Tool support for systematic reviews in software engineering

by

CHRISTOPHER MARSHALL

A thesis submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

KEELE UNIVERSITY

June 2016

Abstract

Background: Systematic reviews have become an established methodology in software engineering. However, they are labour intensive, error prone and time consuming. These and other challenges have led to the development of tools to support the process. However, there is limited evidence about their usefulness.

Aim: To investigate the usefulness of tools to support systematic reviews in software engineering and develop an evaluation framework for an overall support tool.

Method: A literature review, taking the form of a mapping study, was undertaken to identify and classify tools supporting systematic reviews in software engineering. Motivated by its results, a feature analysis was performed to independently compare and evaluate a selection of tools which aimed to support the whole systematic review process. An initial version of an evaluation framework was developed to carry out the feature analysis and later refined based on its results. To obtain a deeper understanding of the technology, a survey was undertaken to explore systematic review tools in other domains. Semi-structured interviews with researchers in healthcare and social science were carried out. Quantitative and qualitative data was collected, analysed and used to further refine the framework.

Results: The literature review showed an encouraging growth of tools to support systematic reviews in software engineering, although many had received limited evaluation. The feature analysis provided new insight into the usefulness of tools, determined the strongest and weakest candidate and established the feasibility of an evaluation framework. The survey provided knowledge about tools used in other domains, which helped further refine the framework.

Conclusions: Tools to support systematic reviews in software engineering are still immature. Their potential, however, remains high and it is anticipated that the need for tools within the community will increase. The evaluation framework presented aims to support the future development, assessment and selection of appropriate tools.

Acknowledgements

The support of my supervisors, Pearl Brereton and Barbara Kitchenham, must first be acknowledged. Pearl, thank you for your incredible support over the last three years. I have had many great experiences throughout my time at Keele University. Meeting and working with you, however, has been one of the very best. Likewise, to Barbara, thank you for all of your comments on work I've sent you over the years. I have learned so much from both of you and I am incredibly grateful for your supervision.

I am also grateful for the financial support provided by the Faculty of Natural Sciences Research Office and Deen Support Fund. This funding enabled me to concentrate on my research and present at several stimulating conferences. Thanks are also owed to those working in the School of Computing and Mathematics. Thank you to all of the technical, administrative and teaching staff who have helped me along the way. I must also express thanks to all those who took part in this research. Thank you to each participant for your time and attention.

Thanks are also due to past and present PhD students at Keele. To Louis Major (now of the University of Cambridge), thank you for 'showing me the ropes' during the first few weeks as a research student, and your continued advice and friendship to this day. Likewise, to Adam Wootton. Adam, good luck with the remainder of your PhD. You've been a fantastic office-mate and I'll miss our Friday KPA lunches (but I won't miss losing to you at chess!).

Thanks are owed to members of my family. To my Mum (Janet) and Dad (Royce), thank you for all of your love, support and encouragement – you are fantastic parents. Also, to my Grandad (Vernon), thank you for the final push I needed to pursue this PhD.

And finally, to my fiancée Gillian Barrie. Gill, I love you. You have helped me in more ways than you'll ever know and I am so fortunate to have you in my life. To Stephen and Joyce (and other members of the Barrie family) thank you for everything you have done for us.

Author's Declaration

During the PhD, work reported in this thesis was presented at a number of conferences. Details of papers that were prepared for publication, along with conference and seminar activity, are presented in this section.

Publications

Marshall, C., Brereton, P., & Kitchenham, B. (2015). Tools to Support Systematic Reviews in Software Engineering: A Cross-Domain Survey using Semi-Structured Interviews. In *19th International Conference on Evaluation and Assessment in Software Engineering EASE 2015* (pp. 23-26). Nanjing, China.

Marshall, C., & Brereton, P. (2015). Systematic Review Toolbox: A Catalogue of Tools to Support Systematic Reviews. In *19th International Conference on Evaluation and Assessment in Software Engineering EASE 2015* (pp. 26-31). Nanjing, China.

