

Slrtool: A Tool to Support Collaborative Systematic Literature Reviews

Balbir S. Barn, Franco Raimondi, Lalith Athappian and Tony Clark

Department of Computer Science, Middlesex University, London, U.K.

Keywords: Systematic Literature Review, Software Tool, Collaborative Research Process.

Abstract: Systematic Literature Reviews (SLRs) are used in a number of fields to produce unbiased accounts of specific research topics. SLRs and meta-analysis techniques are increasingly being used in other fields as well, from Social Sciences to Software Engineering. This paper presents the SLRTool tool - an open source, web-based, multi-user tool that supports the SLR process for a range of research areas. The tool is available at <http://www.slrtool.org> and is developed using a model-driven approach to enable its adaptation to different disciplines. The functionality of the tool is presented in the context of support for various phases of the SLR process. The use of the tool is illustrated by means of a simulated SLR aiming to map out existing research in the domain of Enterprise Architecture (EA). Commentary on the use of the tool and potential additional benefits are discussed, for example, the role of the tool in non-medical meta-studies. The SLRTool tool supports all phases of the SLR process and lends itself to creating and supporting research communities geared to SLR oriented activities. In particular, the tool could be suitable for the novice researcher community.

1 INTRODUCTION

The systematic literature review (SLR) is an increasingly important research strategy for extracting new knowledge from existing research data. The SLR process originated from the medical field but is now common across many disciplines, including Social Sciences and Computer Science. The SLR process is a well defined, methodical approach for identifying, assessing and evaluating existing published studies to support the exploration of one or more specific research questions.

The leading effort to formalize the SLR process has been carried out by Kitchenham et al for Software Engineering (Kitchenham and Charters, 2007). In their work they propose a set of guidelines which are obtained by synthesising the guidelines provided in other fields, including the Cochrane Handbook (<http://www.cochrane.org>).

In Software Engineering (SE) and Information Systems, SLRs have been published on a range of topics including: open source software development (Hauge et al., 2010), classifying requirements specification errors (Walia and Carver, 2009), motivations of developers (Beecham et al., 2008) amongst others. The general usage of SLRs to investigate evidence base Software Engineering led to a meta study where Kitchenham et al. assessed the impact of SLRs on

SE practice (Kitchenham et al., 2009). Findings from this study suggest that the spread of topics covered by SLRs up to 2009 were fairly limited and mainstream Software Engineering topics were not well represented. Such areas could benefit from mapping studies of domains in the manner of Jorgensen and Shepherd (Jorgensen and Shepperd, 2007).

SLRs are different from expert-led ad-hoc literature reviews by providing definitions and supporting documentation for review protocols prior to conducting the review such as: research questions; descriptions of the search strategy; descriptions of criteria to be used to assess each published study and procedures to be used to perform the review (Kitchenham and Charters, 2007).

Experiences of SLRs reported in the SE domain suggest that in general, the SLR process is very resource intensive with particular problems associated with data extraction, management of large data and collaborative activities associated with conducting SLRs, with reviewers playing multiple roles during the SLR process. Further, the relative newness of the use of SLRs in the SE domain indicates that there also issues in provision of appropriate support for novice reviewers. As a consequence, it becomes essential that the SLR process can be easily validated and repeated, if research outputs from the SLR process are to carry an appropriate level of confidence. Validation

tion and repeatability are guaranteed by the compliance with a certain protocol, which is clearly stated in medical SLR following the Cochrane method. A further requirement supporting the need for validation is the ability to demonstrate traceability between decisions.

1.1 Contribution of this Paper

Currently, there is little evidence of using bespoke specialist tools to support SLRs in Software Engineering with respect to protocol definition, data collection and analysis, hence the opportunities for performing meta-level SLRs are consequently also limited. This paper proposes requirements for tools to support the SLR process based on existing SLRs performed within the SE domain and provides an assessment of existing options available. The main contribution is the introduction of the SLRtool that is specifically aimed at providing support for a collaborative SLR whilst addressing many of the key challenges currently prevalent around the SLR process. We suggest that provision of tool support is paramount for embedding a particular practice such as SLR in the community.

