

Comparativa de Selección de Estudios Primarios en una Revisión Sistemática

John W. Castro¹ and Silvia T. Acuña¹

¹ Departamento de Ingeniería Informática, Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente 11, 28049 Madrid, España
john.castro@estudiante.uam.es, silvia.acunna@uam.es

Resumen. Este artículo presenta una comparativa de dos selecciones de estudios primarios durante una revisión sistemática para una investigación exploratoria sobre el proceso de desarrollo de software *open source*. Ambas revisiones utilizan el procedimiento de Kitchenham. Pero, uno selecciona los estudios primarios mediante un procedimiento exhaustivo realizado por un investigador novato y el otro utiliza un procedimiento contextual realizado por un investigador experto. Los criterios de comparación son tiempo consumido en horas y conjunto de artículos relevantes. El porcentaje de artículos relevantes que no encontró el experto es del 20% respecto a los encontrados por el novato. Sin embargo, el tiempo invertido por el experto es una décima parte del tiempo consumido por el novato. Estos resultados sugieren que la experiencia del experto influye en la revisión y proponemos involucrar al investigador experto en el desarrollo del protocolo de revisión, selección y evaluación de calidad de estudios primarios.

Keywords: Revisión Sistemática, Selección de Estudios Primarios, Ingeniería del Software, Investigación Exploratoria, Proceso de Desarrollo de Software Open Source.

1 Introducción

Las revisiones sistemáticas (en inglés *systematic reviews*, SR) han sido recientemente introducidas en el área de la ingeniería del software [13], [14] como una aproximación sistemática y estructurada para realizar revisiones de literatura [15]. En el año 2004, Bárbara Kitchenham propuso pautas adaptadas a los problemas específicos de investigación en el área de la ingeniería del software. Estas pautas han sido seguidas, evaluadas [6], [16], [24] y actualizadas en consecuencia en el 2007 [15]. Desde entonces, las SR han cobrado una importancia significativa en la ingeniería del software como un medio para identificar, evaluar e interpretar todos los datos disponibles correspondientes a una pregunta de investigación en particular, área temática o fenómeno de interés [15].

Siguiendo este método, se tienen más probabilidades de que el conjunto de trabajos seleccionado sea completo, correcto y relevante en relación con las hipótesis de trabajo formuladas. Durante el procedimiento que se sigue en una SR se aplican

diferentes criterios de cribas sistematizadas pero no estandarizadas, que dependen de la experiencia del investigador y de la contextualización de la información obtenida (cadenas de búsqueda, fuentes con factor de impacto, limitaciones de la búsqueda, precisión de las hipótesis de investigación, etc.). Una SR puede llevarse a cabo tanto por investigadores experimentados como por novatos; sin embargo, mientras que las experiencias de los investigadores expertos en la realización de SR son importantes para mejorar el cuerpo de conocimientos, la experiencia de los investigadores noveles son igualmente importantes para determinar los diferentes problemas a los que se enfrentan en la realización de una SR. Con un conocimiento previo de estos problemas, los investigadores noveles puedan planificar mejor sus SR y buscar la guía de investigadores expertos [22].

Mediante una revisión sistemática, recogimos y comparamos la evidencia existente sobre las actividades que forman parte del proceso de desarrollo de software *open source*. No existe un modelo de proceso de desarrollo de software *open source* mundialmente aceptado [23]. El objetivo de la investigación para la cual se realiza la revisión sistemática es definir un proceso de desarrollo de software *open source*.

Aunque las revisiones sistemáticas se deben realizar en equipo, en este trabajo de investigación queremos comparar el desempeño de un investigador novato con un investigador experto en la selección de estudios primarios. Además, describimos nuestra experiencia al aplicar el método de revisión sistemática descrito por Kitchenham [13] para identificar y seleccionar la evidencia de estudios sobre las actividades que componen el proceso de desarrollo de software *open source*. La selección de estudios primarios ha sido llevada a cabo por parte de dos investigadores. El primero, es un investigador novato que está iniciando su doctorado, mientras que el segundo (supervisor del novato) es experto. En este artículo comparamos las estrategias de selección de estudios primarios seguidas por el novato y el experto y los tiempos invertidos.

El artículo se organiza del siguiente modo. La sección 2 examina el método de revisión sistemática. En la sección 3 se mencionan los trabajos relacionados. La sección 4 describe la experiencia del proceso de revisión sistemática del investigador novato, mientras que la sección 5 presenta la experiencia del investigador experto. En la sección 6 se discuten y comparan los resultados. Por último, en la sección 7 se dan las conclusiones.

2 Método de Revisión Sistemática

La revisión sistemática consiste en identificar, evaluar e interpretar de manera rigurosa toda la investigación relevante disponible para una pregunta, área particular de investigación o fenómeno de interés. Los estudios individuales que contribuyen a una revisión sistemática son llamados estudios primarios [13].