Marshall, C., Brereton, P., & Kitchenham, B. (2014). Tools to Support Systematic Reviews in Software Engineering: A Feature Analysis. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering EASE 2014* (pp. 139-148). London, UK.

Marshall, C., & Brereton, P. (2013). Tools to Support Systematic Literature Reviews in Software Engineering: Protocol for a Feature Analysis. In *Proceedings of the 8th Psychology of Programming Interest Group Work-In-Progress Workshop PPIG-WIP 2013* (pp. 37-38). Keele University, UK.

Marshall, C., & Brereton, P. (2013). Tools to Support Systematic Literature Reviews in Software Engineering: A Mapping Study. In *Proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement ESEM 2013* (pp. 296-299). Baltimore, Maryland, USA.

Conference and Seminar Activity

External talks

- 19th International Conference on Evaluation and Assessment in Software Engineering (EASE 2015), Nanjing, China, April 2015.
- School of Health and Related Research (ScHARR) Seminar Series, University of Sheffield, UK, April 2015.
- 18th International Conference on Evaluation and Assessment in Software Engineering (EASE 2014), London, UK, May 2014.
- 7th International Symposium on Empirical Software Engineering and Measurement (ESEM 2013), Baltimore, Maryland, USA, October 2013.

Internal talks

- 5th Computing Postgraduate Research Day, Keele University, April 2015.
- 4th Computing Postgraduate Research Day, Keele University, April 2014.
- 8th Psychology of Programming Interest Group Work-in-Progress Workshop (PPIG-WIP 2013), Keele University, July 2013.
- 3rd Computing Postgraduate Research Day, Keele University, January 2013.

Contents

List of Tables

List of Figures

Chapter One

Introduction

This chapter introduces the main focus of the thesis; specifically, an empirical investigation into tool support for the systematic review methodology in software engineering. An introduction to evidence-based software engineering and the systematic review methodology are provided. The research questions examined throughout are outlined and the motivation and objectives for the work explained. The novelty of the thesis, and how it contributes to knowledge, is also identified. Finally, the structure of the thesis is presented.

1.1 Background

1.1.1 An introduction to the systematic review methodology

A thorough literature review forms the basis of a research project. Many researchers will perform this activity to identify related and relevant research within a particular field. In addition, a literature review can help establish areas within a field where new research can be undertaken.

The problem with a conventional literature review, however, is that the process is rarely underpinned by any clear and systematic procedures to ensure that all relevant literature is surveyed in an objective manner (Budgen & Brereton, 2006). Due to this, a conventional review often fails to provide any real scientific value (Mulrow, 1994; Cook *et al*., 1997; Kitchenham & Charters 2007). Its lack of rigour can bias the results and cause the researcher to miss important relevant literature. Furthermore, the methodology for identifying relevant research is rarely described in much detail and there is usually a very limited objective assessment of individual study validity (Haddaway & Pullin, 2014). Systematic reviews, however, aim to provide a means of carrying out literature reviews that are thorough and unbiased.

A systematic review is a formal, repeatable method for identifying, evaluating and interpreting all available research regarding a particular problem or topic of interest. The rigorous and impartial nature of a systematic review makes its findings of higher scientific value (Mulrow, 1994; Cook *et al*., 1997; Kitchenham & Charters, 2007) and an important tool for obtaining and appraising evidence in a reliable, transparent and objective way (Haddaway & Pullin. 2014). One of the key differing characteristics between a conventional literature review and a systematic review is the level of required planning. Prior to undertaking a systematic review, the researcher must develop a detailed protocol that documents the research questions, search strategy, study selection criteria, quality assessment method, data extraction strategy and data synthesis strategy (see Section 1.1.3).