The remainder of the paper is structured as follows: In the next section we describe the key challenges associated with the SLR process. Section 3 presents related work on building tools for supporting the SLR process and proposes a set of requirements for SLR tools across a range of possibilities. Section 4 provides a description of the key functionality of the SLRTool and illustrate its use for a small example mapping out the literature for Enterprise Architecture. In Section 5, the conclusion is presented along with remarks about the directions of future work.

2 CHALLENGES OF THE SLR PROCESS

One of the consequences of a growing number of SLRs conducted in Software Engineering and elsewhere is the need for consistency. In 2007, Brereton et al. reported on lessons learnt from applying the SLR process within the Software Engineering domain (Brereton et al., 2007), which can be consolidated into a set of requirements such as: support continuous revision of research questions and to keep track of changes; support multiple reviewers for the process and to track their usage; ability to tailor and re-configure the methodology for a particular review; support external validation of the review process; support multiple search strategies as different databases

such as ACM and IEEE use different approaches; and keep detailed record of decisions. Elsewhere Riaz et al. report on how formulation of research questions, quality assessment and the time consuming nature (Babar and Zhang, 2009) of the SLR process create challenges (Riaz et al., 2010). The same study also commented on experiences of novice researchers who also struggled with formulation of research questions, with carrying out the reviews, and in particular with the time factors involved in carrying out the reviews and extracting data. Others have reported on the need to manage how specific roles are allocated and the activities associated with roles, for example the need to cross-check and validate criteria such as that for inclusion / exclusion or quality (Hossain et al., 2009).

Staples and Niazi (Staples and Niazi, 2007) reported on their experience of conducting an SLR using Kitchenham's guidelines. Of particular interest is their analysis in the potential for automation and support for SLRs. They used and benefitted from only a basic level of automation: simple tabular word processing, simple spreadsheets and use of statistical analysis packages. Whilst they viewed the potential prospects for tool support as "*very dim ...to support all phases of arbitrary systematic reviews*" they acknowledge that such mechanisms may provide "a nexus for the improvement of systematic review methodology" and contribute to a notion of a central index of systematic reviews for software engineering. This last point is particularly pertinent in the context of being able to perform meta-level systematic reviews across sub-domains within software engineering.

While not directly concerned with the SLR process itself, the UBIRD study by Stelmaszewska et al (Stelmaszewska et al., 2010) investigated the information seeking skills of students in the business domain. Findings from the study reported that students: used personal / social networks to refine search strategies; experienced significant difficulty in storing information collated from the searching activities (time being one such factor); and struggled with user interfaces of different digital repositories. The latter is consistent with Brereton et al's comment on the differences between the ACM and IEEE approaches.

Summarising, the current experience of using the SLR process suggests strongly that there is need for tool support in: configurable protocol definition while executing the SLR process by multiple stakeholders; reducing the time spent on managing the activities in carrying out the reviews and extracting the data for storage for subsequent review and analysis.

3 RELATED WORK IN TOOL BUILDING

Despite the increasing importance of SLRs to the Software Engineering and to other communities, there has been remarkably scant effort made to develop supporting tools for the SLR process. Several tools have been built: for instance, the SM-VTM tool provides a text mining based approach to visualising the SLR clustering activities but does not provide support for an end to end SLR process (Felizardo et al., 2010). The SLR tool developed by Fernandez-Saez et al. (Fernandez-Saez et al., 2010) focuses on the SLR process but some of the technical hurdles associated with the tool (such as installation and pre-configuration requirements, single user, and desktop-only version) limit its use (Hernandes et al., 2012). Further, there has been no other evidence of its subsequent development. The StART tool reported by Hernandez et al (Hernandes et al., 2012) presents less of these technical limitations but remains limited to the Windows platform and does not readily lend itself to multiple users participating in the SLR process. Most closely related to the SLRTool is SLurP, (Bowes et al., 2012) which has a functionality similar to our SLRTool, provides multiple user access and is a web based system implemented on top of the Linux distribution Debian and developed using Java (while we build upon the standard chain Apache+PHP+Mysql). The key difference is that our tool is fully open-source and implements a model-driven approach, in order to be adapted to a range of disciplines.