Una revisión sistemática comprende diferentes actividades. Tomando como base las pautas de la literatura médica [1], [2], [7], [12] y adaptándolas a la ingeniería del software, Kitchenham [13] presentó un procedimiento general para el desarrollo de revisiones sistemáticas. Este proceso de revisión sistemática propuesto por Kitchenham es el más usado en la ingeniería del software. En la Tabla 1 se muestran

sus diferentes fases. Estas fases pueden parecer secuenciales, pero es importante notar que muchas de las fases involucran iteración. Una estrategia de búsqueda sistemática especifica las palabras claves y las fuentes usadas para encontrar los estudios relevantes en bases de datos bibliográficas y otras fuentes electrónicas. Esta búsqueda se puede limitar por fechas, revistas, bases de datos, etc.

Tabla 1. Fases del Proceso de Revisión Sistemática [13].

Fases	Descripción
1. Planificar la Revisión	Identifica la necesidad de la revisión y el protocolo a seguir durante su ejecución.
1.1. Identificar la Necesidad de una Revisión Sistemática	Surge de los requisitos de los investigadores para resumir de forma cuidadosa e imparcial, toda la información existente acerca de una pregunta de investigación particular.
1.2. Desarrollar el Protocolo de Revisión	Especifica los métodos que serán usados al realizar una revisión sistemática determinada. Es necesario un protocolo predefinido para reducir el sesgo por parte de los investigadores.
2. Conducir la Revisión	Inicia el propio proceso de revisión sistemática.
2.1. Identificar la Investigación	El objetivo de una revisión sistemática es encontrar tantos estudios primarios relacionados con la pregunta de investigación como sea posible usando una estrategia de investigación imparcial. En esta fase, se define dicha estrategia y se documenta la búsqueda.
2.2. Seleccionar los Estudios Primarios	Una vez han sido obtenidos los estudios primarios potencialmente relevantes, estos deben ser evaluados para determinar su relevancia. Esta evaluación se realiza con base en los criterios de selección definidos en el protocolo.
2.3. Evaluar la Calidad de los Estudios	Es importante evaluar la “calidad” de los estudios primarios, para suministrar aún más criterios detallados de inclusión/exclusión, dar recomendaciones para investigaciones futuras, entre otras.
2.4. Extraer los Datos	Diseñar los formularios de extracción de datos para registrar exactamente la información obtenida de los estudios primarios. Para reducir el sesgo, los formularios de extracción de datos deben ser establecidos durante la definición del protocolo.
2.5. Sintetizar los Datos	Involucra unir y resumir los resultados de los estudios primarios incluidos. La síntesis puede ser descriptiva (no cuantitativa). Sin embargo, algunas veces es posible complementar una síntesis descriptiva con un resumen cuantitativo.
3. Reportar la Revisión	La revisión sistemática es usualmente reportada en un artículo de revista/conferencia, en un reporte técnico o en una tesis doctoral.

Para describir el proceso de revisión sistemática y de síntesis de evidencia de investigación son usados a menudo diferentes términos, entre los que se incluyen: “revisión sistemática”, “revisión de literatura sistemática”, “síntesis de investigación” y “meta-análisis”. Los primeros tres son usados intercaladamente. Sin embargo, la revisión sistemática y el meta-análisis son distintos uno del otro. El meta-análisis es una técnica estadística y metodológica específica que combina datos cuantitativos

[19]. Así, el meta-análisis es una de las diferentes herramientas que pueden ser usadas al realizar una revisión sistemática.

La revisión sistemática es un estudio retrospectivo de cuidadosa observación y está por consiguiente sujeto al error sistemático y aleatorio. Por lo tanto, la calidad y valor de una revisión depende de la medida en que los métodos de revisión científica hayan sido usados para minimizar el error y el sesgo. Ésta es la característica clave que distingue las revisiones narrativas tradicionales de la revisión sistemática [20]. En este sentido, la revisión sistemática puede ser vista como un método de investigación en sí misma, que en muchos aspectos es muy similar a una encuesta, exceptuando que en una encuesta se involucra a personas, mientras que en una revisión sistemática se implica la literatura. Por lo tanto, la fortaleza de los métodos de revisión sistemática radica en su intento explícito de reducir al mínimo las posibilidades de extraer conclusiones erróneas o engañosas, como resultado de los sesgos en los estudios primarios o de los sesgos derivados del propio proceso de revisión.

3 Trabajos Relacionados

Existen estudios que reportan las lecciones aprendidas y las experiencias al llevar a cabo SR [6], [10], [24]. Como el uso de SR en la ingeniería del software es nuevo, varios investigadores han reportado problemas relacionados con diferentes actividades de la SR tales como: la formulación de las preguntas de investigación [6], [24], llevar a cabo la búsqueda de la literatura [10], [24], control de la fiabilidad [24] y aseguramiento de la calidad de los estudios primarios [10], [11], [24].

El proceso en general de SR consume mucho tiempo, en particular, los procesos de planificación, ejecución [18] y de extracción de datos [6]. También existen problemas relacionados con la dificultad en el aprendizaje inicial [5] y la falta de conocimiento del dominio de la investigación [3], [15]. En la estrategia de búsqueda, los problemas presentes radican en que no hay palabras claves estandarizadas [6] y en la dificultad de identificar sinónimos [5]. Los problemas en el proceso de búsqueda surgen de las limitaciones que tienen las bases de datos en cuanto a expresiones booleanas [5], [10], número de caracteres [5] y diferentes sintaxis de búsqueda [6]; por ejemplo, en algunas bases de datos como ACM no pueden limitarse a los resúmenes y títulos [10].