Systematic reviews were first established in Clinical Medicine (Sacket *et al*., 1996; Higgins, 2008). Medical researchers defined the systematic review process to help mitigate the drawbacks of a

conventional literature review (Mulrow, 1994). In response to Archie Cochrane's call for more systematic reviews to assess the results of medical randomised controlled trials, the Cochrane Collaboration was founded in 1993 and became the first formal body to establish guidelines for the conduct of a systematic review (Allen & Richmond, 2011). Since then, researchers and practitioners in Clinical Medicine have long relied on systematic reviews, as a way of integrating and critically evaluating current knowledge to support decisions (Grimshaw & Russell, 1993; Hearn & Higginson, 1998; Karunananthan *et al*., 2009). Seeking the same benefits, many other domains began to adopt the systematic review process.

In 1999, the Campbell Collaboration was subsequently established to facilitate the production of systematic reviews relating to social interventions; including fields such as, education, criminology, social welfare and international development. Similarly, in 2003, the Centre for Evidence-Based Conservation was established in order to support decision making in conservation and environmental management. In 2004, Kitchenham *et al*. introduced the concept of Evidence-Based Software Engineering (EBSE) as an approach to integrate academic research with industry and improve decision making regarding the development and maintenance of software (Kitchenham *et al*., 2004).

1.1.2 Evidence-Based Software Engineering (EBSE)

EBSE aims "to provide a means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software" (Kitchenham *et al*., 2004).

Essentially, EBSE aims to help bridge the gap between research and practice (Dybå *et al*., 2005). Similar to initiatives in other domains, EBSE provides a process for solving practical problems based on a rigorous research approach, which in part, involves mapping and aggregating evidence. The EBSE process can be structured as five steps (Kitchenham *et al*., 2004):

1. Converting the need for information into an answerable question.
2. Finding the best evidence with which to answer the question.
3. Critically appraising the evidence for its validity, impact and applicability.
4. Integrating the critical appraisal with software engineering expertise.
5. Evaluating the effectiveness and efficiency in the previous steps (1 – 4) and seek ways to improve them.

The first three steps in this process are achieved by undertaking a systematic review. Since EBSE was defined in 2004, there has been a wealth of contributions from software engineering researchers, many of whom have employed the systematic review methodology as an integral part of their work (Kitchenham *et al*., 2009). The first version of guidelines for performing systematic reviews in software engineering were established in 2004 (Kitchenham, 2004) and updated in 2007 (Kitchenham & Charters, 2007). A further update is due for release in 2015.

With a growing emphasis on empirical software engineering research, the popularity and importance of systematic reviews has grown considerably (da Silva, 2011; Kitchenham & Brereton, 2013). Many software engineering researchers have performed systematic reviews to better understand the suitability and effectiveness of various tools and techniques (Hossain et al., 2009; Aleti *et al*., 2013; Radjenovic *et al*., 2013). Furthermore, systematic reviews have proven a useful starting point for research students (Riaz *et al*., 2010; Carver *et al*., 2013; Santos & da Silva, 2013). Students can undertake a systematic review to gain an understanding of a field and identify new research opportunities. Moreover, it has been identified that many of the motivations for performing systematic reviews have become academically rather than industrially driven (Santos & da Silva, 2013). This finding supports and justifies the increasing acceptance of the method within the software engineering research community.

1.1.3 The systematic review process

Systematic reviews follow a predefined strategy. They begin with the creation of a comprehensive protocol, detailing the nature and intended execution of the review, followed by the actual review process (that is comprised of several stages), and ending with the aggregation and documentation of results (Kitchenham & Charters, 2007).

A mapping study is a more 'open' form of systematic review (Budgen *et al*., 2008). A mapping study can provide an overview of a research area by assessing the quantity of evidence that exists on a particular topic (Peterson *et al*., 2015). A mapping study is commonly conducted as a preliminary activity with the intention to later undertake a full systematic review or empirical study into the topic of interest (Riaz *et al*., 2010). A more detailed description of mapping studies and how they differ from systematic reviews is provided in Section 2.1.

