidad de una revisión.....	7
3.3.2.Puesta en marcha de la revisión.....	8
3.3.3.Especificar las preguntas de investigación.....	8
3.3.4.Desarrollar el protocolo de revisión.....	9
3.3.5.Evaluar el protocolo de revisión.....	11
3.4.Realizar la Revisión.....	11
3.4.1.Identificar la investigación.....	11
3.4.2.Seleccionar los estudios primarios.....	12
3.4.3.Evaluar la calidad de los estudios.....	13
3.4.4.Extraer datos.....	14
3.4.5.Sintetizar los datos.....	15
3.5.Informar sobre la revisión.....	16
3.5.1.Especificar los mecanismos de difusión.....	16
3.5.2.Formatar el informe principal.....	16
3.5.3.Evaluar el informe.....	18
4.Estudios de Mapeo.....	18
4.1.Diferencias entre un estudio de mapeo y una SLR.....	18
4.2.Esquemas de clasificación.....	19
4.3.Presentación de resultados.....	20
5.Revisiones Terciarias.....	22
6.Anotaciones sobre librerías digitales y motores de búsqueda.....	22
6.1.ACM Digital.....	23
6.2.IEEEExplore.....	25
6.3.ScienceDirect.....	28
6.4.Scopus.....	30
6.5.SpringerLink.....	31
7.Referencias.....	31
8.Material complementario.....	34

1. Introducción

Este reporte tiene como objetivo presentar los fundamentos de la Ingeniería de Software Basada en Evidencias (o ESBE por sus siglas en inglés), y en particular se incluye una breve guía para realizar revisiones sistemáticas en ingeniería de software. Se pretende que este documento sea utilizado por Proyectos de Grado de la carrera Ingeniería en Computación, Maestrías, Doctorados, etc. que se encuentren realizando trabajos de ESBE con el Grupo de Ingeniería de Software (GrIS). De esta manera los estudiantes de grado y posgrado pueden usar este documento como punto de partida para comprender la ESBE. Además, pueden incluir este documento como parte de su informe de proyecto o su Tesis evitando tener un enfoque distinto de la ESBE en cada Proyecto de Grado y/o Tesis.

En la sección 2 se presenta la definición de la ESBE así como sus motivaciones. En la sección 3 se describen las revisiones sistemáticas de literatura, en la sección 4 los estudios de mapeo. En la sección 5 se introduce a las revisiones terciarias. En la sección 6 se incluyen anotaciones sobre librerías digitales y motores de búsqueda.

Por último se incluyen 3 anexos seleccionados del sitio de ingeniería de software basado en evidencias – Evidence-Based Software Engineering (<http://community.dur.ac.uk/ebse/>). Estos corresponden a un plantilla para revisiones sistemáticas de literatura, una plantilla para estudios de mapeo y un ejemplo de una revisión sistemática de literatura. Los tres documentos no ha sido modificados y se reproducen bajo la licencia de Creative Commons (<http://creativecommons.org/licenses/by-sa/2.0/uk/>).

2. Ingeniería de Software Basada en Evidencias

La ingeniería de software basada en evidencias tiene como propósito mejorar la toma de decisiones relacionada al desarrollo y mantenimiento de software integrando la mejor evidencia actual de la investigación con experiencias prácticas y valores humanos. Para lograr esto, alienta un fuerte énfasis en el rigor metodológico involucrando los siguientes cinco pasos [1]:

- 1) Convertir un problema relevante o una necesidad de información en una pregunta que pueda ser respondida.
- 2) Buscar en la literatura la mejor evidencia para responder a esa pregunta.
- 3) Evaluar de forma crítica la validez, el impacto y la aplicación de la evidencia.
- 4) Integrar la evidencia evaluada con la experiencia práctica y los valores y circunstancias de los interesados.
- 5) Evaluar el desempeño y buscar maneras de mejorarlo.

La EBSE surge como una analogía a la práctica médica en donde el paradigma ha cambiado drásticamente la investigación. En esa rama de las ciencias se utilizan revisiones para resumir, por ejemplo, los resultados de una nueva droga presente en distintos medicamentos probados en diferentes lugares del mundo. La rigurosidad metodológica hace que los resultados sean más confiables ya que es posible estudiar el procedimiento llevado a cabo para su obtención así como también reproducirlo.

El mecanismo normal para la identificación y la agregación de evidencia de investigación es la Revisión Sistemática de Literatura.

3. Revisiones Sistemáticas de Literatura

3.1. Motivación

La necesidad de agregar evidencia desde múltiples estudios empíricos tiene varias motivaciones. Seguramente al comenzar una investigación se quiera utilizar toda la investigación previa relacionada con el fin de no partir de cero. También es posible querer contemplar los resultados de varios estudios particulares en conjunto para contestar preguntas amplias que no podrían contestarse de forma individual. Este proceso de recolección y síntesis debe contar con cierto estándar científico y rigurosidad metodológica.

Las revisiones sistemáticas de literatura permiten recolectar y sintetizar evidencia de distintas fuentes. La característica clave que las distingue de las revisiones tradicionales narrativas (o clásicas) es su intento explícito de minimizar las posibilidades de llegar a conclusiones erradas, que puedan resultar de sesgo en los estudios primarios o en el proceso mismo de revisión [2].

Sobre este tema:

- Una breve descripción de las revisiones sistemáticas de literatura en ingeniería de software puede encontrarse en [3] o en el capítulo 4 de [4], la referencia más importante es la guía de Kitchenham [5].
- Shull et al. presentan una breve pero útil introducción a los distintos métodos empíricos utilizados en la investigación en ingeniería de software [6].
- Kitchenham et al. brinda una descripción más detallada de las razones e importancia de las revisiones sistemáticas, así como de sus ventajas y desventajas [5].

3.2. Definición y Etapas

Una revisión sistemática de literatura (o SLR por sus siglas en inglés) es un método para identificar, evaluar e interpretar todas las investigaciones pertinentes a una determinada pregunta de investigación, área temática o fenómeno de interés. Los estudios individuales que contribuyen a una revisión sistemática son llamados estudios primarios; mientras que una revisión sistemática es por tanto un estudio secundario [5].

Una SLR tiene ciertas etapas discretas contempladas en tres fases principales [5]:

1) Planificar la Revisión

- Identificar la necesidad de una revisión.
- Poner en marcha la revisión (opcional).
- Especificar las preguntas de investigación.
- Desarrollar el protocolo de revisión.
- Evaluar el protocolo de revisión (recomendada).

2) Realizar la Revisión

- Identificar la investigación.
- Seleccionar los estudios primarios.
- Evaluar la calidad de los estudios.
- Extraer datos y monitorear.
- Sintetizar los datos.

3) Informar la Revisión

- Especificar los mecanismos de difusión.
- Formatear el informe principal.
- Evaluar el informe (recomendada).

La lista de etapas no es estrictamente secuencial y algunas pueden repetirse más de una vez y pueden involucrar iteración o reimplementación.

3.3. Planificar la Revisión

3.3.1. Identificar la necesidad de una revisión

Antes de comenzar es necesario chequear si no existen revisiones previas o en curso y si una nueva revisión es necesaria. Esta tarea se puede realizar buscando revisiones en los motores de búsqueda relacionados al área temática. Por ejemplo si se quiere realizar una revisión sobre educación de ingeniería de software se puede chequear revisiones existentes utilizando la cadena "*software engineering education systematic literature revision*" en diferentes motores de búsqueda. Si se identifican una o más revisiones que contestan la pregunta de interés, entonces se deben evaluar si cuentan con calidad suficiente.

Una evaluación crítica puede ser realizada utilizando una lista de verificación o checklist. A continuación se incluye una de ejemplo [7]:

- ¿Fue la pregunta de investigación claramente definida en términos de población, intervenciones, comparaciones, resultados y diseño de estudios?
- ¿Fue la estrategia de búsqueda adecuada y apropiada? ¿Hubo alguna restricción de lenguaje, estado o fecha de publicación?
- ¿Se realizaron acciones preventivas para minimizar sesgo y errores en el proceso de selección de estudios?
- ¿Se usó un criterio apropiado para evaluar la calidad de los estudios primarios y se realizaron acciones preventivas para minimizar sesgo y errores en el proceso de evaluación de la calidad?
- ¿Se realizaron acciones preventivas para minimizar sesgo y errores en el proceso de extracción de datos?
- ¿Son adecuados los detalles presentados de cada estudio primario?
- ¿Son apropiados los métodos utilizados para la síntesis de datos? ¿Hay diferencias entre los estudios evaluados? ¿Se agruparon los estudios, y si fue así: era apropiado y significativo hacerlo?
- ¿Reflejan las conclusiones de los autores la evidencia que se revisó?

3.3.2. Puesta en marcha de la revisión

A veces una organización requiere de información sobre un tema específico pero no cuenta con el tiempo o la experiencia para realizar una revisión sistemática propia. En tales casos, podrá encargar a investigadores la revisión sistemática de literatura sobre el tema. En estos casos, la organización deberá presentar un documento de puesta en marcha especificando el trabajo requerido. Un documento de este estilo puede tener carácter contractual y debe explicitar cabalmente los requisitos de la organización.

3.3.3. Especificar las preguntas de investigación

La especificación de las preguntas de investigación es la parte más importante de cualquier revisión sistemática ya que éstas constituyen la base de toda la revisión: definen qué estudios serán incluidos o excluidos de la revisión, definen los datos que se deberían extraer de la literatura revisada y finalmente son respondidas en el reporte final de la revisión utilizando el proceso de síntesis de datos elegido.

De acuerdo a Kitchenham [5], es conveniente crear las preguntas correctas. En general una pregunta correcta es aquella que:

- Es significativa e importante tanto para investigadores como para profesionales.
- Dará lugar a cambios o incrementará la confianza en las prácticas actuales de la ingeniería de software.
- Identificará las discrepancias entre las creencias comúnmente aceptadas y la realidad.

Aspectos a considerar al crear las preguntas de investigación incluyen [5], [4]:

- La *población* en la cual se recolecta la evidencia. Por ejemplo el grupo de programadores que probaron un nuevo método para diseñar software.
- La *intervención* aplicada en el estudio empírico. Por ejemplo, el nuevo método para diseñar software.
- La *comparación* contra la cual se compara la intervención. Por ejemplo, el método antiguo que los programadores utilizaban para diseñar software.
- Los *resultados*. Además de ser estadísticamente satisfactorios deben ser significativos desde el punto de vista práctico. Por ejemplo, seguramente se pueda obviar en los resultados que el nuevo método de diseño requiere el uso de hojas oficio en lugar de A4.
- El *contexto* del estudio. En general es una extensión de la población, puede incluir si fue realizado en la industria o la academia, en qué segmento de la industria o por ejemplo, los incentivos otorgados a los sujetos.
- Los *diseños experimentales* a incluir en las preguntas de investigación. Quizás se requiera restringir a revisar sólo estudios primarios correspondientes a casos de estudio, por ejemplo.