The medical domain, by virtue of the Cochrane organisation, has tool support available from *RevMan*¹. Even in the case of *RevMan*, the tool only provides a stand-alone text editor enriched with functionalities that guarantee that all the steps in the review protocol are followed. Crucially, there is no support for collaborative work.

Related to tools supporting the SLR process are those which are essentially bibliography management systems. Some tools such as EndNote, Procite and Reference Manager focus on managing bibliographic databases and citation of references from such databases in the context of document preparation using desktop based applications. Other such as Zotero and Mendeley provide web based capability and in addition provide some level of Web 2.0 collaborative activities including, for example, the sharing of bibliographic data between users. However, the key areas of collaboration, traceability of decisions, repeatability and ease of collection proposed by the functions of

¹<http://ims.cochrane.org/revman>.

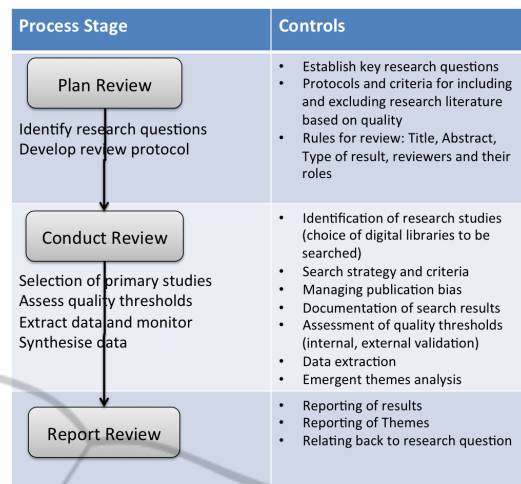


Figure 1: The SLR process.

the tool described here are not collectively available in these technologies. Thus, generally the key weakness of these types of tools is the lack of support for the actual SLR process and there is only the possibility of manual record keeping of SLR data.

Both the lessons learnt from conducting SLRs as detailed in the Challenges section and the review of current tools formed the basis of the requirements for the software tool introduced in this paper.

4 SLRTool FUNCTIONALITY

The SLRTool is designed to support SLRs conducted using the guidelines described by Kitchenham et al. (Kitchenham, 2004; Kitchenham and Charters, 2007), but is open to extensions, as described in Section 4.5. There are three key stages at the core of the guidelines: Plan Review; Conduct Review and Report Review. These are shown in Figure 1, along with some of the key activities within each stage.

In the following sections we introduce an illustrative case study of a SLR for Enterprise Architecture. The case study services to provide a demonstration of how the SLRTool tool supports the SLR process and some of the salient requirements.

4.1 Case Study

We first introduce our SLR case study on Enterprise Architecture (EA). EA is the means by which the essentials of a business, its IT and its evolution, and analysis is performed. It is a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise's organisational structure, business processes, information systems and in-

Category	Value
Research Topic	Language Design, Method, Framework, Technology
Research Approach	Simulation, Case Study, Quantitative, Qualitative, Conceptual, Survey, History based Evaluation, Review, Experiential, Other
Study Context	Student/Student Project, Industrial project, Higher Education Domain, Not Relevant
Data Set	Authentic Historical Data, Simulated Data, Commercial Data

Figure 2: Classification Framework for the Enterprise Architecture SLR project.

frastructure. To date, there has been no SLR of this topic. In particular, the type of mapping study in the manner of Jorgensen and Shepherd (Jorgensen and Shepperd, 2007) would be particularly beneficial. Thus, the SLR study has the following research questions:

1. What are the main research topics being addressed by EA research?
2. What are the main methodological approaches being used in EA research?
3. What are the main publication outlets for research in EA?