Un resumen de los problemas encontrados al realizar SR ha sido reportado en [22]. Todos los trabajos de investigación usados para elaborar este resumen han sido conducidos por expertos y están basados en sus experiencias. Sólo hay dos excepciones [3], [17], que están basados en entrevistas con profesionales de SR y en una replicación de una SR, respectivamente. Además, Oates y Capper [21] discuten su experiencia en la enseñanza de SR a estudiantes de Máster y Baldassarre et al. [4] reporta su experiencia al involucrar estudiantes de postgrado en el proceso de extracción de datos.

La mayoría de los estudios que reportan SR o experiencias realizando SR se han llevado a cabo desde la perspectiva de los investigadores experimentados. Babar y Zhang [3] examinan la realización de SR desde la perspectiva de los investigadores novatos. Ellos sintetizan las experiencias de varios investigadores, entre los que se incluyen defensores, seguidores y principiantes. Estas experiencias fueron

recolectadas a través de una encuesta basada en entrevistas. Además, reportan prácticas, retos y sugerencias para mejorar las SR. Aunque en su trabajo involucraron a investigadores novatos, las experiencias reportadas no fueron obtenidas de primera mano y carecen de detalle, siendo muy genéricas. Mientras que Riaz et al. [22] detallan explícitamente las experiencias al realizar SR desde la perspectiva de investigadores novatos (estudiantes de doctorado).

Riaz et al. [22] reportan tanto los problemas a los que se enfrentaron los investigadores novatos mientras llevaban a cabo SR, como las acciones tomadas para superarlos. Además, comparan las experiencias del supervisor (experto) de los estudiantes de doctorado, con las experiencias de los novatos para identificar los problemas comunes y diferenciar entre los problemas que enfrentan los expertos y los novatos. Los investigadores novatos requieren de más tiempo que los expertos para llevar a cabo una SR [22]. Casi todos los problemas encontrados por los novatos son el resultado del poco conocimiento del dominio del tema de investigación y/o de SR. Los problemas comunes a los que se enfrentan tanto los investigadores novatos como los expertos, están relacionados con las bases de datos donde se realizan las búsquedas, la evaluación de la calidad y la calidad de los estudios primarios [22].

4 Experiencia del Proceso de Revisión Novata

La actividad más importante durante la fase de planificación de la revisión sistemática es la formulación de la cuestión o cuestiones de investigación que serán respondidas. Todos los demás aspectos del proceso de revisión dependerán de esta(s) cuestión(es).

La revisión sistemática realizada se enmarca dentro de la investigación doctoral de uno de los autores y tiene como objeto responder a la siguiente cuestión: ¿Qué trabajos de investigación existen sobre las actividades que forman parte del proceso de desarrollo de software *open source*? El procedimiento de revisión sistemática seguido es el propuesto por Kitchenham [13].

La búsqueda sistemática se inició con la identificación de las palabras claves y los términos de búsqueda que fueron construidos a partir de la cuestión de investigación. Se realizó inicialmente una búsqueda tradicional, a partir de la cual se obtuvieron algunos artículos, que fueron estudiados para determinar los términos de búsqueda más apropiados para la revisión sistemática y que fueron posteriormente validados y completadas por dos expertos investigadores en el área de ingeniería del software. Los términos de búsqueda empleados fueron:

- OS-SPM **Open Source AND Software Process Model**
- OS-SDP **Open Source AND Software Development Process**
- OS-DP **Open Source AND Development Process**
- FS-SPM **Free Source AND Software Process Model**
- FS-SDP **Free Source AND Software Development Process**
- FS-DP **Free Source AND Development Process**

Las siguientes fueron las bases de datos electrónicas consideradas en la búsqueda sistemática: IEEE Xplore, ACM Digital Library, SpringerLink, Science Direct y Scopus. La Tabla 2 presenta para cada una de las bases de datos electrónicas consideradas, los campos donde fueron aplicados los términos de búsqueda definidos

previamente. Los campos disponibles para realizar la búsqueda no eran siempre los mismos, ya que dependían de las opciones que brindaba cada una de las base de datos. Los campos “*Abstract*” y “*Title*” fueron comunes a todas las bases de datos (BBDD). Posteriormente, se realizó una búsqueda empleando los seis términos en cada uno de los campos definidos según la base de datos.

Tabla 2. Campos de Búsqueda Empleados en cada Base de Datos.