Recomendaciones [8], [9]

- Es probable que sea necesario revisar las preguntas durante el desarrollo del protocolo, a medida que aumente el entendimiento del problema.
- Quizás realizar un estudio de mapeo previo a la revisión sirva de ayuda para determinar el alcance de las preguntas de investigación.
- Es conveniente evaluar que los motores de búsqueda sean capaces de ejecutar las cadenas de búsqueda.

3.3.4. Desarrollar el protocolo de revisión

Un protocolo de revisión especifica los métodos que se utilizarán para llevar a cabo una revisión sistemática específica y su definición de antemano tiene como objetivo minimizar cualquier sesgo en el proceso. Por ejemplo, sin un protocolo, es posible que la selección y/o el análisis de los estudios individuales se vean influenciados por las expectativas del investigador.

Los componentes de un protocolo incluyen todos los elementos de la revisión más alguna información adicional de planificación:

- Antecedentes y justificación de la revisión.
- Preguntas de investigación.
- Estrategia de búsqueda para estudios primarios. Debe incluir los términos de búsqueda y recursos donde se realizará la búsqueda. Los recursos incluyen librerías digitales, revistas específicas, y actas de congresos.
- Criterios de selección de estudios. Usados para determinar cuáles estudios son incluidos o excluidos de la revisión sistemática
- Procedimientos de selección de estudios. El protocolo debe describir cómo se aplicará el criterio de selección, por ejemplo, cuántos asesores evaluarán cada estudio primario, y cómo se resolverán los desacuerdos entre los evaluadores.
- Procedimientos y listas de verificación para evaluar la calidad.
- Estrategia de extracción de datos.
- Síntesis de los datos extraídos.
- Estrategia de difusión.
- Calendario del proyecto de revisión.

En la Tabla 1 se incluye una posible estructura para el protocolo [9].

1. Question Formularization
1.1. Question Focus
1.2. Question Quality and Amplitude
- Problem
- Question
- Keywords and Synonyms
- Intervention
- Control
- Effect
- Outcome Measure
- Population
- Application
- Experimental Design

2. Sources Selection
2.1. Sources Selection Criteria Definition
2.2. Studies Languages
2.3. Sources Identification
- Sources Search Methods
- Search String
- Sources List
2.4. Sources Selection after Evaluation
2.5. References Checking
3. Studies Selection
3.1. Studies Definition
- Studies Inclusion and Exclusion Criteria Definition
- Studies Types Definition
- Procedures for Studies Selection
3.2. Selection Execution
- Initial Studies Selection
- Studies Quality Evaluation
- Selection Review
4. Information Extraction
4.1. Information Inclusion and Exclusion Criteria Definition
4.2. Data Extraction Forms
4.3. Extraction Execution
- Objective Results Extraction
i) Study Identification
ii) Study Methodology
iii) Study Results
iv) Study Problems
- Subjective Results Extraction
i) Information through authors
ii) General Impressions and Abstractions
4.4. Resolution of divergences among reviewers
5. Results Summarization
5.1. Results Statistical Calculus
5.2. Results Presentation in Tables
5.3. Sensitivity Analysis
5.4. Plotting
5.5. Final Comments
- Number of Studies
- Search, Selection and Extraction Bias
- Publication Bias
- Inter-Reviewers Variation
- Results Application
- Recommendations

Tabla 1 – Estructura común de los protocolos de revisiones sistemáticas de literatura.

Recomendaciones [8], [3]

- Conviene que todos los miembros de la revisión sistemática participen activamente en el desarrollo del protocolo de revisión.
- Resulta esencial realizar una prueba piloto del protocolo de investigación. Esto permite encontrar errores en los procedimientos de recolección y agregación de los datos. Es posible que surjan consideraciones que hagan cambiar la metodología que se pretendía usar para contestar las preguntas de investigación.
- Es probable que el protocolo sea modificado a lo largo del proceso de la revisión, por tanto se deben documentar y justificar todos los cambios realizados.

3.3.5. Evaluar el protocolo de revisión

Ya que el protocolo es un elemento crítico de cualquier revisión sistemática, los investigadores deberían acordar un procedimiento para su evaluación.

Se pueden utilizar los mismos criterios utilizados para la identificación de la revisión. Además, se puede chequear la consistencia interna del protocolo para confirmar que: las cadenas de búsqueda fueron derivadas correctamente; los datos a extraer y el procedimiento de análisis de los datos responderán adecuadamente a las preguntas de investigación.

Recomendaciones [8]

- Para la extracción de datos conviene contar con un documento breve que contenga únicamente la definición de los datos y las guías de extracción de datos del protocolo.
- Es necesario que se acuerde un proceso de validación separado de las pruebas piloto del protocolo. Sería ideal que este proceso fuera realizado por revisores externos.

3.4. Realizar la Revisión

3.4.1. Identificar la investigación

En esta etapa se establece y utiliza una estrategia de búsqueda para obtener una lista de todas las publicaciones relevantes para las preguntas de investigación. La estrategia de búsqueda tiene que ser la definida en el protocolo de la revisión y debe estar publicada de forma transparente y replicable.

Es bueno descomponer las preguntas de investigación e identificar cadenas iniciales de búsqueda de acuerdo a población, intervención, comparación, resultados, contexto y diseño del estudio. Además conviene crear listas de sinónimos, abreviaciones y ortografías alternativas.

Otra alternativa, a veces utilizada como complemento, es el uso de la referencias en la búsqueda de nuevos estudios. Se parte de un estudio primario relevante encontrado previamente y se utilizan sus referencias para encontrar nuevos estudios relevantes (*backward snowballing*) o se buscan que estudios lo ubican en sus referencias (*forward snowballing*) [4].

Al crear la estrategia de búsqueda se debe perseguir el equilibrio entre intentar conseguir todos los estudios relevantes y no recuperar demasiados falsos positivos que deban ser excluidos a mano.

Sobre este tema:

- La definición de la estrategia de búsqueda es un tema no menor, sobretodo si se quiere evaluar de alguna forma qué tan buena es. Zhang et al. proponen una técnica sistemática de aproximación a una búsqueda "casi perfecta" utilizando una sub-búsqueda manual e iteraciones [10].

El sesgo en la publicación se refiere a que es más probable que sean publicados estudios con resultados (en algún sentido) positivos a estudios que presentan resultados negativos. Para evitar esto se puede consultar la literatura gris, como por ejemplo reportes técnicos, publicaciones rechazadas o trabajos en progreso.

Recomendaciones [8]

- Existen diferentes estrategias de búsqueda asociadas a diferentes tipos de criterios de búsqueda. Se debe elegir y justificar la estrategia de búsqueda más apropiada para las preguntas de investigación a responder.
- Se deben utilizar diferentes fuentes electrónicas; seguramente no exista una única fuente que pueda proporcionar todos los estudios primarios.
- Actualmente, los motores de búsqueda de ingeniería de software no están diseñados para soportar las revisiones sistemáticas de literatura. Así es que muchas veces las búsquedas quedan diseñadas dependientes de la fuente.
- Las búsquedas deben aplicarse sobre los mismos campos (esto es: título, *abstract*, etc) y utilizando los mismos filtros para todos los artículos en todas las fuentes. De todas maneras, la cadena de búsqueda debe adaptarse para poder ejecutarse en las distintas bibliotecas digitales.

3.4.2. Seleccionar los estudios primarios

Una vez que se han obtenido los estudios primarios potenciales es necesario evaluar cabalmente su relevancia. Para esto se utilizan criterios de inclusión y exclusión, los primeros definen que estudios se deben incluir como relevantes mientras que los últimos se aplican sobre los estudios seleccionados o sobre la lista inicial para remover estudios irrelevantes.

Los criterios de selección de estudios pretenden identificar los estudios primarios que proporcionan evidencia directa acerca de las preguntas de investigación. A fin de reducir la probabilidad de sesgo, los criterios de selección deben ser decididos durante la definición de protocolo, aunque tal vez puedan ser refinados durante el proceso de búsqueda.

Los criterios de inclusión y exclusión deberían estar basados en las preguntas de investigación; y probados para asegurar que son interpretados de forma confiable y que clasifican los estudios correctamente.

Recomendaciones [8]

- En general, los resúmenes estándar en IT e ingeniería de software son demasiado pobres para poder seleccionar estudios primarios. Así que es recomendable revisar también las conclusiones.

3.4.3. Evaluar la calidad de los estudios

El objetivo de esta etapa es analizar y evaluar la calidad de cada estudio seleccionado a fin de determinar su inclusión o no en el proceso de extracción de datos y reporte de resultados de la revisión. En general, el propósito de la evaluación de la calidad es asegurar que los hallazgos de un estudio son relevantes y no sesgados.

Ante la dificultad inicial de contar con un acuerdo en la definición de "calidad" de un estudio Kitchenham [5] sugiere que la calidad se refiere a la medida en que el estudio minimiza el sesgo y maximiza tanto la validez interna y como la externa. A continuación se detallan estos conceptos:

- Sesgo (o error sistemático): Tendencia a producir resultados que se apartan sistemáticamente de los resultados "verdaderos".
- Validez interna: El grado en que el diseño y la realización de un estudio tienden a prevenir el error sistemático.
- Validez externa: El grado en que los efectos observados en el estudio son aplicables fuera del estudio en sí.

En general, se llama "instrumentos de calidad" a las evaluaciones detalladas de calidad; las cuales son, generalmente, listas de verificación de los posibles factores de sesgo que es necesario evaluar en cada estudio. La mayoría de las listas de verificación de calidad incluyen preguntas para evaluar el grado en que los artículos consideran el sesgo y la validez. También, es posible considerar: ítems genéricos relacionados a características del diseño particular del estudio e ítems específicos relacionados al área de la revisión.

A modo de referencia Kitchenham [5] incluye una lista bastante amplia de preguntas para estudios cuantitativos y otra para estudios cualitativos. También indica la alternativa de utilizar una escala de medida para cada ítem en las ocasiones en las cuales una respuesta de Si/No puede resultar engañosa.

Es importante que los investigadores no solo definan en el protocolo el instrumento de calidad sino que también especifiquen cómo será usada la información de calidad. Las opciones son las siguientes:

- Para asistir en la selección de estudios primarios. En este caso, la información de calidad se utiliza para construir criterios de inclusión/exclusión detallados.
- Para asistir en la síntesis y análisis de los datos. En este caso la información de calidad es usada para identificar subconjuntos de los estudios primarios e investigar si las diferencias de calidad están asociadas con diferentes resultados de los estudios primarios.

Es posible contar con ambos tipos de información de calidad en una misma revisión sistemática.