As the study is intended to focus on prevalent research activity in EA, resources such as literature opinion papers, panel discussions, commercial white papers are to be excluded. Categorisation of resources is however, particularly important as the study aims to address the main research areas and the approaches taken, hence categories such as: Research Topic (values are: Language design, Method, Framework, Technology), Research Approach (values are: Simulation, Conceptual, Survey, Experiential and others) are needed. The full framework is shown in Figure 2. We can see how research outputs can be classified using these categories to help contribute towards answering research questions 1 and 2.

4.2 Plan Review Functionality

The Plan Review stage aims to identify research questions (examples of research questions are listed in the previous section) and develop the review protocols such as definition of quality criteria, inclusion / exclusion of resources, reviewers and their roles and the definition of a classification framework. The SLRTool defines a review as a project and provides various dialogues for defining research questions, and the cat-

egories of resources that contribute to the classification framework, see Figure 3. The details of the protocol may differ in different disciplines: Section 4.5 below describes how to accommodate this variability in the SLRTool. Repeating a particular SLR is then a matter of importing a particular protocol and reusing it. Currently the following requirements are relevant:

1. Ability to define Research Questions, Inclusion/Exclusion Criteria, Categories for refining research questions and the values such properties can take.
2. Ability to define dependencies and other relationships between criteria and categories.
3. Use of ontologies to provide a structured language for defining categories, criterias and their values.
4. Ability to define roles to multiple users.

Tool capability that can provide a structured language for describing the various categories, criteria and values in the form of an ontology is an advanced capability that is currently planned for the SLR Tool but is not yet available.

4.3 Conduct Review

This is the most time consuming section of the SLR process. Activities include: defining the sources of data (the digital resources, journals etc.); the search strategies; data extraction; assessing the quality of resources; performing classifications based on the reading of the papers (resources) and of course storing and managing the data. The SLRTool provides tool functionality for these activities which are optimised for reducing the time resource spent on this.

Some of the key functions (shown in figure 3.) include:

1. Ability to define search criteria independent of target resource database with interfaces to bibliographics support tools such as Google Scholar.
2. Automatic extraction of BiBTeX data of all resources located, along with automatic download of full text PDFs subject to permissions for the host institution. These are done as batch downloads on selected resources.
3. Ability to include / exclude resources and to record reasons based on previous defined criteria.
4. Ability to categorise resources according to the review protocols defined earlier.
5. Support for multiple reviewers on a SLR project.
6. Ability for reviewers to comment on decisions made and to assess reviewer actions.

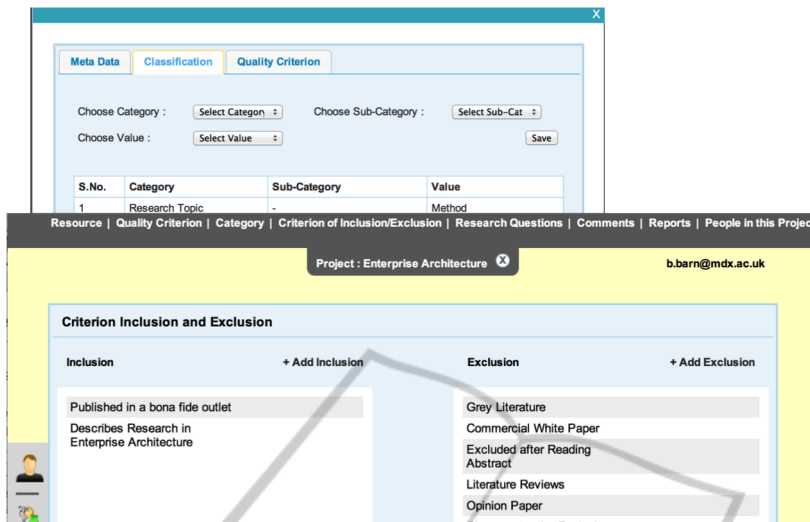


Figure 3: Classification of resources and Inclusion/Exclusion criteria.