BBDD	Campos de Búsqueda
IEEE Xplore	“Abstract”, “Publication Title”, “ Index Terms ”
ACM Digital Library	“ Abstract ”, “Title”, “Abstract OR Title”
SpringerLink	“Abstract”, “Title”, “All Text”, “ All Text OR Abstract ”
Science Direct	“Abstract”, “Title”, “Keywords”, “ Abstract OR Title OR Keywords ”
Scopus	“ Article Title OR Abstract OR Keywords ”

Con base en el número de registros obtenidos por campo de búsqueda en cada BBDD y usando una heurística se definió el campo que sería usado finalmente en la búsqueda sistemática. Esta heurística dependía del número total de registros obtenidos y de qué tanto diferían entre cada campo de búsqueda. Por ejemplo, en la base de datos ACM Digital Library el número total de registros obtenidos por los campos “*Abstract*” y “*Abstract OR Title*” eran valores muy cercanos, mientras que el número total de registros obtenidos por el campo “*Title*” eran muy pocos comparados con los primeros (“*Abstract*” y “*Abstract OR Title*”). Así, se descarta el campo “*Title*” debido al poco número de registros que aporta. Ahora, como la diferencia de registros entre los campos “*Abstract*” y “*Abstract OR Title*” es muy poca, se decide tomar el campo de búsqueda que proporciona más registros.

En la Tabla 2 se puede apreciar en *negrita* el campo de búsqueda seleccionado para cada base de datos. La estrategia de selección de artículos es la siguiente: en primer lugar, una vez definido el campo de búsqueda para cada BBDD, se procedió a seleccionar qué artículos del grupo obtenido (denominado *Encontrados*) con dicho campo de búsqueda, podrían contener información acerca del proceso de desarrollo de software *open source*. Este grupo de artículos fue etiquetado como *Preseleccionados*. Para incluir un artículo dentro de los preseleccionados, se tuvo en cuenta el título, las palabras claves y el resumen. Estos dos últimos, cuando estaban disponibles en el listado. Cuando se completó el grupo final de artículos *Preseleccionados*, se eliminaron aquellos que se encontraban duplicados entre cada BBDD (es decir, entre los diferentes términos de búsqueda de la misma BBDD) y luego se eliminaron los duplicados entre todas las BBDD. Este resultado, se etiquetó como *Preseleccionados Diferentes*.

Para cada uno de los artículos pertenecientes al grupo *Preseleccionados Diferentes*, se leyó el resumen, la introducción y se realizó un *skimming* (lectura rápida del artículo) general para determinar si describía o no el proceso de desarrollo de software *open source*. El nuevo grupo obtenido con estos artículos, se denominó *Prometedores*. Finalmente, este grupo de artículos *Prometedores* es estudiado para determinar si responde o no a la cuestión de investigación planteada al inicio del proceso de

revisión sistemática. Para algunos de los artículos preseleccionados diferentes se leyó la bibliografía en busca de artículos que pudieran resultar relevantes para la investigación. Estos artículos fueron incluidos dentro del listado de artículos preseleccionados diferentes siempre y cuando no se encontraran previamente seleccionados.

La Tabla 3 muestra el número de artículos de cada uno de los grupos obtenidos (Encontrados, Preseleccionados, Preseleccionados Diferentes y Prometedores) en el proceso de selección de los artículos *Prometedores* por cada BBDD seguido por el investigador novato. Por ejemplo, en la Tabla 3 se puede apreciar que en la BBDD de ACM Digital Library para el término de búsqueda OS-SPM, el número total de artículos que conforman el grupo de *Preseleccionados* fue de 215, mientras que para el término de búsqueda OS-SDP el número total de artículos para el mismo grupo fue de 166. Finalmente, el número total de artículos del grupo *Preseleccionados* de la BBDD ACM Digital Library para todos los términos de búsqueda fue de 616.

Tabla 3. Número de Artículos Obtenidos por el Investigador Novato en cada Grupo por BBDD.

BBDD	Términos de Búsqueda	Encontrados	Preseleccionados	Preseleccionados Diferentes	Prometedores
IEEE Xplore	OS-SPM	71	30	30	1
	OS-SDP	105	58	47	7
	OS-DP	115	51	2	0
	FS-SPM	46	5	2	0
	FS-SDP	25	13	8	1
	FS-DP	25	6	0	0
ACM Digital Library	OS-SPM	1342	215	200	14
	OS-SDP	1507	166	42	2
	OS-DP	1354	117	17	1
	FS-SPM	674	26	3	0
	FS-SDP	675	43	14	2
	FS-DP	568	49	8	0
Springer Link	OS-SPM	14	5	5	0
	OS-SDP	425	53	50	1
	OS-DP	1709	94	65	3
	FS-SPM	1	1	0	0
	FS-SDP	70	22	6	0
	FS-DP	240	56	21	0
Science Direct	OS-SPM	26	8	8	1
	OS-SDP	30	12	8	1
	OS-DP	47	1	0	0
	FS-SPM	37	3	3	0
	FS-SDP	29	3	2	0
	FS-DP	30	3	0	0
Scopus	OS-SPM	342	58	27	1
	OS-SDP	534	79	37	0
	OS-DP	397	15	13	0
	FS-SPM	591	2	1	0
	FS-SDP	839	1	1	0
	FS-DP	401	25	1	0
TOTAL		12.269	1.220	621	35

El proceso descrito anteriormente supuso un gran esfuerzo y una inversión considerable de tiempo como se describe más adelante. Se consideraron 5 bases de datos electrónicas y para cada una se examinaron 6 términos de búsqueda, obteniendo un total de 30 búsquedas diferentes. Estas búsquedas generaron 12.269 artículos sobre los que se aplicó el proceso definido anteriormente, para finalmente obtener 35 artículos *Prometedores*. Es importante comentar que dos de los artículos *Prometedores* fueron obtenidos al leer las referencias de algunos de los artículos *Preseleccionados Diferentes*. Estos dos nuevos artículos no se encontraban en ninguna de las 5 BBDD.