Undertaking a systematic review comprises several discrete stages that can be grouped into three core phases: planning, conducting the review and reporting the review. Figure 1-1 illustrates the 10 stage process for undertaking a systematic review in software engineering.

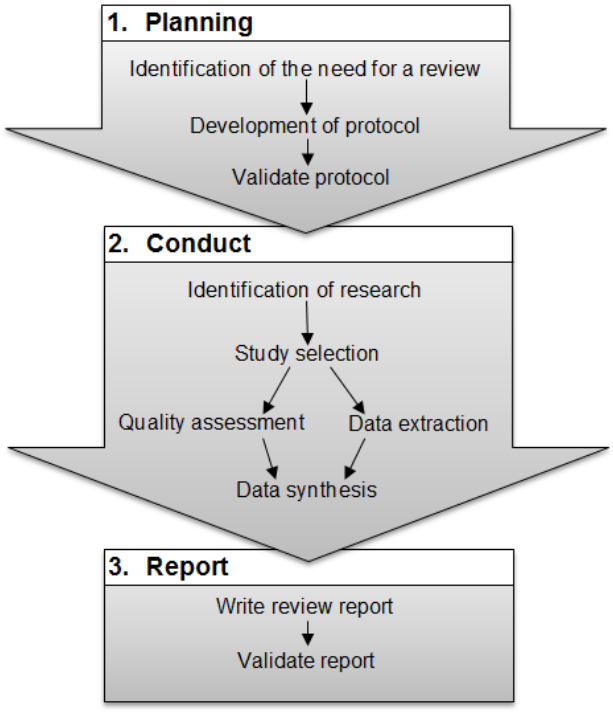


Figure 1-1. The 10 stage systematic review process

Phase 1: Plan review

The planning phase addresses the activity of designing how the study is to be performed. This phase includes three stages.

- *Stage One* is to define a set of research questions. The research question provides a framework for the formulation of search strings, the determination of data to be extracted and how data will be aggregated (Brereton *et al*., 2007).
- *Stage Two* entails the development of a review protocol. The protocol will act as a detailed plan that defines the process to be followed. This includes defining the conditions to apply when selecting primary studies, the instrument to be used when assessing the quality of included studies, and the allocation of reviewers to particular review activities.
- *Stage Three* is the validation of the review protocol. This may include piloting the data extraction strategy on a small sample of papers, or perhaps a formal review of the protocol by experienced reviewers. In any case, the protocol is of critical importance to the systematic review, and validation is necessary to ensure any revisions to the protocol can be made where appropriate.

Phase 2: Conduct review

Once an initial version of the protocol has been agreed amongst the review team, conducting the systematic review can commence. It should be noted that the protocol is not ‘cast in stone’ and may be revised throughout the review process. This phase comprises five activities:

- *Stage Four* is to identify relevant research using the search strategy defined in the protocol. The search will often take the form of an automated search of digital libraries and a manual search of conference and journal proceedings. The ultimate aim is to find as many primary studies relating to the research question as possible (Kitchenham & Charters, 2007).
- *Stage Five* involves the selection of primary studies. This will usually follow a two-stage process. First the title and the abstract of identified studies from the initial search are

reviewed, and any clearly irrelevant papers discarded. At this point the remaining papers included will be analysed more closely (i.e. papers are read in full) against the inclusion/exclusion criteria defined in the protocol. This multi-stage process is particularly necessary when undertaking a systematic review in software engineering because the "standard of IT and software engineering abstracts is too poor to rely on when selecting primary studies" (Brereton *et al*., 2007). It is recommended that this process should be conducted by at least two researchers. This allows any disagreements over a paper's inclusion/exclusion to be discussed and resolved.