Limitaciones de la Evaluación de la Calidad [5]

A menudo el informe de los estudios primarios es pobre, por lo que quizás no sea posible determinar la forma de evaluar un criterio de calidad. Es tentador suponer que porque algo no se encuentra reportado, no se hizo. Esta suposición puede ser incorrecta. Los investigadores deberían intentar obtener más información por parte de los autores del estudio.

Es posible identificar un análisis estadístico inadecuado o inapropiado, pero sin acceso a los datos originales no es posible corregirlo. Es común que los datos sean confidenciales y que no puedan estar disponibles. En algunos casos, los involucrados pueden negarse a compartir sus datos con otros investigadores debido a que desean continuar publicando sobre el tema.

Recomendaciones [8]

- Existe un énfasis en la necesidad de evaluar la calidad de los estudios primarios. Sin embargo, esto depende del tipo de revisión sistemática de literatura que se lleve a cabo. Por ejemplo, es común que para algunos tipos de revisión ya es un criterio de calidad suficiente que los estudios primarios hayan sido publicados en alguna revista académica.
- Es importante asegurar que la evaluación de calidad sea utilizada en las actividades siguientes de agregación y análisis de datos.

3.4.4. Extraer datos

El objetivo de esta etapa es el diseño de formularios de extracción de datos para que los investigadores puedan registrar adecuadamente la información que obtienen de los estudios primarios. A fin de reducir posibles sesgos, se recomienda definir los formularios de extracción de datos y realizar pruebas piloto durante la definición del protocolo de la revisión.

Los formularios de extracción de datos deben ser diseñados para recolectar toda la información necesaria para contestar las preguntas de la revisión así como los criterios de calidad de los estudios. Si los criterios de calidad van a ser utilizados para identificar criterios de inclusión/exclusión, entonces se requieren formularios separados (ya que la información debe ser recolectada previo a la extracción de datos).

En general un formulario de extracción de datos debería incluir las siguientes partes (secciones) [3]: información de la extracción (responsable y fecha de la extracción, responsable de verificar), información general de los estudios (identificador del estudio, título y detalles de publicación), preguntas para responder las preguntas de la revisión, preguntas para evaluar la calidad de los estudios y resumen de los datos.

Es importante no incluir en la síntesis de una revisión sistemática múltiples publicaciones de los mismos datos ya que informes duplicados podrían afectar seriamente el sesgo de cualquier resultado. Cuando hay publicaciones duplicadas conviene usar la más completa.

Recomendaciones [8]

- Cuando sea necesario revisar un gran número de artículos puede resultar útil que un lector extraiga los datos y otro los valide.
- Si es necesario manipular datos se debe describir y justificar las manipulaciones realizadas.

- En caso de que los datos de un estudio no estén completos se puede escribir a los autores para conseguirlos.

3.4.5. Sintetizar los datos

La síntesis de los datos involucra recolectar y resumir los resultados incluidos en los estudios primarios. En general, hay dos tipos de síntesis de datos: síntesis descriptiva (o narrativa) y síntesis cuantitativa [3].

Con el fin de lograr conclusiones confiables la síntesis debería considerar la fuerza de la evidencia, explorar la consistencia y discutir inconsistencias. El enfoque debería estar definido en el protocolo y se determina por el tipo de las preguntas de investigación, pero también por el tipo de estudios disponibles y por la calidad de los datos.

Sin importar el tipo de síntesis, se debería comenzar con la creación de un resumen de los estudios incluidos. En general se presenta una tabla con detalles importantes como son el tipo, intervenciones, número y característica de los participantes, resultados, etc. También se presentan (en la misma o en otra tabla) elementos de la calidad de los estudios y riesgos de sesgo. Además, este proceso descriptivo debería ser explícitamente riguroso y ayudar a concluir que los estudios son similares y confiables para su síntesis [3].

La información extraída sobre los estudios (por ejemplo: intervención, población, contexto, tamaño de la muestra, resultados, calidad del estudio) debería presentarse tabulada de una manera consistente con las preguntas de la revisión. Las tablas deberían estar estructuradas para resaltar similitudes y diferencias entre los resultados de los estudios.

Es importante identificar si los resultados de los estudios son consistentes unos con otros (o sea, homogéneos) o inconsistentes (o sea, heterogéneos). Los resultados quizás puedan ser tabulados para mostrar el impacto de potenciales fuentes de heterogeneidad (por ejemplo: tipos de estudio, calidad del estudio o tamaño de la muestra).

Kitchenham sugiere realizar un análisis de sensibilidad para determinar si los estudios de baja calidad tienen impacto significativo sobre los resultados de la síntesis. El análisis de sensibilidad también puede realizarse sobre diferentes subconjuntos de los estudios primarios para determinar la robustez de los resultados.

Sobre este tema:

- La síntesis de los datos es un tema con bastantes componentes estadísticos y matemáticos. Inicialmente se pensaba que no iba ser posible realizar síntesis cuantitativas en ingeniería de software (debido a la calidad de los datos) y solo se explicaba en detalle la síntesis narrativa [5]. En [11] plantean una revisión terciaria estudiando los tipos de síntesis realizadas hasta el momento y se percibe cierto incremento en la formalidad al realizar síntesis de datos. El estudio es muy completo y es bastante ilustrativo, sobre todo porque hace una introducción bastante profunda del tema, así como una lista de los tipos de síntesis disponibles y recomendaciones sobre su uso. Para complementar este tema, se puede consultar [12] donde se presenta una visión sobre la síntesis particular de casos de estudios. Un estudio bastante más exhaustivo

sobre síntesis de datos y meta-análisis puede leerse en [7].

3.5. Informar sobre la revisión

La fase final de una revisión sistemática involucra redactar los resultados de la revisión y difundir los resultados a potenciales interesados.

Sobre este tema:

- Escribir artículos científicos, o material publicable en general, es difícil al principio y es necesario completar cierta curva de aprendizaje. Como siempre la práctica hace al maestro. Un libro muy bueno para comenzar es uno publicado por Björn [13], aunque nace en el área de la medicina es simple y sus recomendaciones aplicables en general.

3.5.1. Especificar los mecanismos de difusión

Es importante comunicar los resultados de la revisión sistemática. Por esto es que algunas guías recomiendan planificar la estrategia de difusión durante la etapa de puesta en marcha de la revisión (si existe) o cuando se prepara el protocolo de revisión sistemática.

La mayoría de los académicos asumen que la difusión se trata de publicar en revistas académicas o presentar en conferencias. Sin embargo, si se quiere que los resultados tengan influencia sobre la práctica profesional seguramente sean necesarios otros medios de comunicación, en particular: Revistas con orientación profesional, prensa popular o especializada, folletos breves con resúmenes, posters, páginas webs, entre otros.

3.5.2. Formatear el informe principal

Kitchenham [5] sugiere la siguiente estructura y contenido para informes de revisiones sistemáticas.

Sección	Subsección	Alcance / Comentarios
Título		Debería ser corto pero informativo, así como basado en las preguntas a responder por la revisión. Para revistas académicas, en general se incluye que se trata de una revisión sistemática.
Autoría		
Resumen Ejecutivo o Estructurado	Contexto	La importancia de las preguntas de investigación que aborda la revisión.
	Objetivos	Las preguntas contestadas por la revisión sistemática.
	Métodos	Fuentes de datos, selección de estudios, evaluación de calidad y extracción de datos.
	Resultados	Principales hallazgos incluyendo meta-análisis y análisis de sensibilidad.
	Conclusiones	Implicancias a la práctica y futura investigación.
Antecedentes		Justificación de la necesidad de la revisión. Resumen de revisiones previas. <i>Descripción de la técnica de ingeniería de software que está siendo investigada así como su importancia potencial.</i>
Preguntas de la		Se debe especificar cada una de las preguntas

revisión		de la revisión.
Métodos de la revisión	Fuentes de datos y estrategia de búsqueda	<i>Estos datos se deberían basar en el protocolo original incluyendo información de los cambios que haya sufrido.</i>
	Selección de estudios	
	Evaluación de calidad de los estudios	
	Extracción de datos	
	Síntesis de los datos	
Estudios Incluidos y Excluidos		<p>Criterios de inclusión y exclusión. Lista de los estudios excluidos, así como las razones para la exclusión.</p> <p><i>Los criterios de inclusión y exclusión de los estudios a veces son representados como un diagrama de flujo, ya que los estudios pueden ser excluidos en distintas etapas de la revisión.</i></p>
Resultados	Hallazgos	<p>Descripción de los estudios primarios. Resultados de cualquier resumen cuantitativo. Detalles de cualquier meta-análisis.</p> <p><i>Se debería incluir resúmenes no cuantitativos de los estudios de forma de presentarlos en forma tabular. Los resúmenes cuantitativos se deberían presentar en tablas y gráficos.</i></p>
	Análisis de sensibilidad	
Discusión	Hallazgos principales	<i>Esto debe corresponder con los hallazgos presentados en la sección de resultados.</i>
	Fortalezas y debilidades	<p>Fortalezas y debilidades de la evidencia incluida en la revisión. Relación a otras revisiones, considerando particularmente cualquier diferencia en calidad y resultados.</p> <p><i>Una discusión de la validez de la evidencia considerando el sesgo en la revisión sistemática permite que el lector pueda evaluar la confianza que puede depositar en la evidencia recolectada.</i></p>
	Significado de los hallazgos	<p>Dirección y magnitud de los efectos observados en los estudios resumidos. Aplicabilidad (generalización) de los hallazgos.</p> <p><i>Dejar en claro en qué medida los resultados implican causalidad discutiendo el nivel de evidencia. Discutir todos los beneficios, efectos adversos y riesgos. Discutir variaciones en los efectos y sus razones.</i></p>
Conclusiones	Recomendaciones	<p>Implicaciones prácticas para el desarrollo de software. Preguntas sin responder e implicaciones para investigación futura.</p>
Agradecimientos		Todas las personas que colaboraron en la investigación pero no califican en los criterios de autoría.
Conflictos de intereses		<i>Se debe declarar cualquier interés secundario por parte de los investigadores (por ejemplo: un interés financiero en la tecnología que está siendo evaluada).</i>
Referencias y Anexos		<i>Los anexos pueden ser utilizados para: listar estudios incluidos y excluidos, documentar detalles de la estrategia de búsqueda, y listar datos sin pulir de los estudios incluidos.</i>
Tabla 2 – Índice sugerido para informe de una revisión sistemática de literatura.		

3.5.3. Evaluar el informe

Los informes de la revisión sistemática suelen ser evaluados en el marco de su difusión: un artículo en una revista académica será revisado previo a su publicación, expertos revisarán una tesis de doctorado. Si el protocolo fue revisado por un grupo de expertos previo a la ejecución de la revisión entonces se recomienda que el mismo grupo revise el informe final. El proceso de evaluación puede utilizar las listas de verificación de calidad para revisión sistemáticas similares a las vistas en la sección.