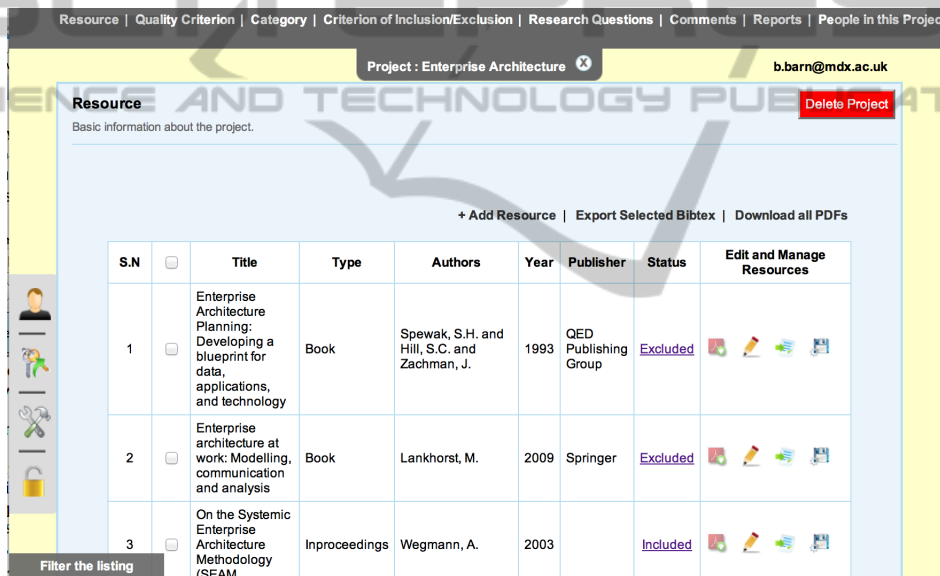


Figure 4: Resources.

We have planned development activity to increase support for this phase of the SLR process to include requirements for delivering further collaborative actions such as: difference analysis and Reconciliation services between users including traceability between decisions and the ability to support summarisation and keyword extraction for automated classification.

4.4 Report Review

This stage of the SLR process includes both descriptive statistics and more detailed analysis linking back to the themes and research questions intended to be explored in the review. The SLRTool provides functions to help in these activities as follows: Report

summaries of publisher details; Report summaries of publication dates; Reporting on Inclusion and Quality criteria; and Exporting of data to external systems.

As the tool is still being developed, other reports planned include reporting on the classification framework and the relationships linked to the research questions. Other scheduled activity includes the ability to link classification decisions to specific sections in the resource documents and provision of canned analysis techniques.

Reports from the case study used in this paper illustrate the ease by which these can be produced: Figure 5 illustrates the distribution of publishers and the distribution of quality criteria for the papers identified in our study.



Figure 5: Reporting.

4.5 Technical aspects of the tool

In our analysis of existing technologies in this area, all the existing tools had some specific technical limitations around platform dependence and lack of multi-user access. This was influential in the approach that we took towards the selection of the technical architecture of the tool.

The SLRTool uses a standard open source web development platform comprising the Apache web server, MySQL and PHP. We have successfully installed it on a number of operating systems, including Linux, Mac, Windows, and even on a Raspberry Pi: the latter installation has the benefit of being mobile so that the *whole* set of repositories can be moved between institutions, if necessary.

Another key element in our approach was an early realisation that the potential for substantial automated analysis of reviews could be made possible if we took a model based approach to tool development. Thus we see the concepts underpinning the tool as an example of a domain specific language. The overall SLR process is defined in a similar way, and all the code is generated from this model. As a result, we are able to accommodate changes in the protocol and in other parts of the SLR process just by modifying the model, and then generating (and re-using, where available) code to support different disciplines.

4.6 Current Limitations

We consider the tool as a prototype: the version available on-line implements a connector with

Google Scholar, but we are planning to include other databases (subject to institutional agreements etc.). Google Scholar was given a dominant role as the study on information seeking behaviours of students provided empirical evidence of its widespread use in Higher Education Institutions (Stelmaszewska et al., 2010). The current operation of the use of the connector to Google Scholar is available on the opening project screen. Users are able to search for a variety of search parameters and note results arising from the searches. By limiting searching to just Google Scholar, we recognise that papers will be missed because of issues such as how different digital libraries configure their search facilities. One side effect though, is that it does provide a consistency in the protocols across SLRs. Examples of searching are shown in Figure 6.