- *Stage Six* is quality assessment. Once a set of primary studies have been agreed, they may be individually assessed for their quality (although it is not always necessary for a mapping study) using an instrument defined in the protocol (Kitchenham & Charters, 2007). The quality assessment can be used to support different stages of the systematic review process; notably, the study selection and data synthesis activities (Kitchenham & Charters, 2007).
- Stage Seven involves the data extraction of an included paper. The objective here is to decide what information to extract in order to address the research questions and to design and employ a data extraction form to accurately record the information from the primary studies (Kitchenham & Charters, 2007). The forms should be trialled during the development of the review protocol in order to reduce the opportunity for bias (Brereton *et al*., 2007).
- The final activity (*Stage Eight*) when conducting a systematic review is data synthesis. This activity involves "collating and summarising the results of the included primary studies" (Kitchenham & Charters, 2007) in accordance with the research questions. The synthesis should be carried out in a suitable manner, as defined in the protocol. Tabulating the data is a popular and useful means to aggregate data (Miles *et al.*, 2014). However, "when data is tabulated it may not be clear whether the research questions of the review have been answered" (Brereton *et al*., 2007). Therefore, whilst using this approach, it

would be useful for the researcher to explain how the data actually answers the research questions directly (Brereton *et al*., 2007).

Phase 3: Report review

Following the execution of the systematic review, its conduct must be fully documented. There are two stages to the report phase:

- The structure and content of the final report (*Stage Nine*), is described in the guidelines (Kitchenham & Charters, 2007). It is suggested that the protocol; in particular, the rationale for the study and the methodology used, can be used as the basis for certain sections in the final report (Brereton *et al*., 2007).
- Finally, once documentation has been completed, the report should be validated (*Stage 10*) either by an independent reviewer or through peer review assessment.

1.1.4 Characteristics of systematic reviews in software engineering

As discussed in Section 1.1.1, the systematic review methodology has been adopted in many different domains. Although many of the stages of a systematic review are similar across disciplines, there are a number of differences (and with those differences, additional challenges), which are inherent to systematic reviews within software engineering. Kitchenham, Budgen and Brereton discuss some of the characteristics of software engineering, which influence the systematic review process (Kitchenham *et al*, 2015).

In one of software engineering's seminal papers, Brooks Jr outlines some of the challenges faced by researchers when conducting primary research (Brooks Jr, 1987). The quality of a secondary study, such as a systematic review, relies heavily on the quality of results from primary studies. Therefore, it is important to consider these characteristics (or challenges) from a reviewer's perspective, and how they influence the systematic review process. For example, primary studies in software engineering usually involve active participation by its participants. Unlike other domains

such as clinical medicine; in software engineering, participants (or subjects) often take part in an active task (e.g. programming, reviewing, classifying) rather than simply receiving some form of treatment (Kitchenham *et al*., 2015). Therefore, the findings of a primary study may be influenced by the characteristics of its participants, such as their skills and previous experience. This particular issue can increase the difficulty of the synthesis stage of a systematic review. Furthermore, the terminology used in software engineering is often imprecise and inconsistent (Brereton *et al*., 2007; Kitchenham *et al*., 2015). This can further complicate a systematic review's search process, since all possible terminology needs to be considered when developing the search strategy. Primary studies in software engineering also lack statistical power (Kitchenham *et al*., 2015). This is because studies usually require specialist skills and knowledge, which makes participant recruitment difficult. Many studies in software engineering, therefore, fall short of the number of participants required to generate (what is generally regarded as) an acceptable level of statistical power (Dyba *et al.*, 2006). This again limits the strength of synthesis, which can be achieved in a systematic review and makes performing meta-analysis particularly challenging. Moreover, the reporting standards of many primary studies in software engineering are often poor (Brereton *et al*., 2007; Kitchenham *et al*., 2015). Kitchenham *et al.* state that many primary studies "still ignore the likelihood that a systematic reviewer will use the paper in the future". This can make study selection, quality assessment and data extraction activities more difficult.

Further issues inherent to systematic reviews undertaken in software engineering, as well as the more general difficulties associated with the method, are discussed in the next section.