4. Estudios de Mapeo

Los estudios de mapeo (o de alcance) son un tipo de revisión sistemática de literatura, pero a diferencia de las revisiones sistemáticas convencionales su propósito es encontrar y clasificar estudios primarios dentro de un tópico específico [14].

4.1. Diferencias entre un estudio de mapeo y una SLR

La diferencia principal es que una revisión sistemática convencional intenta agregar los estudios primarios en términos de los resultados de la investigación e investiga si son consistentes o contradictorios. En cambio, un estudio de mapeo aspira solamente a clasificar la literatura relevante y a clasificar los estudios con respecto a categorías definidas. A continuación, se presentan las diferencias entre los estudios de mapeo y las revisiones sistemáticas convencionales [15].

Proceso de la SLR	Estudio de Mapeo	SLR
Preguntas de investigación	Generales — relacionadas a las tendencias en investigación. Qué subtópicos se manejan, qué investigadores, cuánta actividad, qué tipos de estudios, etc.	Específicas — relacionadas a los resultados de los estudios empíricos. De la forma: ¿es la tecnología/método A mejor o no que el B?
Proceso de búsqueda	Definido por el área temática.	Definido por las preguntas de investigación.
Estrategia de búsqueda	No muy estricta si sólo son de interés las tendencias en investigación.	Extremadamente estricta — todos los estudios relevantes deben ser encontrados.
Evaluación de calidad	No es esencial.	Es importante para asegurar que los resultados se basan en la evidencia de mejor calidad.
Resultados	Conjunto de artículos relacionados a un área temática y su clasificación según varias categorías.	Respuestas a preguntas de investigación específicas, posiblemente con calificadores (por ejemplo: los resultados aplican únicamente a novatos).

Tabla 3 – Resumen de diferencias entre las revisiones sistemáticas y los estudios de mapeo.

4.2. Esquemas de clasificación

Uno de los principales focos al realizar estudios de mapeo es la creación de un esquema de clasificación. Aunque si bien pueden tenerse ideas iniciales, su construcción finaliza cuando se dispone de los datos de los estudios relevantes encontrados.

En un estudio de mapeo en general se trabaja con uno o más esquemas de clasificación a fin de categorizar los estudios primarios encontrados, por ejemplo, identificando a qué país pertenece o estableciendo de qué tópico trata a partir de una lista de temas. Esto puede implicar una sola dimensión (por ejemplo clasificar según país de publicación) o varias dimensiones (por ejemplo clasificar según país, pero también según tema y fuente de publicación).

Para elegir un esquema de clasificación se puede consultar investigación previa o bibliografía relevante antes de comenzar el estudio de mapeo. En general, es buena práctica validar el esquema revisando un subconjunto de las publicaciones que serán cubiertas mediante el estudio previo a realizarlo.

Si es necesario la construcción de un esquema de clasificación es posible utilizar, por ejemplo, un *keywording* de los *abstracts* [16], esto implica la extracción de los términos clave de los resúmenes de los artículos encontrados. Esto puede ser bueno para algunos casos, pero en general conviene tener una noción básica de aspectos de clasificación y construcción de vocabularios controlados (que es el concepto real asociado a los esquemas de clasificación). Para esto se pueden consultar la guía de Zeng [17].

Al comenzar a trabajar con estudios de mapeo es difícil imaginar que esquemas de clasificación pueden resultar útiles. Algunos de las siguientes ideas pueden ayudar:

- Fechas de publicación — Una buena dimensión a estudiar es la fecha de publicación, en general permite analizar la evolución de los demás dimensiones a través del tiempo. En general se extrae el año solamente, pero para estudios particulares podría utilizarse el mes también.
- País de publicación — Muchas veces es de interés clasificar los estudios primarios por país de publicación.
- Fuente — También puede ser de interés clasificar los artículos por fuente de publicación.
- Métodos de investigación — Un aspecto muy interesante al evaluar artículos es determinar que método de investigación utilizan los autores. Pueden encontrarse distintas manera de realizar esta clasificación, pero está bastante generalizado el uso del esquema planteado por Wieringa [23]. Para cada método de investigación Wieringa presenta criterios de evaluación que pueden ser utilizados, por ejemplo, para crear criterios de evaluación de la calidad. En la Tabla 4 se muestra un resumen de la clasificación de Wieringa.

Categoría	Descripción
Evaluation research	Se investiga la aplicación práctica de una técnica. Se incluye cómo se implementa la técnica y cuáles son sus consecuencias (ventajas y desventajas). En general la técnica no es novedosa pero su aplicación práctica sí.
Proposal of solution	Se propone una solución a un problema y se argumenta su relevancia sin una validación en toda regla. La solución propuesta debe ser novedosa o por lo menos una extensión significativa de una técnica ya existente.
Validation research	Este tipo de artículos investigan las propiedades de una solución propuesta pero que aún no ha sido implementada. Posibles métodos para realizar esta validación pueden ser: experimentos, simulaciones, prototipos, análisis matemático, etc.
Philosophical papers	Explican una manera nueva de ver las cosas, un nuevo marco conceptual, etc.
Opinion papers	Contienen la opinión del autor sobre si algo está mal o bien, o cómo se hacen las cosas, etc.
Personal experience papers	En este tipo de artículos se explican experiencias personales. Su foco es el qué y no el porqué. La evidencia presentada en este tipo de artículos en general es anecdótica.

Tabla 4 – Esquema de clasificación según métodos de investigación [23].

Sobre este tema:

- En [16] brindan una explicación bastante completa del proceso de creación que utilizaron en un caso de ejemplo.
- En [14] se presenta una comparación de varios estudios de mapeo y se comentan características de este tipo de revisiones.
- Otras referencias a los vocabularios controlados son: [18] y [19].

4.3. Presentación de resultados

La presentación de resultados de los estudios de mapeo es un aspecto significativo. Mientras que en las revisiones sistemáticas de literatura los resultados, en general, pueden sintetizarse de alguna forma (cualitativa o cuantitativa), en los estudios de mapeo esto puede ser que no resulte tan fácil de realizar. Por este motivo, es posible que se opte por tablas o visualizaciones gráficas de los resultados.

Como el propósito de un estudio de mapeo está fuertemente ligado a la clasificación de los estudios primarios es frecuente ver gráficos donde se presenta la distribución de las publicaciones para una dimensión (una categoría, se pueden utilizar gráficos de torta u otros) o varias dimensiones (cruzando dos o tres categorías, por ejemplo en un gráfico de burbujas). Se muestran algunos ejemplos a continuación en las Figuras 1, 2 y 3.

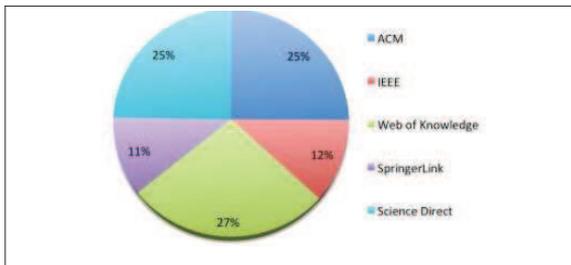


Figura 1 – Porcentaje de publicaciones por fuente bibliográficas [20] — Ejemplo de presentación de cantidad de publicaciones según una dimensión.

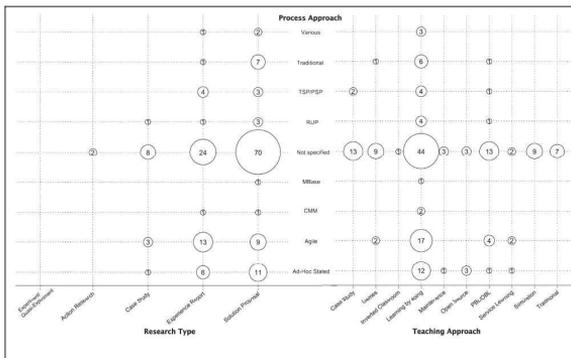


Figura 2 – Tipo de investigación versus Enfoque del proceso versus Enfoque de enseñanza [20] — Ejemplo de gráfico mostrando cantidad de publicaciones según 3 dimensiones (notar que en realidad se cruzan de a dos).

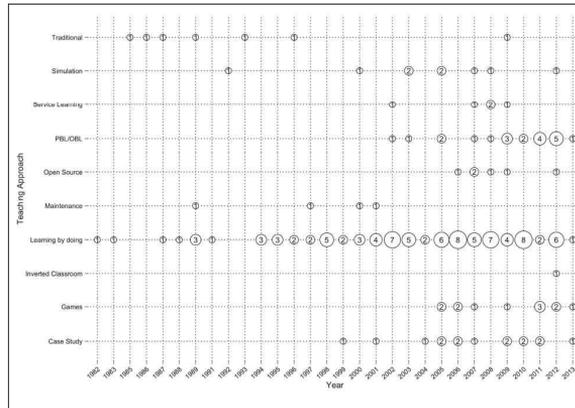


Figura 3 – Evolución de los enfoques de enseñanza a lo largo del tiempo [20] — Ejemplo de presentar la evolución de una dimensión a lo largo del tiempo.

5. Revisiones Terciarias

En un dominio en el cual ya exista un número de revisiones sistemáticas puede ser posible conducir una revisión terciaria (una revisión sistemática de revisiones sistemáticas) para contestar preguntas más amplias. Una revisión terciaria utiliza exactamente la misma metodología que las revisiones sistemáticas de literatura estándares.

Sobre este tema:

- A modo de ejemplo o para entender aspectos avanzados sobre revisiones sistemáticas en ingeniería de software se pueden consultar revisiones terciarias en ese tema en [21], [22], [23] y [24].

6. Anotaciones sobre librerías digitales y motores de búsqueda

El objetivo de esta sección es proveer información relacionada a los motores de búsqueda y librerías digitales que son utilizados para realizar investigaciones basadas en evidencia en ingeniería de software. La intención es que se actualice de forma periódica con nueva información recabada.

A modo de caso de estudio se presenta una búsqueda como ejemplo. El propósito de este caso es obtener *Evidencia empírica sobre la inclusión de los métodos formales en los currículos de informática y computación*. La cadena de búsqueda base se muestra en la Tabla 5. Se incluyen ejemplos de como utilizar las distintas bibliotecas para realizar la búsqueda adaptando cuando sea necesario nuestra cadena de búsqueda.

("formal method") AND ("curriculum" OR "curricula" OR "syllabus" OR "education" OR "course of study" OR "career" OR "degree" OR "training") AND ("software engineering" OR "computer science" OR "Computer Engineering" OR "informatics" OR "information technology" OR "information system")

Tabla 5 – Cadena de búsqueda de ejemplo. Query string para "Evidencia empírica sobre la inclusión de los métodos formales en los currículos de informática y computación".