5 Experiencia del Proceso de Revisión Experta

Como se ha mencionado, el experto sólo seleccionó un término de búsqueda, que consideró el que mejor representaba el objetivo de la investigación. Del listado de artículos encontrados en cada BBDD a partir del término de búsqueda OS-SDP, el investigador experto realizó una selección de artículos empleando como criterios de selección el título, los autores y las revistas o proceedings donde fue publicado. El título debía contener alguna de las palabras o combinación de las mismas del término de búsqueda, siendo las más importantes: *process* o *development* o *development process* o *software development process*. Los autores debían ser relevantes en el área de proceso y *open source*. Las revistas o proceedings debían ser importantes en el área de la ingeniería del software. Estos artículos constituyen el grupo de *Preseleccionados*.

Finalmente, se eliminaron los artículos duplicados del grupo de *Preseleccionados*, obteniendo así el grupo de artículos *Preseleccionados Diferentes*. El investigador experto consideró que el título, los objetivos del abstract y las palabras claves del artículo tienen que estar relacionados. Si están relacionados, se emplea como criterio leer el abstract, si éste respondía a los objetivos de la propia investigación y las conclusiones determinaban la definición de un proceso de desarrollo *open source*, se realizaba entonces una lectura rápida de la introducción y las conclusiones para determinar si el artículo era considerado prometedor. Este grupo de artículos fue denominado *Prometedores*. La Tabla 4 muestra el número total de artículos por grupo (Encontrados, Preseleccionados, Preseleccionados Diferentes y Prometedores) obtenidos en el proceso de selección de los artículos *Prometedores* del investigador experto. Esta tabla es análoga a la Tabla 3, del investigador Novato.

Tabla 4. Número de Artículos Obtenidos por el Investigador Experto en cada Grupo por BBDD.

BBDD	Términos Búsqueda	Encontrados	Preseleccionados	Preselecc. Diferentes	Prometed.
IEEE Xplore	OS-SDP	105	24	25	8
ACM Digital Library	OS-SDP	1507	29	16	16
SpringerLink	OS-SDP	425	10	10	1
Science Direct	OS-SDP	30	14	4	2
Scopus	OS-SDP	534	28	1	1
TOTAL		2601	95	55	28

Cómo se puede apreciar en la Tabla 4, el número total de artículos *Prometedores Diferentes* encontrados por el investigador experto fue de 28. En el siguiente apartado, haremos una comparativa de los resultados obtenidos por el investigador novato y el experto.

6 Análisis Comparativo y Lecciones Aprendidas

La Tabla 5 presenta una comparativa entre el número de artículos *Prometedores* que cada BBDD aporta del total de *Prometedores* encontrados. En esta tabla, cada BBDD es considerada de forma independiente de las demás. Así, es posible que existan artículos *Prometedores* duplicados entre BBDD.

Tabla 5. Comparativa de Artículos Prometedores del Investigador Novato y el Experto.

BBDD	Nro. Prometedores Investigador Novato	Nro. Prometedores Investigador Experto
IEEE Xplore	9	8
ACM Digital Library	25	18
SpringerLink	4	1
Science Direct	2	2
Scopus	15	14

El investigador novato encontró un total de 35 artículos *Prometedores Diferentes* (ver Tabla 3), mientras que el experto encontró 28 (ver Tabla 4), es decir, el investigador experto encontró el 80% del total de artículos *Prometedores* del investigador novato, pero haciendo uso de sólo un término de búsqueda, analizando así muchos menos artículos en cada paso del proceso de selección, con el consecuente ahorro de tiempo. Dentro de los 7 artículos que no encontró el experto se encuentran los 2 artículos prometedores obtenidos por el novato al leer las referencias de algunos de los artículos del grupo de *Preseleccionados*. Estos 2 artículos nunca hubieran sido encontrados por el experto aunque incluyera todos los términos de búsqueda.

En la Figura 1 se puede apreciar la comparativa de los criterios de selección de artículos empleados tanto por el investigador novato como por el experto durante el desarrollo de la SR. En la Tabla 6, se puede apreciar una comparativa del tiempo empleado, tanto por el investigador novato como por el experto, en cada actividad del proceso de selección de estudios primarios. Para cada actividad del proceso de selección se presenta el número total de artículos obtenidos y el tiempo empleado.

En la búsqueda sistemática el investigador novato partió de un total de 12.269 artículos, mientras que el experto solo contaba con 2.601 artículos (recordemos que sólo seleccionó un término de búsqueda), es decir con una quinta parte del número de artículos del novato. Así, el tiempo invertido por el investigador experto, se redujo a una quinta parte del tiempo del novato. El tiempo total invertido por el investigador novato para obtener los 35 artículos *Prometedores* fue de 194 horas, lo que equivale a 24 días por 8 horas diarias de trabajo. Por su parte, el investigador experto empleó un total de 21,4 horas en obtener 28 artículos *Prometedores*. Como se puede apreciar en la Tabla 6, la estrategia de búsqueda del investigador experto es más eficiente que la

del investigador novato, pues invirtiendo una décima parte del tiempo del novato obtuvo un 80% de los artículos *Prometedores*.