1.1.5 Motivation for this research

Despite their usefulness and importance to the maturation of empirical software engineering research, undertaking a systematic review remains a highly manual, error prone and labour intensive process. In particular, there are challenges concerning the study selection, data extraction and data synthesis stages, amongst other collaborative activities (Brereton *et al*., 2007; Babar & Zhang, 2009; Riaz *et al*., 2010; Imitaz *et al*., 2013; Carver *et al*., 2013). Furthermore, the relative

recency of using systematic reviews within software engineering indicates issues surrounding the provision of appropriate support for novices (Riaz *et al*., 2010; Babar & Zhang, 2009; Imitaz *et al*., 2013; Carver *et al*., 2013). These drawbacks, along with others, make the systematic review methodology a prime candidate to benefit from automated tool support (Staples & Niazi, 2007; Riaz *et al*., 2010; Ramampiaro *et al*., 2010; Imitaz *et al*., 2013; Carver *et al*., 2013).

Carver *et al*. identifies some of the primary areas of the systematic review process in need of automated support. Firstly, identifying relevant papers is a largely manual and labour intensive process. This process is made increasingly difficult when performing systematic reviews specifically within software engineering, as search facilities are not as advanced as those in other domains (Brereton *et al*., 2007; Carver *et al*., 2013). Researchers in software engineering will usually have to perform resource-dependent searches, which makes searching consistently across a range of electronic resources challenging. Although some advancement has been made, software engineering still lacks adequate tools to assist in the extraction and storage of relevant papers (Carver *et al*., 2013). Tool support is also lacking for collaborative systematic reviews (Ramampiaro *et al*., 2010; Bowes *et al*., 2012; Carver *et al*., 2013).

Collaboration is a key component of a successful systematic review and has an impact on many of its stages. For example, after identifying a relevant set of articles, it is recommended that multiple researchers extract data from each paper and compare their results. Furthermore, members of a review team working on a systematic review may be based in different geographical locations; thus, making the logistics and coordination of a team-based systematic review challenging. Currently, there are no tools that allow extracted data to be stored, updated and reused effectively (Cruzes *et al*., 2007; Carver *et al*., 2013). Systematic reviews are particularly useful for identifying new areas to pursue further research. In many cases, this can result in the commission of a new systematic review that investigates a similar or related topic. This might result, therefore, in a lot of extracted data identified by a previous systematic review now being relevant for inclusion in the new one. However, there is currently no central repository in place that stores such data. This

means reviewers usually have to fully repeat the extraction step for each new systematic review they undertake. A mechanism which supports this aspect of a systematic review would significantly reduce the time and effort involved for this stage. Furthermore, as research into the topic of a published systematic review progresses, its original findings can quickly become out-dated. A tool that allows systematic reviews to be preserved, maintained and updated over time could, potentially, be of great benefit to both researchers and practitioners.

A range of tools have been developed and used to assist systematic reviewers in software engineering and in other disciplines. These include basic productivity tools, such as word processors and spreadsheets, reference managers, statistics packages and purpose-built tools which target all (or most) of the stages of the review process. A number of studies have investigated the use of tools to support systematic reviewers. Within the healthcare domain, a survey of information systems to support or automate systematic review tasks found a wide range of tools, especially, relating to reviews of randomised controlled trials (Tsafnat *et al*., 2013). Tools discussed by Tsafnat *et al*. include the Cochrane Commission's Review Manager (*RevMan*)[1], federated search engines such as *Quick Clinical*[2], citation managers (such as *EndNote*[3] and *ProCite*[4]), the *Abstrackr*[5] system to support screening of abstracts and meta-analysis tools (which are "already in wide use"). Also, within healthcare, a comparative study of data extraction tools and approaches found, not surprisingly, that each type of tool had some benefits and some drawbacks. The authors of the study concluded that "specialized web-based software is well suited in most ways, but is associated with higher setup costs" (Elamin *et al*., 2009). A more focused cross-domain mapping study of visual data mining support for systematic reviews found that "most of the studies (16 out of 20 studies) have been conducted in the field of medicine" (Felizardo *et al*., 2012). The authors of the study reported that data extraction and data synthesis were the most likely stages of the systematic review process to be supported by visual data mining tools. Within the software engineering