6.1. ACM Digital

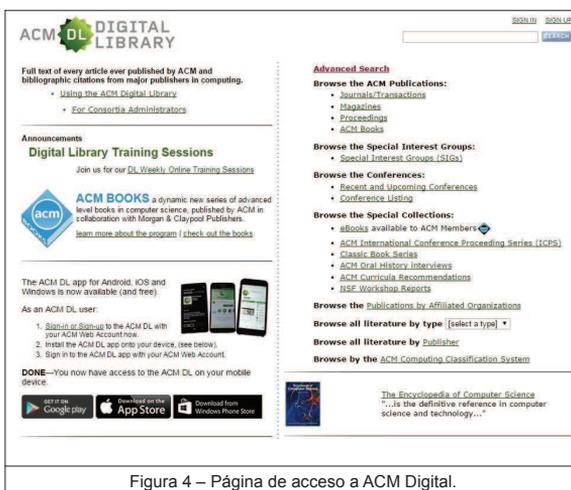


Figura 4 – Página de acceso a ACM Digital.

ACM Digital indexa *journals* y *proceedings* de conferencias publicados por la ACM así como por organizaciones afiliadas. La biblioteca tiene una sección de Advanced search que se accede mediante el link en la esquina superior derecha (Figura 4). En principio es poco intuitivo el uso ya que no presenta un campo para ingresar una *query string* ni una guía de la sintaxis (Figura 5).

ACM DL DIGITAL LIBRARY

Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

SEARCH

Words or Phrases
Find [any field] with []
all of this text (and) []
any of this text (or) []
none of this text (not) []

Names
Find [any field] with []
names []
using all any none of the names

Keywords
Find author's keywords []
using all any none of the keywords

Affiliations
Find company or school []
using all any none of the affiliations

Publication
Find publication []
using all any none of the names

Find publisher []
using all any none of the names

Published since [year] Published before [year]

In publication types Journal Proceeding Transaction Magazine Newsletter

Conference
Find sponsor names []
using all any none of the names

Find location []
using all any none of the locations

Find year (yyyy) []
using all any none of the years

Identification codes
Find ISBN/EISSN []
Find DOI []

Computing Classification System (CCS)
Find code []

Required components
Results must have Full Text Abstract Review

Figura 5 – Primer acceso a la Advanced search.

Se pueden utilizar los campos del formulario de búsqueda y así generar la query que se necesite. Acepta *wildcards*, pero los operadores + y - son reservados. Es posible disponer del cuadro de ingreso de la *query string*, pero para esto se debe realizar una consulta cualquiera y luego acceder al link *Advanced search* desde los resultados de la búsqueda. Esto se muestra en la Figura 6.

ACM DL DIGITAL LIBRARY

Searching for: (software and personal) [start a new search instead](#)
Found 167,530 within The ACM Guide to Compul

Limit your search to Publications from ACM an

REFINE YOUR SEARCH Search Results
Results 1 - 20 of []

Refine by Keywords
[]

Refine by Phrases
Names
Locations
Authors
Editors
Advisors
Discussions

Refine by Publications
Publication Year
Publication Names
ACM Publications
All Publications
Contact Contacts
Publishers

Refine by Conference
Locations
Events
Proceedings Series

ADVANCED SEARCH
[Advanced Search](#)

CONTACT US

Search within Results: 167,530 found
[Start a new search instead](#)

Enter words, phrases or names below to refine your search. Surround phrases or full names with double quotation marks.

SEARCH

Words or Phrases
Find [any field] with []
all of this text (and) []
any of this text (or) []
none of this text (not) []

Names
Find [any field] with []
names []
using all any none of the names

Keywords
Find author's keywords []
using all any none of the keywords

Affiliations
Find company or school []
using all any none of the affiliations

Figura 6 – Acceso al cuadro de ingreso de la query string.

En la Tabla 6 se incluye la cadena de búsqueda del caso de estudio adaptada para esta librería.

```
(Title:"formal method" or Abstract:"formal method") AND ((Title:curriculum or Abstract:curriculum) OR (Title:curricula or Abstract:curricula) OR (Title:syllabus or Abstract:syllabus) OR (Title:education or Abstract:education) OR (Title:"course of study" or Abstract:"course of study") OR (Title:career or Abstract:career) OR (Title:degree or Abstract:degree) OR (Title:training or Abstract:training)) AND ((Title:"software engineering" or Abstract:"software engineering") OR (Title:"computer science" or Abstract:"computer science") OR (Title:"Computer Engineering" or Abstract:"Computer Engineering") OR (Title:"informatics" or Abstract:"informatics") OR (Title:"information technology" or Abstract:"information technology") OR (Title:"information technology" or Abstract:"information technology") OR (Title:"information system" or Abstract:"information system"))
```

Tabla 6 – Cadena de búsqueda adaptada para ACM Digital.

Atención: Si bien Scopus indexa ACM, utilizando ambos buscadores se encontró que algunos artículos que aparecen en ACM no eran devueltos por Scopus.

6.2. IEEEXplore

Esta biblioteca indexa todas las publicaciones de la IEEE. Tiene una sección de *Advanced search* (Figura 7) bastante intuitiva y además cuenta con la opción de *Command search* desde la cual se puede ingresar directamente una *query string* (Figura 8).

Es importante notar que los resultados se muestran por defecto ranqueados por "relevancia", donde la relevancia se calcula teniendo en cuenta frecuencia y proximidad de los términos de búsqueda en el resumen y en el registro de citas, así como el largo del resumen y del registro de citas.

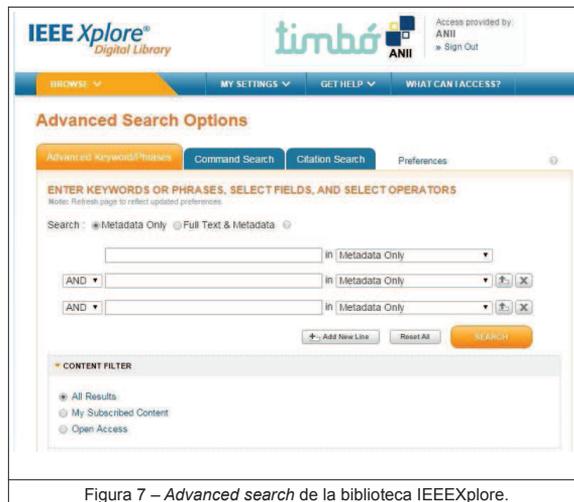


Figura 7 – Advanced search de la biblioteca IEEEExplore.

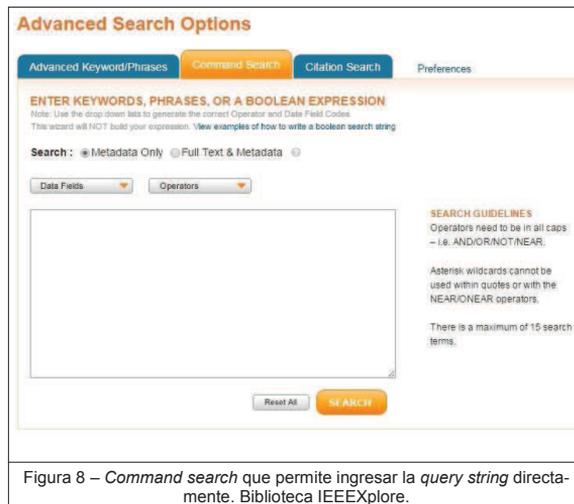


Figura 8 – Command search que permite ingresar la query string directamente. Biblioteca IEEEExplore.

Se puede buscar por metadata, la cual incluye el *abstract*, *indexing terms* y *bibliographic citation information* (*article title*, *author*, *publication title*, etc.). Esta opción es bastante útil por su simplicidad. En la Ta-

bla 7 se muestra la cadena de búsqueda del caso de estudio adaptada utilizando esta opción.

```
((("formal method") AND ("curriculum" OR "curricula" OR "syllabus" OR "education" OR "course of study" OR "career" OR "degree" OR "training") AND ("software engineering" OR "computer science" OR "Computer Engineering" OR "informatics" OR "information technology" OR "information system")))
```

Tabla 7 – Cadena de búsqueda adaptada para IEEEExplore, buscando sobre la metadata.

Para buscar por *abstract* basta con poner el prefijo "Abstract": a cada término (por ejemplo: "Abstract":"formal method"). Lo mismo para *title* con el prefijo "Document Title".

Si la sintaxis de la búsqueda es incorrecta, por ejemplo por omitir un operador, no indica un mensaje de error, sino que muestra que no se encontró ningún resultado. De todas formas hay una guía de búsqueda que advierte que se tienen que utilizar operadores y tiene varios ejemplos. Dicha guía también advierte que se pueden utilizar como máximo 15 términos, pero no dice que tiene un límite de 40 palabras.

Si se ingresa una palabra o frase pero sin comillas, IEEE busca también variantes de esa frase o palabra. Por ejemplo palabras en plural, conjugaciones de verbos. Por ejemplo si la cadena de búsqueda es: "run", también busca, por ejemplo, por "runs", "running", etc. Si la palabra está en inglés británico, también la busca en inglés americano. Por ejemplo si se ingresa la palabra "color", IEEE también busca por "colour".

Si se quiere que una frase o palabra aparezca exactamente en el artículo, se debe poner entre comillas. El carácter " no es reconocido por el buscador (sería lo mismo que no ingresar ninguna comilla), en cambio " si. Por ejemplo, si la cadena de búsqueda es: "formal method" AND "information technology" va a filtrar por artículos con la frase "formal method" y con la palabra "information" o "technology" (o alguna variante), devolviendo más resultados del esperado.

Hay ciertas palabras que son ignoradas por el buscador por ser generales o demasiado frecuentes. Por ejemplo, si la cadena de búsqueda es: "the computer", IEEE Explorer va a buscar por la palabra "computer". Estas palabras son: a, as, have, of, what, about, at, how, on, when, above, can, I, or, where, an, do, in, over, why, and, find, is, show, with, any, for, me, the, you, are, from, not, under, your. Si se quiere que en la búsqueda aparezcan estas palabras, la frase se tiene que poner entre comillas.

El buscador ignora algunos caracteres de puntuación. Por ejemplo si la cadena de búsqueda es "solid-state", IEEE va a buscar por "solid-state", "solid state" y "solid_state", así como cualquier otra ocurrencia de la frase "solid state" con caracteres de puntuación entre las palabras. Si la cadena de búsqueda es: "formal method" va a devolver el resultado del artículo que contiene la frase, por ejemplo, "non-formal method".

Los caracteres especiales, incluyendo caracteres no alfanuméricos, como %, - , @, etc, son ignorados por el buscador. Los únicos caracteres especiales reconocidos por IEEE Xplore son &, +, y /. El carácter * también es especial en el buscador.