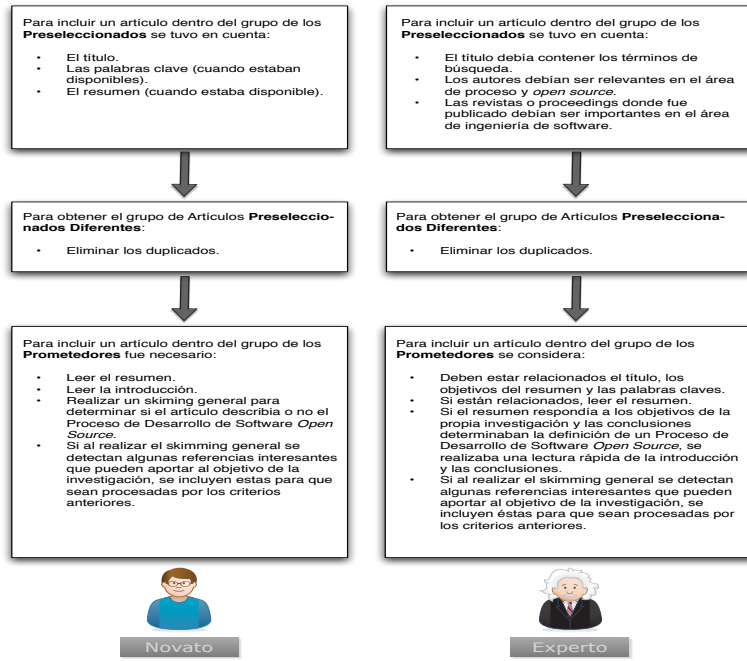


Fig. 1. Comparativa de Criterios de Selección Empleados por el Investigador Novato.

Tabla 6. Comparativa de Tiempos entre Investigador Novato y Experto.

Actividad	Investigador Novato		Investigador Experto	
	Nro. Total Artículos	Tiempo (en horas)	Nro. Total Artículos	Tiempo (en horas)
Realizar la búsqueda en cada una de las BBDD y obtener el listado de artículos encontrados.	12.269	31,0	2.601	6,5
Seleccionar en los listados obtenidos en el punto anterior, los artículos que podrían ser relevantes (<i>Preseleccionados</i>) para la investigación, eliminando repetidos entre términos de búsqueda.	1.220	22,5	95	2,7
Eliminar los artículos repetidos entre los términos de búsqueda de la misma BBDD.	N/A	16,0	N/A	3,5
Del listado obtenido en el punto anterior, eliminar los artículos repetidos entre las BBDD (<i>Preseleccionados Diferentes</i>).	621	16,0	55	0,5
Descargar todos los artículos <i>Preseleccionados Diferentes</i> , para su posterior análisis.	N/A	47,5	N/A	4,2
Determinar cuáles artículos <i>Preseleccionados Diferentes</i> son considerados <i>Prometedores</i> .	N/A	61,0	N/A	4,0
Tiempo Total (en horas)		194,0		21,40
Tiempo Total (en días)		24,3		2,68

La Figura 2 ilustra el número de artículos *Prometedores* aportados por cada BBDD, según los datos aportados por el investigador novato. El número total de artículos *Prometedores* fue de 35, así por ejemplo, la BBDD IEEE Xplore contiene 9 de estos artículos. Como se puede apreciar en la Figura 2, la mejor BBDD es ACM Digital Library pues es la que más artículos *Prometedores* aporta, con un total de 25. Mientras que la peor BBDD es Science Direct aportando tan sólo 2.

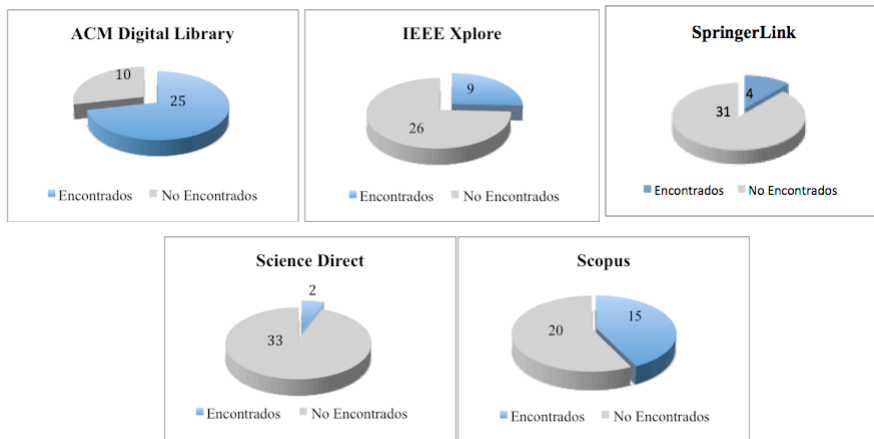


Fig. 2. Número de Artículos Prometedores Aportados por Cada BBDD según el Investigador Novato.