We envisage SLR tool development activity to continue to progress along the spectrum we outlined earlier where there is increasing sophisticated support for the activities that we consider to be key: Management of resources (especially auto-upload); Recording evidence (the classification of resources and linking them to specific evidence within a resource) and Collaboration (how better to support team and inter-team working). These activities are essential if there is to be standardisation across SLR cases.

Other areas of the tool that we intend to develop further include more reporting options of the data collected, in particular the classification frameworks used. An embedded help guide can be particularly useful to novice researchers; this help feature can direct users towards the correct use of features of the tool in the context of the SLR process. We are currently developing these on-line help facilities.

Staples and Niazi in their report on their experiences of performing SLRs made specific reference to the notion that software tools can provide a nexus for the improvement of systematic review methodology. We have interpreted this beyond their original intention and propose that model driven engineering practice may provide valuable inroads in developing functionality that can specifically address functions that allow configurability of SLR processes. By designing the SLRtool using model driven principles we are aiming to support specification of an SLR process for a particular domain such that the specification can derive an instance of the tool for use for that domain.

Finally, we recognise that the tool has only been used in a small scale, non-independent manner. It is our intention to carry out evaluation studies with the tool both to refine the tool and to examine how the SLR process is possibly modified as a result of using tools.

5 FINAL REMARKS

The current literature indicates that a well developed SLR that provides reliable outputs is a difficult and time consuming task, mainly because of the management and administration of large complex volumes of data. When this is coupled with multiple users as part of a team activity then the problem increases. In this paper we have sought to describe the development of the SLRTool, which aims to address these issues and to enhance the capabilities required in performing an SLR.

While we have been motivated by our our experience as researchers struggling with bottlenecks and inefficiencies in attempting to carry out SLRs, we think that expert reviews typically found in research papers in general could also benefit from such a tool. Given this, we anticipate building a community around our approach, and future versions of this tool will have an appeal to many researchers in a range of research areas. We have noted that Brereton et al reported that the Simula Lab was the leading research lab on conducting SLRs because they had developed very effective research procedures which were embedded in the organisation. Such procedures require tool support and the provision of the SLRTool and its usage in the research community could help developing capability similar to the Simula Lab elsewhere. In particular we are keen to embed the tool as part of the research training undertaken by doctoral students at our institution and elsewhere.

It is likely that SLRs will become more pervasive and be relevant to most (if not all) disciplines, hence software tools that can support the SLR process will be a necessary requirement. Further, as different disciplines approach the SLR process definition differently (the protocols and activities for example), an important challenge that arises is the need to support different flavours of SLR methodologies. The project reported in this paper has concentrated on a particular style of SLR methodology; however, the fundamental design principles that underpin the software are based on model driven software engineering practice, and future versions of the software will be able to support user configurable SLR methodologies. Related to this, the setting up of an SLR protocol, the criteria for inclusion/exclusion, classifiers and so on can be re-used across specific sub-domains, leading to systematic meta studies of SLRs. For example, it is not inconceivable that SLRs aimed at mapping out sub-domains such as Component Based Design, Service-Oriented Architecture, Event driven Architecture (all broadly related to software architecture) could utilise the same SLR protocol enabling a much richer meta-analysis of the set of SLRs. We are progressing fur-

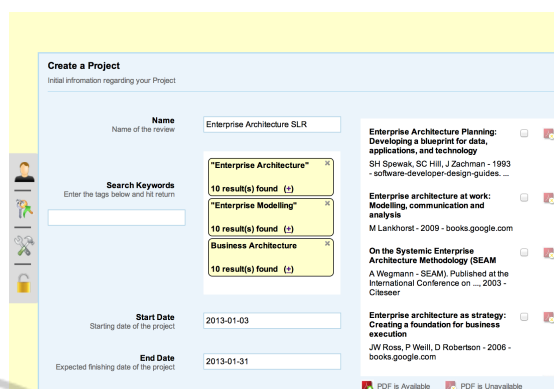


Figure 6: Searching Google Scholar.

ther development of the SLRTool to address these proposals and the limitations and requirements discussed earlier.