En la sección *Advanced Search Options* hay una opción *Preferences* (también se puede acceder desde *My settings-->Preferences*). En la misma se pueden setear las distintas opciones de búsqueda, como por ejemplo la cantidad de resultados por página. La Figura 9 muestra las opciones de preferencias.

Preferences

Search Options

Search History Recording:
 On
 Off

Publisher:
 All
 IEEE
 IET
 AIP
 AIP
 SIAM
 VDE
 TUP
 SIAM
 MTP
 AlkabeLucient
 Morgan & Claypool

Search:
 Metadata Only
 Full Text & Metadata

Display Options for Search Results

Results Layout:
 Title Only
 Title & Citation (Default)
 Title, Citation & Abstract

Results per Page:
 25

Sort By:
 Relevance

Download Options

Bibliographic Citation Format:
 Include:
 Citation Only
 Citation & Abstract

Format:
 Plain Text
 BibTex
 RefWorks
 Endnote, ProCite, RefMan

Email Setting Options

Email Address:
 ivanecaseba2@gmail.com
This will only be used for receiving e-mail alerts from IEEE Xplore. Changing this will not affect the e-mail address associated with your IEEE Account.

Email Format:
 Plain Text
 HTML

Please Note: These preferences will only be applied when signed into IEEE Xplore with your personal username and password. The option to restrict results to a selected publisher does not apply to all search interfaces. Find out more.

Figura 9 – Opciones de las preferencias. Biblioteca IEEEExplore.

Algo que resulta muy útil es la opción *Search history recording*, que si está en *on* guarda las búsquedas que se realizan (por defecto esta en *off*). Todas las búsquedas realizadas se pueden ver en *My Settings-->Search History*.

Atención: Si bien Scopus indexa IEEE, utilizando ambos buscadores se encontró que algunos artículos que aparecen en IEEE no eran devueltos por Scopus.

6.3. ScienceDirect

ScienceDirect es una biblioteca que indexa *journals* publicados por Elsevier y otros publicados por CrossRef (<http://www.crossref.org/>). Esta biblioteca tiene dos opciones avanzadas: *Advanced search* y *Expert search*. La primera (Figura 10) permiten construir de forma asistida una consulta, seleccionando una combinación de AND, OR o AND NOT. Además se pueden aplicar algunos filtros, como años, disciplinas, etc. Permite buscar por *abstract*, *title*, *keywords*, *references*, etc.

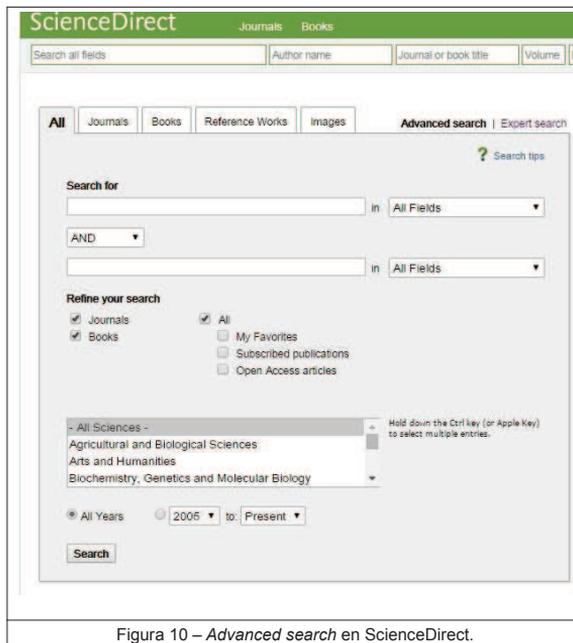


Figura 10 – *Advanced search* en ScienceDirect.

La funcionalidad de *Expert search* permite realizar una búsqueda a través de una *query*. También se pueden aplicar algunos filtros, como años, disciplinas, etc.

No hay directamente una opción para buscar por *abstract*, *title*, *keywords*, *references*, etc. Si se quiere buscar específicamente por algún campo, hay que adaptar la cadena de búsqueda. Una opción es generar una búsqueda utilizando la opción *Advanced search* (la cual te muestra la cadena de búsqueda correspondiente) para luego poder escribir la cadena de búsqueda en *Expert search* de forma adecuada. En la Tabla 8 se muestra la cadena de búsqueda del caso de estudio adaptada a esta plataforma.

```
TITLE-ABSTR-KEY("formal method") AND (TITLE-ABSTR-KEY("curriculum") OR TITLE-ABSTR-KEY("curricula") OR TITLE-ABSTR-KEY("syllabus") OR TITLE-ABSTR-KEY("education") OR TITLE-ABSTR-KEY("course of study") OR TITLE-ABSTR-KEY("career") OR TITLE-ABSTR-KEY("degree") OR TITLE-ABSTR-KEY("training")) AND (TITLE-ABSTR-KEY("software engineering") OR TITLE-ABSTR-KEY("computer science") OR TITLE-ABSTR-KEY("Computer Engineering") OR TITLE-ABSTR-KEY("informatics") OR TITLE-ABSTR-KEY("information technology") OR TITLE-ABSTR-KEY("information system"))
```

Tabla 8 – Cadena de búsqueda adaptada para ScienceDirect

Existen una opción *Search history* que por defecto está en *Turn on* y una opción *Search tips*.

Atención: Si bien Scopus indexa ScienceDirect, utilizando ambos buscadores se encontró que algunos artículos que aparecen en ScienceDirect no eran devueltos por Scopus.

6.4. Scopus

Es un motor de búsqueda que indexa diferentes *journals* y bibliotecas, la lista completa puede accederse desde <http://www.elsevier.com/solutions/scopus/content>. Tiene una sección para *Advanced search* (Figura 11).

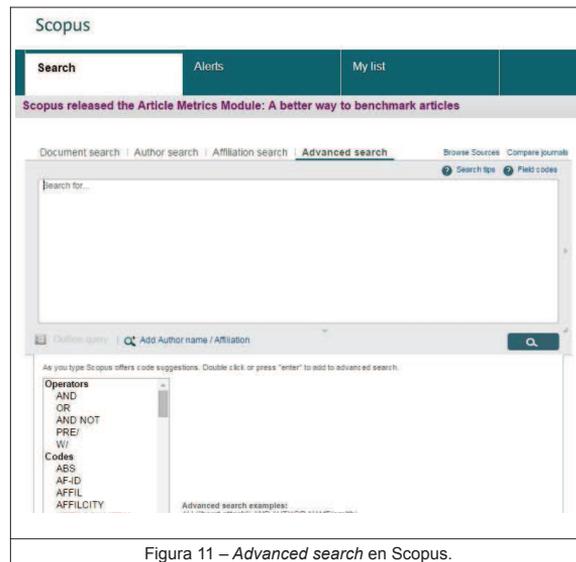


Figura 11 – *Advanced search* en Scopus.

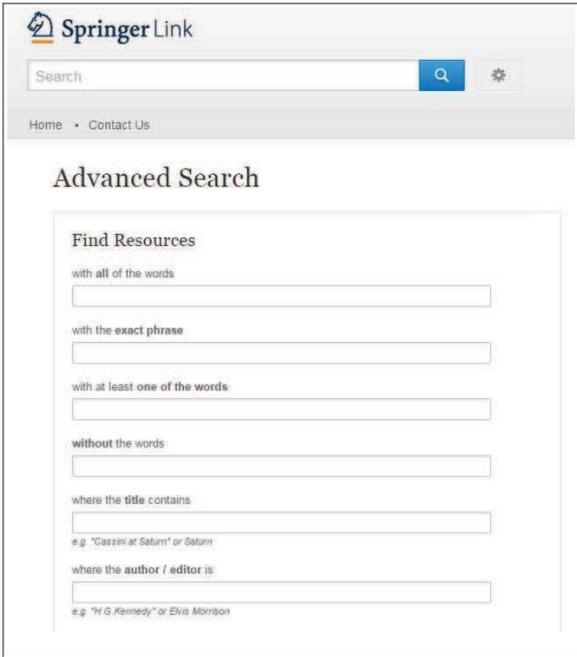
Es posible utilizar diferentes filtros para obtener resultados más acertados, por ejemplo: períodos de años, áreas temáticas, etc.

```
("formal method") AND ("curriculum" OR "curricula" OR "syllabus" OR "education" OR "course of study" OR "career" OR "degree" OR "training") AND ("software engineering" OR "computer science" OR "Computer Engineering" OR "informatics" OR "information technology" OR "information system")
```

Tabla 9 – Cadena de búsqueda adaptada para Scopus

6.5. SpringerLink

Esta biblioteca digital tiene una sección para *Advanced search* (Figura 11). La biblioteca busca en texto completo por lo que la cadena de búsqueda original retorna demasiados resultados. Se puede refinar específicamente la cadena para este buscador, evitando excesivas combinaciones de palabras, pero a veces es inviable. Puede ser que alcance con buscar en Scopus dependiendo de las publicaciones en las que se esté buscando.



Springer Link

Search

Home • Contact Us

Advanced Search

Find Resources

with all of the words

with the exact phrase

with at least one of the words

without the words

where the title contains

e.g. "Cassini at Saturn" or Saturn

where the author / editor is

e.g. "H. G. Kennedy" or Elvia Morrison

Figura 12 – *Advanced search* en SpringerLink.