Por último, hemos comprobado que el mejor término de búsqueda para 4 de las 5 BBDD es el término OS-SDP, coincidiendo con el término de búsqueda seleccionado por el investigador experto. Reafirmando así, que la experiencia y el conocimiento son factores importantes al momento de realizar una búsqueda sistemática. Además, con base en los datos obtenidos tanto por el investigador novato como por el investigador experto, podemos afirmar que la mejor combinación de BBDD y término de búsqueda es ACM Digital Library y OS-SDP, respectivamente.

7 Conclusiones

En este artículo realizamos una comparativa entre un investigador novato (estudiante de doctorado) y un experto (supervisor del estudiante) al aplicar el método de SR descrito por Kitchenham, para obtener el conjunto de artículos *Prometedores* que evidencien las actividades que componen el proceso de desarrollo de software *open source*. Las SR del investigador novato y el experto son comparadas para determinar las similitudes y diferencias entre los criterios de selección empleados por cada uno, así como el tiempo empleado en cada actividad de la SR. Si bien es cierto que las revisiones sistemáticas se realizan de manera colaborativa, en esta investigación comparamos el desempeño de un investigador novato con un investigador experto en la selección de estudios primarios.

El investigador novato emplea como criba un procedimiento exhaustivo, no contextual, mientras que el experto utiliza un procedimiento contextual. El novato usa todos los seis términos de búsqueda definidos en el protocolo y el experto sólo un término de búsqueda. Este término de búsqueda seleccionado por el experto es el que mejor representa los objetivos de la investigación y coincide además con el término de búsqueda que más artículos prometedores encuentra al compararlos con los demás términos empleados por el novato. Esto evidencia la importancia del conocimiento del dominio de la investigación y la experiencia del experto. Ambos investigadores usan las mismas bases de datos para realizar las búsquedas, siendo ACM Digital Library la que más artículos prometedores aporta.

El novato al contar con un universo de búsqueda más amplio, encuentra un número mayor de artículos prometedores que el investigador experto. El experto encontró un 20% menos de artículos de los que encontró el novato, pero invirtió sólo una décima parte del tiempo total empleado por el novato. Si el investigador experto hubiese considerado todos los términos de búsqueda, probablemente el porcentaje de artículos que no encontró con respecto al novato sería menor. El grupo de artículos *Prometedores* que encontró el investigador experto está contenido en su totalidad en el grupo de artículos *Prometedores* encontrados por el investigador novato, es decir, hay un solapamiento de artículos *Prometedores*. La actividad en la que el novato invirtió más tiempo fue en la de determinar cuáles eran los artículos prometedores. En esta actividad el novato empleó más del 30% del tiempo total. La actividad que demandó la mayor cantidad de tiempo por parte del experto (30% del tiempo total) fue realizar la búsqueda en cada una de las bases de datos para obtener el listado inicial de artículos. Estos resultados sugieren que la experiencia del experto influye en la revisión y proponemos involucrar al investigador experto en el desarrollo del protocolo de revisión, selección y evaluación de calidad de estudios primarios.

La validéz del estudio presentado en este artículo se ve amenazado por el número de personas que participaron en el mismo (solo dos), pero a la vez supone una primera aproximación a trabajos de investigación que vayan en la línea de proponer mejoras en la forma en que se desarrolla el trabajo colaborativo entre investigadores novatos y expertos. Como trabajos futuros, está el replicar este estudio con más personas, pero usando el mismo protocolo.

Para mejorar el proceso de SR llevado a cabo por un novato, es necesario:

- Definir los términos de búsqueda en conjunto con un investigador experto.
- Seleccionar con la ayuda del experto el (los) termino(s) de búsqueda que serán empleados.
- Suministrar al investigador novato información acerca de las revistas o proceedings más importantes según el dominio de las preguntas de investigación planteadas, con el fin de facilitar la selección de artículos.
- Priorizar por parte del investigador experto el listado final de artículos prometedores seleccionados, con el objetivo de ordenar el análisis de los mismos. Esta prioridad también puede emplearse como criterio de evaluación de calidad de los artículos.

En concreto, se pueden dar las siguientes guías:

- Si hay tiempo y seguridad que el novato va a hacer la investigación exhaustivamente, entonces no existirá el problema de pérdida de artículos relevantes, por el contrario habrán muchos más.
- Si no hay tiempo, va a ser peligroso porque el novato no distinguirá tal como lo hace el experto los criterios adecuados de selección de artículos.
- En este caso, la solución es combinar bien las habilidades para que en poco tiempo puedan encontrarse la mayoría de los artículos relevantes.
- Si lo puede hacer un experto aunque no tenga suficiente tiempo y no lo haga exhaustivamente, se puede obtener un conjunto de artículos relevantes en forma satisfactoria porque su experiencia también cuenta y tiene su impacto en la revisión.

Agradecimientos. Esta investigación ha sido financiada en el marco del proyecto titulado “Generación de Evidencias mediante Combinación de Resultados Experimentales”, Código: TIN2008-00555, por el Ministerio de Ciencia e Innovación de España.