The model driven approach we have deployed in the tool may introduce the possibility for new insights to be generated from the act of performing the SLR process. For example, implicit relationships between classification frameworks and quality criteria may become more apparent. In the case study used to illustrate this paper, we observed that because of the explicit SLR processes that are supported by the tool. Conversely, analysis that exposes inconsistencies both in the primary data and in the data arising from the analysis undergone as part of the SLR process may also be possible. As we continue to work on the development of the tool, we anticipate that new syntheses of research literature will reveal additional meta-properties that could guide the production of more informed research output.

We note that one area of concern remains regardless of any tool implementation. There is a policy issue at stake. For a tool like this to be truly embedded in the research practice of academics and students (at all levels), the tool will need to be part of the armoury of utilities resourced by managers of institutional libraries. Ease of use of the tool, particularly access to full text resources, is dependent on permissions available at the home library. Often, access to digital resources is determined by various identity management systems such as Athens² and Shibboleth³ in the UK. Such systems will need to be integrated the SLRTool tool for embedding to take place.

²<http://www.eduserv.org.uk/identity-access>.

³<http://shibboleth.net>.

ACKNOWLEDGEMENTS

We thank the UK's Joint Information Systems Committee (JISC) which supported this research at Middlesex University.

REFERENCES

- Babar, M. and Zhang, H. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 346–355. IEEE Computer Society.
- Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, 50(9):860–878.
- Bowes, D., Hall, T., and Beecham, S. (2012). Slurp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In *Proceedings of the 2nd international workshop on Evidential assessment of software technologies*, pages 33–36. ACM.
- Brereton, P., Kitchenham, B., Budgen, D., Turner, M., and Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571–583.
- Felizardo, K., Nakawaga, E., Feitosa, D., Minghim, R., and Maldonado, J. (2010). An approach based on visual text mining to support categorization and classification in the systematic mapping. In *Proc. of EASE*, volume 10, pages 1–10.
- Fernandez-Saez, A. M., Bocco, M. G., Romero, F. P., and Romero, F. P. (2010). Slr-tool - a tool for performing systematic literature reviews. In *ICSOFT (2)*, pages 157–166.
- Hauge, Ø., Ayala, C., and Conradi, R. (2010). Adoption of open source software in software-intensive organizations—a systematic literature review. *Information and Software Technology*, 52(11):1133–1154.
- Hernandes, E., Zamboni, A., Fabbri, S., and Di Thommazo, A. (2012). Using GQM and TAM to evaluate StArt—a tool that supports systematic review. *CLEI Electronic Journal*, 15(1).
- Hossain, E., Babar, M. A., and Paik, H.-y. (2009). Using scrum in global software development: a systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pages 175–184. Ieee.
- Jorgensen, M. and Shepperd, M. (2007). A systematic review of software development cost estimation studies. *Software Engineering, IEEE Transactions on*, 33(1):33–53.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:2004.
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering. a systematic literature review. *Information and Software Technology*, 51(1):7 – 15.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Riaz, M., Sulayman, M., Salleh, N., and Mendes, E. (2010). Experiences conducting systematic reviews from novices perspective. In *Proc. of EASE*, volume 10, pages 1–10.
- Staples, M. and Niazi, M. (2007). Experiences using systematic review guidelines. *Journal of Systems and Software*, 80(9):1425–1437.
- Stelmaszewska, H., Wong, B., Bhimani, N., and Barn, B. (2010). User behaviour: searching for scholarly material using electronic resource discovery systems. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, pages 17–26. British Computer Society.
- Walia, G. and Carver, J. (2009). A systematic literature review to identify and classify software requirement errors. *Information and Software Technology*, 51(7):1087–1109.