7. Referencias

- [1] B. A. Kitchenham, T. Dyba, y M. Jorgensen, «Evidence-based software engineering», en *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, 2004, pp. 273-281.
- [2] T. Dyba y T. Dingsøyr, «Strength of Evidence in Systematic Reviews in Software Engineering», en *Proceedings of the Second ACM-IEEE*

- International Symposium on Empirical Software Engineering and Measurement*, New York, NY, USA, 2008, pp. 178–187.
- [3] Z. Stapić, E. G. López, A. G. Cabot, L. de Marcos Ortega, y V. Strahonja, «Performing systematic literature review in software engineering», *Proc. 23rd Cent. Eur. Conf. Inf. Intell. Syst. Varažd.*, pp. 441–447, 2012.
 - [4] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, y A. Wesslin, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.
 - [5] B. Kitchenham y S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.
 - [6] S. Easterbrook, J. Singer, M.-A. Storey, y D. Damian, «Selecting Empirical Methods for Software Engineering Research», en *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, y D. K. Sjøberg, Eds. Springer London, 2008, pp. 285-311.
 - [7] J. Akers, R. Aguiar-Ibáñez, A. Baba-Akbari Sari, S. Beynon, A. Booth, J. Burch, D. Chambers, D. Craig, J. Dalton, S. Duffy, A. Eastwood, D. Fayter, T. Fonseca, D. Fox, J. Glanville, S. Golder, S. Hempel, K. Light, C. McDaid, G. Norman, B. Phillips, C. Pierce, S. Rice, A. Rithalia, M. Rodgers, F. Sharp, A. Sowden, L. Stewart, C. Stock, R. Trowman, R. Wade, M. Westwood, P. Wilson, N. Woolacott, G. Worthy, y K. Wright, *Systematic Reviews: CRD's Guidance for Undertaking Reviews in Health Care*, vol. 3rd. York, UK: Centre for Reviews and Dissemination, University of York, 2009.
 - [8] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, y M. Khalil, «Lessons from applying the systematic literature review process within the software engineering domain», *J. Syst. Softw.*, vol. 80, n.º 4, pp. 571-583, abr. 2007.
 - [9] J. Biolchini, P. G. Mian, y A. C. C. Natali, «Systematic Review in Software Engineering», COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, RT-ES 679/05, may 2005.
 - [10] H. Zhang, M. A. Babar, y P. Tell, «Identifying relevant studies in software engineering», *Inf. Softw. Technol.*, vol. 53, n.º 6, pp. 625 - 637, 2011.
 - [11] D. S. Cruzes y T. Dyba, «Research synthesis in software engineering: A tertiary study», *Inf. Softw. Technol.*, vol. 53, n.º 5, pp. 440 - 455, 2011.
 - [12] D. S. Cruzes, T. Dyba, P. Runeson, y M. Host, «Case Studies Synthesis: Brief Experience and Challenges for the Future», en *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*, 2011, pp. 343-346.
 - [13] B. Gustavii, *How to Write and Illustrate a Scientific Paper*, Second. Cambridge University Press, 2008.
 - [14] D. Budgen, M. Turner, P. Brereton, y B. Kitchenham, «Using Mapping Studies in Software Engineering», en *PPIG 2008: In 20th Annual Meeting of the Psychology of Programming Interest Group*, 2008, pp. 195–204.
 - [15] B. A. Kitchenham, D. Budgen, y O. P. Brereton, «The Value of Mapping Studies: A Participant-observer Case Study», en *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, Swinton, UK, UK, 2010, pp. 25–33.
 - [16] K. Petersen, R. Feldt, S. Mujtaba, y M. Mattsson, «Systematic Mapping Studies in Software Engineering», en *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Swinton, UK, UK, 2008, pp. 68–77.
 - [17] M. L. Zeng, *Constructing Controlled Vocabularies: A primer*. 2005.

- [18] M. R. Marques, A. Quispe, y S. F. Ochoa, «A systematic mapping study on practical approaches to teaching software engineering», en *Frontiers in Education Conference (FIE), 2014 IEEE*, 2014, pp. 1-8.
- [19] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, y S. Linkman, «Systematic Literature Reviews in Software Engineering - A Systematic Literature Review», *Inf Softw Technol*, vol. 51, n.º 1, pp. 7–15, ene. 2009.
- [20] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, y S. Linkman, «Systematic literature reviews in software engineering – A tertiary study», *Inf. Softw. Technol.*, vol. 52, n.º 8, pp. 792 - 805, 2010.
- [21] F. Q. B. da Silva, A. L. M. Santos, S. Soares, A. C. C. França, C. V. F. Monteiro, y F. F. Maciel, «Six years of systematic literature reviews in software engineering: An updated tertiary study», *Inf. Softw. Technol.*, vol. 53, n.º 9, pp. 899 - 913, 2011.
- [22] H. Zhang y M. A. Babar, «Systematic reviews in software engineering: An empirical investigation», *Inf. Softw. Technol.*, vol. 55, n.º 7, pp. 1341 - 1354, 2013.
- [23] R. Wieringa, N. Maiden, N. Mead, y C. Rolland, «Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion», *Requir. Eng.*, vol. 11, n.º 1, pp. 102–107, dic. 2005.

8. Material complementario

En esta sección se incluyen 3 anexos seleccionados del sitio de ingeniería de software basado en evidencias – *Evidence-Based Software Engineering* (<http://community.dur.ac.uk/ebse/>).

Estos corresponden a:

- Una plantilla para revisiones sistemáticas de literatura — <http://community.dur.ac.uk/ebse/resources/templates/SLRTemplate.pdf>
- Una plantilla para estudios de mapeo — <http://community.dur.ac.uk/ebse/resources/templates/MappingStudyTemplate.pdf>
- Un ejemplo de una revisión sistemática de literatura — <http://community.dur.ac.uk/ebse/resources/studies/protocol/Agile-Methods-literature-review-protocol.pdf>

Los tres documentos no han sido modificados y se reproducen bajo la licencia de Creative Commons (<http://creativecommons.org/licenses/by-sa/2.0/uk/>).

Template for a Systematic Literature Review Protocol

1. Change Record

This should be a list or table summarizing the main updates and changes embodied in each version of the protocol and (where appropriate), the reasons for these.

2. Background

- a) explain why there is a need for a study on this topic
- b) specify the main research question being addressed by this study
- c) specify any additional research questions that will be addressed
- d) if extending previous research on the topic, explain why a new study is needed

3. Search Strategy

- a) specify and justify basic strategy: manual search, automated search, or mixed
- b) for automated searches, specify search terms and compounds of these (and record results of any prototyping of the search strings)
- c) for automated searches, identify resources to be used (digital libraries and search engines)
- d) for manual searches, identify the journals and conferences to be searched
- e) specify the time period to be covered by the review and any reasons for your choice
- f) identify any ancillary search procedures, e.g. asking leading researchers or research groups, or accessing their web sites; or checking reference lists of primary studies
- g) specify how the search process is to be evaluated (e.g. against a known subset of papers; or against the results from a previous systematic review)

4. Selection Criteria

- a) identify the *inclusion* criteria for primary studies
- b) identify the *exclusion* criteria
- c) define how selection will be undertaken (roles of analysts)
- d) define how agreement among analysts will be evaluated
- e) define how any differences between analysts will be resolved

5. Study Quality Assessment

- a) specify the quality checklists to be used
- b) specify how the checklist will be evaluated
- c) define how agreement among data extractors will be evaluated
- d) define how any differences between data extractors will be resolved
- e) identify the procedures to use for applying the checklists (e.g. details inclusion/exclusion; partitioning the primary studies during aggregation or meta-analysis; explaining the results of primary studies)

6. Data Extraction

- a) design data extraction form (and check via a dry run)
- b) specify the strategy for extracting the data and the form (paper, on-line etc.)
- c) identify how the data extraction process is to be undertaken and validated, particularly any data that require numerical calculations, or are subjective

7. Synthesis

- a) specify the form of analysis to be used (e.g. narrative, tabulation, meta-analysis)
- b) assess the threats to validity (construct, internal, external), particularly constraints on the search process and deviations from standard practice

8. Study Limitations

Specify residual validity issues including potential conflicts of interest (i.e. that are inherent in the context of the study, rather than arising from the plan).

9. Reporting

Identify target audience, relationship to other studies, planned publications, authors of the publications.

10. Schedule

Provide time estimates for all of the major steps.

Template for a Mapping Study Protocol

1. Change Record

This should be a list or table summarizing the main updates and changes embodied in each version of the protocol and (where appropriate), the reasons for these.

2. Background

- a) explain why there is a need for a study on this topic
- b) identify the topic that is to be 'scoped' in the study
- c) specify any research questions that will be addressed
- d) if extending previous research on the topic, explain why a new study is needed

3. Search Strategy

- a) specify and justify basic strategy: manual search, automated search, or mixed
- b) for automated searches, specify search terms and compounds of these (and record results of any prototyping of the search strings)
- c) for automated searches, identify resources to be used (digital libraries and search engines)
- d) for manual searches, identify the journals and conferences to be searched
- e) specify the time period to be covered by the review and any reasons for your choice
- f) identify any ancillary search procedures, e.g. asking leading researchers or research groups, or accessing their web sites; or checking reference lists of primary studies
- g) specify how the search process is to be evaluated (e.g. against a known subset of papers; or against the results from a previous systematic review or mapping study)

4. Selection Criteria

- a) identify the *inclusion* criteria for primary studies
- b) identify the *exclusion* criteria
- c) define how selection will be undertaken (roles of analysts)
- d) define how agreement among analysts will be evaluated
- e) define how any differences between analysts will be resolved

5. Data Extraction

- a) design data extraction form (and check via a dry run)
- b) specify the strategy for extracting the data and the form (paper, on-line etc.)
- c) identify how the data extraction process is to be undertaken and validated, particularly any data that require numerical calculations, or are subjective

6. Synthesis

- a) specify the categorization schemes to be used
- b) assess the threats to validity (construct, internal, external), particularly constraints on the search process and deviations from standard practice

7. Study Limitations

Specify residual validity issues including potential conflicts of interest (i.e. that are inherent in the context of the study, rather than arising from the plan).

8. Reporting

Identify target audience, relationship to other studies, planned publications, authors of the publications.

9. Schedule

Provide time estimates for all of the major steps.

Review Protocol – Agile Software Development

Tore Dybå

1. Background

The concept of “Agile Software Development” has sparked a lot of interest in both industry and academia. Advocates of agile methods consider them the best thing that has happened to SE in recent years, while opponents view them as a backlash comparing them to hacking.

There is therefore a need to undertake a systematic review of the available empirical research relevant to agile software development to present a fair evaluation of the benefits and limitations by using a trustworthy, rigorous, and auditable methodology.

Thus, the purpose of this protocol is to specify the plan which the review will follow to identify, appraise and collate the evidence on the benefits and limitations of agile software development.

2. Objectives

The objectives of the review are to:

- Undertake a systematic review of empirical research on the benefits and limitations of agile software development.
- Select a sub-set of studies to review in-depth.
- Synthesize the evidence from these studies about the benefits and limitation of agile software development.
- Identify the research methods used.
- Identify any gaps in current research in order to suggest areas for further investigation.
- Provide a framework/background in order to appropriately position new research activities.

More specifically, we want to find out:

1. What is currently known about the benefits and limitations of agile software development?
2. What is the strength of the evidence in support of the claims regarding agile software development?
3. What are the implications of these studies for the software industry and the research community?

In addition to producing substantive findings regarding agile software development, the review also aims to advance methodology for integrating diverse study types, including ‘qualitative’ research, within systematic reviews of software engineering interventions.

3. Criteria for considering studies for this review

Types of studies:

Articles to be included in the review must present empirical data and pass the minimum quality threshold described below.

Types of participants:

Studies of both students and professional software developers will be included.

Types of intervention:

Inclusion of studies will not be restricted to any specific type of intervention.

Types of outcome measures:

Studies must include one of the primary outcomes of time-to-market, cost, or process or product quality to be included.

4. Search strategy for identification of studies

The search strategies will include electronic databases and hand searches of conference proceedings.

Electronic databases:

The following electronic databases will be searched: ISI Web of Science, IEEE Xplore, ACM Digital Library, Kluwer Online, ScienceDirect – Elsevier, Springer, Wiley Inter Science Journal Finder, and Compendex.