References

1. ANHMRC: How to Review the Evidence: Systematic Identification and Review of the Scientific Literature. Australian National Health and Medical Research Council (2000)
2. ANHMRC: How to Use the Evidence: Assessment and Application of Scientific Evidence. Australian National Health and Medical Research Council (2000)
3. Babar, M.A., Zhang, H.: Systematic Literature Reviews in Software Engineering: Preliminary Results from Interviews with Researchers. In: Empirical Software Engineering and Measurement (ESEM'09), pp. 15--16. Lake Buena Vista, FL, USA (2009)
4. Baldassarre, M.T., Caivano, D., Kitchenham, B.A., Visaggio, G.: Systematic Review of Statistical Process Control: An Experience Report. In: Evaluation and Assessment in Software Engineering (EASE'07), pp. 1--9. UK. British Computer Society (2007)
5. Biolchini, J.C., Mian, G.M., Natali, A.C.C., Conte, T.U., Travassos, H.H.: Scientific Research Ontology to Support Systematic Review in Software Engineering. *Advanced Engineering Informatics*. 21, 133--151 (2007)
6. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from Applying the Systematic Literature Review Process Within the Software Engineering Domain. *The Journal of Systems and Software*. 80(4), 571--583 (2007)
7. Cochrane Collaboration: Cochrane Reviewers' Handbook. Version 4.2.1. December (2003)
8. Cooper, H. Hedges, L.V. (Eds.): *Handbook of Research Synthesis*. New York: Russell Sage Found (1994)
9. Dixon-Woods, M., Agarwal, S., Jones, D., Young, B., Sutton, A.: Synthesising Qualitative and Quantitative Evidence: A Review of Possible Methods. *The Journal of Health Services Research & Policy*. 10(1), 45--53 (2005)
10. Dybå, T., Dingsøyr, T., Hanssen, G.K.: Applying Systematic Reviews to Diverse Study Types: An Experience Report. In: First International Symposium on Empirical Software Engineering and Measurement (ESEM'07), pp. 225--234. Madrid, Spain (2007)
11. Dybå, T., Dingsøyr, T.: Strength of Evidence in Systematic Reviews in Software Engineering. In: Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'08), pp. 178--187. Kaiserslautern, ACM, New York (2008)

12. Khan, K.S., Riet, G., Glanville, J., Sowden, A. J., Kleijnen, J. (Eds): Undertaking Systematic Review of Research on Effectiveness. CRD's Guidance for those Carrying Out or Commissioning Reviews. CRD Report Number 4 (2nd Edition), NHS Centre for Reviews and Dissemination, University of York, IBSN 1 900640 20 1, March (2001)
13. Kitchenham, B.A.: Procedures for Performing Systematic Reviews. Keele University, Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1 (2004)
14. Kitchenham, B.A., Dyba, T., Jorgensen, M.: Evidence-Based Software Engineering. In: 26th International Conference on Software Engineering (ICSE'08), pp. 273--281 (2004)
15. Kitchenham, B.A.: Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. Technical Report S.o.C.S.a.M. Software Engineering Group, Keele University and Department of Computer Science University of Durham (2007)
16. Kitchenham, B.A., Mendes, E., Travassos, G.H.: Cross Versus Within-Company Cost Estimation Studies: A Systematic Review. *IEEE Transactions on Software Engineering*. 33(5), 316--329 (2007)
17. Kitchenham, B.A, Breerton, P., Turner, M., Niazi, M., Linkman, S., Pretorius, R., Budgen, D.: The Impact of Limited Search Procedures for Systematic Literature Reviews - A Participant-Observer Case Study. In: 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09), pp. 15--16. Lake Buena Vista, FL, USA (2009)
18. Mian, P., Conte, T., Natali, A., Biolchini, J., Travassos, G.A.: Systematic Review Process for Software Engineering. In: 2nd Experimental Software Engineering Latin American Workshop (ESELAW'05), pp. 1--6, Brasil, (2005)
19. Miller, J.: Applying Meta-analytical Procedures to Software Engineering Experiments. *The Journal of Systems and Software*. 54(1), 29-39 (2000)
20. Mulrow, C., Cook, D. (Eds.): *Systematic Reviews: Synthesis of Best Evidence for Health Care Decisions*. Philadelphia: Am. College of Physicians (1998)
21. Oates, B.J., Capper, G.: Using Systematic Reviews and Evidence-Based Software Engineering with Masters Students. In: 13th International Conference on Evaluation and Assessment in Software Engineering (EASE'09), pp. 20--21 (2009)
22. Riaz, M., Sulayman, M., Salleh, N., Mendes, E.: Experiences Conducting Systematic Reviews from Novices' Perspective. In: 14th International Conference on Evaluation and Assessment in Software Engineering (EASE'10), pp. 1--10. Keele University, UK (2010)
23. Scacchi, W.: Free and Open Source Development Practices in the Game Community. *IEEE Software*. 21 (1), 59--66 (2004)
24. Staples, M., Niazi, M.: Experiences Using Systematic Review Guidelines. *The Journal of Systems and Software*. 80(9), 1425--1437 (2007)