Conference proceedings:

The following conference proceedings will be hand searched for research papers: XP, XP/Agile Universe, and Agile Development Conference.

Search strategy:

The title, abstract, and keywords of the articles in the included electronic databases and conference proceedings will be searched according to the following search strategy:

1. agile AND software
2. extreme programming
3. xp AND software
4. scrum AND software
5. crystal AND software AND (clear OR orange OR red OR blue)
6. dsdm AND software
7. fdd AND software
8. feature AND driven AND development AND software
9. lean AND software AND development

5. Methods of the review

Selection of studies:

The search strategy will identify all relevant articles and these will be reviewed by two reviewers. Only studies written in the English language will be included.

Studies will be excluded if:

- (i) the study's focus, or main focus, is NOT agile software development;
- (ii) the study does NOT present empirical data.

If it is unclear from the title, abstract, and keywords whether a study conforms to these screening criteria, it will be included for a detailed quality assessment (see below).

Assessment of methodological quality:

Each study will be assessed independently by two reviewers according to 10 criteria (see Appendix 1). These criteria are informed by those proposed for the Critical Appraisal Skills Programme (CASP), in particular those for assessing the quality of 'qualitative' research (Greenhalgh, 2001), and by principles of good practice for conducting empirical research in software engineering (Kitchenham *et al.*, 2002).

The 10 criteria cover three main quality issues that need to be considered when appraising the studies of this research:

- *Rigour*: has a thorough and appropriate approach been applied to key research methods in the study?
- *Credibility*: are the findings well presented and meaningful?
- *Relevance*: how useful are the findings to the software industry and the research community?

Two screening criteria relate to the quality of the *reporting* of a study's rationale, aims, and context. Each study will be assessed according to whether:

- (i) the aims and objectives are clearly reported (including a rationale for why the study was undertaken);
- (ii) there is an adequate description of the context in which the research was carried out.

A further five criteria relate to the *rigour* of the research methods employed to establish the validity of data collection tools and the analysis methods, and hence the trustworthiness of the findings. Each study will be assessed according to whether:

- (iii) the research design was appropriate to address the aims of the research
- (iv) there is an adequate description of the sample used and the methods for how the sample was identified and recruited;
- (v) any control groups were used to compare treatments
- (vi) appropriate data collection methods were used and described;
- (vii) there is adequate description of the methods used to analyse data and whether appropriate methods for ensuring the data analysis was grounded in the data.

In addition, two criteria relate to the assessment of the *credibility* of the study methods for ensuring that the findings are valid and meaningful. In relation to this, the reviewers will judge studies according to whether:

- (viii) the relationship between the researcher and participants has been adequately considered;
- (ix) the study provide clearly stated findings with credible results and justified conclusions.

The final criterion relates to the assessment of the *relevance* of the study for the software industry at large and the research community. Thus, the reviewers will judge studies according to whether:

- (x) they provide value for research or practice.

Taken together, these 10 criteria provide a measure of the extent to which we can be confident that a particular study's findings can make a valuable contribution to this review.

Two researchers will carry out all the procedures in this section independently, and then meet to compare their assessments and establishing consensus. Any differences will be resolved by consulting a third reviewer.

Data management and extraction:

All data will be extracted by two reviewers. Differences in data extraction will be resolved by referring back to the original article and establishing consensus. A third reviewer will be consulted to help resolve differences.

When necessary the authors of the primary studies will be contacted to obtain additional information

The following data will be extracted regarding the *description* of the studies:

- (i) bibliographic reference
- (ii) type of article (e.g. journal, conference, workshop)
- (iii) research question
- (iv) study characteristics/phenomenon
- (v) study context

The following data will be extracted regarding the *findings* of the studies:

- (i) study design
- (ii) sampling
- (iii) data collection
- (iv) data analysis
- (v) findings and conclusions
- (vi) validity
- (vii) relevance

The coding forms for data extraction are included in Appendix 2.

Data synthesis:

The qualitative findings and conclusions of each study will be copied verbatim as reported by study authors into NVivo from QSR Software, a specialist software package for undertaking qualitative analysis of textual data. Examining these findings, every sentence or paragraph within the report of findings will be assigned a code to describe it. Two independent reviewers will look for similarities and differences between the codes in order to group them into a hierarchical tree structure of descriptive themes. A narrative summary of the findings across the studies will then be written and organized by the descriptive themes.

References

Critical Appraisal Skills Programme (CASP), www.phru.nhs.uk/casp/casp.htm

GREENHALGH, T. (2001) *How to Read a Paper*, 2nd ed., BMJ Books.

KITCHENHAM, B.A. (2004) *Procedures for Performing Systematic Reviews*, Keele University, Technical Report TR/SE-0401-ISSN:1353-7776 and NICTA Technical Report 0400011T.1.

KITCHENHAM, B.A., PFLEEGER, S.L., PICKARD, L.M., JONES, P.W., HOAGLIN, D.C., EL EMAM, K., AND ROSENBERG, J. (2002) Preliminary Guidelines for Empirical Research in Software Engineering, *IEEE Transactions on Software Engineering*, 28(8):721-734.

Appendix 1: Quality assessment form

Screening questions

1. Is this a research paper? <i>Consider:</i> – Is the paper based on research (or is it merely a “lessons learned” report based on expert opinion)	<input type="checkbox"/> Yes <input type="checkbox"/> No
2. Is there a clear statement of the aims of the research? <i>Consider:</i> – Is there a rationale for why the study was undertaken? – Is the study’s focus or main focus on Agile Software Development – Does the study present empirical data? – Is there a clear statement of the study’s primary outcome (i.e. time-to-market, cost, or product or process quality)?	<input type="checkbox"/> Yes <input type="checkbox"/> No
3. Is there an adequate description of the context in which the research was carried out <i>Consider if the researcher has identified:</i> – The industry in which products are used (e.g. banking, telecommunications, consumer goods, travel, etc)? – The nature of the software development organization (e.g. in-house department or independent software supplier)? – The skills and experience of software staff (e.g. with a language, a method, a tool, an application domain)? – The type of software products used (e.g. a design tool, a compiler) – The software processes being used (e.g. a company standard process, the quality assurance procedures, the configuration management process)?	<input type="checkbox"/> Yes <input type="checkbox"/> No

If questions 1 or 2 receive a “No” response do not continue with the quality assessment.

Detailed questions

<p>Research design</p> <p>4. Was the research design appropriate to address the aims of the research? <i>Consider:</i> – Has the researcher justified the research design (e.g. have they discussed how they decided which methods to use)?</p>	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Sampling</p> <p>5. Was the recruitment strategy appropriate to the aims of the research? <i>Consider:</i> – Has the researcher explained how the participants or cases were identified and selected? – Are the cases defined and described precisely? – Was the cases representative of a defined population? – Have the researchers explained why the participants or cases they selected were the most appropriate to provide access to the type of knowledge sought by the study? – Was the sample size sufficiently large?</p>	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Control group</p> <p>6. Was there a control group with which to compare treatments? <i>Consider:</i> – How were the controls selected? – Were they representative of a defined population? – Was there something special about the controls? – Was the non-response high? Could non-respondents be different in any way?</p>	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Data collection</p> <p>7. Was the data collected in a way that addressed the research issue? <i>Consider:</i> – Were all measures clearly defined (e.g. unit and counting rules)? – Is it clear how data was collected (e.g. semi-structured interviews, focus group etc.)? – Has the researcher justified the methods that were chosen? – Has the researcher made the methods explicit (e.g. is there an indication of how interviews were conducted, did they use an interview guide)? – If the methods were modified during the study has the researcher explained how and why? – If the form of data is clear (e.g. tape recording, video material, notes etc.)? – If quality control methods were used to ensure completeness and accuracy of data collection?</p>	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>

<p>Data analysis</p> <p>8. Was the data analysis sufficiently rigorous? <i>Consider:</i></p> <ul style="list-style-type: none"> - Was there an in-depth description of the analysis process? - If thematic analysis was used is it clear how the categories/ themes were derived from the data? - Has sufficient data been presented to support the findings? - To what extent has contradictory data been taken into account? - If quality control methods were used to verify the results? 	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Reflexivity (research partnership relations/recognition of researcher bias)</p> <p>9. Has the relationship between researcher and participants been adequately considered? <i>Consider:</i></p> <ul style="list-style-type: none"> - Has the researcher critically examined their own role, potential bias and influence during formulation of research questions, sample recruitment, data collection, and analysis and selection of data for presentation? - How the researcher responded to events during the study and whether they considered the implications of any changes in the research design? 	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Findings</p> <p>10. Is there a clear statement of findings? <i>Consider:</i></p> <ul style="list-style-type: none"> - Are the findings explicit (e.g. magnitude of effect)? - Has an adequate discussion of the evidence, both for and against the researcher's arguments, been demonstrated? - Has the researcher discussed the credibility of their findings (e.g. triangulation, respondent validation, more than one analyst)? - Are limitations of the study explicitly discussed? - Are the findings discussed in relation to the original research questions? - Are the conclusions justified from the results? 	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>
<p>Value of the research</p> <p>11. Is the study of value for research or practice? <i>Consider:</i></p> <ul style="list-style-type: none"> - Does the researcher discuss the contribution the study makes to existing knowledge or understanding (e.g. do they consider the findings in relation to current practice or relevant research-based literature)? - Does the research identify new areas where research is necessary? - Does the researcher discuss whether or how the findings can be transferred to other populations or considered other ways the research can be used? 	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>

Appendix 2: Data extraction form

<p>Study description</p> <ol style="list-style-type: none"> 1. Study identifier: 2. Date of data extraction 3. Bibliographic reference: 4. Type of article: 5. Study aims: 6. Objectives: 7. Design of study: 8. Research hypothesis: 9. Definition of agile software development given in study: 10. Sample description 11. Setting of study 12. Control group 13. Data collection 14. Data analysis 	<p>Unique id for the study</p> <p>Author, year, title, source</p> <p>Journal article, conference paper, workshop paper, book section</p> <p>What were the aims of the study?</p> <p>What were the objectives?</p> <p>Qualitative, quantitative (experiment, survey, case study, action research)</p> <p>Statement of hypotheses, if any</p> <p>Verbatim from the study</p> <p>Size, students, professionals (age, education, experience)</p> <p>Industry, in-house/supplier, products and processes used</p> <p>Yes, no (#groups, sample size)</p> <p>How was the data obtained? (questionnaires, interviews, forms)</p> <p>How was the data analyzed? (qualitative, quantitative)</p>
<p>Study findings</p> <ol style="list-style-type: none"> 1. Findings and conclusions 2. Validity 3. Relevance 	<p>What were the findings and conclusions? (verbatim from the study)</p> <p>Limitations, threats to validity</p> <p>Research, practice</p>