[1] http://tech.cochrane.org/Revman
[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1550689/
[3] http://endnote.com/
[4] http://www.procite.com/
[5] http://abstrackr.cebm.brown.edu/account/login

domain, a mapping study of tools for systematic reviews (other than basic productivity tools, spreadsheets and reference managers) also found that a range of, predominantly, visualisation and text mining tools had been developed and used to support study selection, data extraction and data synthesis (Marshall & Brereton, 2013).

It is clear that tools have the potential to provide valuable support for many aspects of the systematic review process. Currently, the tool landscape is changing quite rapidly, with a growing number of tools, especially those targeting the software engineering domain, being developed, adapted and enhanced. Therefore, an in-depth investigation into the usefulness and development of such tools would provide a beneficial contribution to the research community.

1.2 Research Questions

The overall aim of this thesis is to investigate the usefulness and development of tools that provide support for the systematic review process in software engineering.

Two research questions were developed to direct the focus of this project:

RQ1 - Can tools provide useful support when undertaking a systematic review in software engineering?

RQ2 – What are the most important features of tools to support systematic reviews in software engineering?

This thesis presents a programme of work undertaken to develop, refine and validate an evaluation framework for an overall tool to support systematic reviews in software engineering.

1.3 Original Contributions

This thesis reports a novel investigation into the usefulness and development of systematic review tools in software engineering. The evaluation framework for an overall tool to support systematic reviews in software engineering (presented in this thesis) is the first of its kind and provides a valuable contribution to the topic. In particular, the current usefulness and future potential of systematic review tools has been determined, and the most important features for a systematic review tool in software engineering have been established. The remainder of this section provides more detail on how specific elements of the work have contributed to knowledge in this area.

As discussed in Chapter Two, the mapping study, performed in the early stages of the research, established the potential to investigate tool support for systematic reviews in software engineering. The mapping study (and supplementary literature review that followed) provides the foundation for the objectives reported in this thesis. In contrast to other work in this area, the mapping study is the first of its kind to investigate the topic from a broader perspective. This work was presented at the 7th International Symposium on Empirical Software Engineering and Measurement (Marshall & Brereton, 2013).

The mapping study established the need to evaluate a selection of tools that provide support for the overall systematic review process in software engineering. The work undertaken to address this need; namely, a feature analysis, investigated these types of tools for the first time in a novel manner. The feature analysis is the first independent evaluation of 'whole process' tools to support systematic reviews in software engineering. This study is reported in Chapter Three of this thesis. Its findings were reported at the 18th International Conference on Evaluation and Assessment in Software Engineering (Marshall *et al*., 2014).

The work reported in Chapter Two and Chapter Three are both examples of studies focusing on the identification and examination of tools to support systematic reviews. Whilst studies like these are useful, it can, however, be challenging for reviewers to easily locate tools to support the conduct of

their systematic reviews. In Chapter Four, the "*Systematic Review (SR) Toolbox*[6]" is presented. *SR Toolbox* is a web-based catalogue of tools that support systematic reviews, which aims to help reviewers find appropriate tools based on their needs. A short paper introducing this novel resource was presented at the 19th International Conference on Evaluation and Assessment in Software Engineering (Marshall & Brereton, 2015).

In Chapters Five and Six, a cross-domain survey, which explores the scope and practice of tool support in other disciplines, is reported. Its findings were reported at the 19th International Conference on Evaluation and Assessment in Software Engineering (Marshall *et al*., 2015). In this study, part of the aim was to identify what participants in other domains consider to be the most important characteristics (or features) of tools to support systematic reviews. These features and importance levels were compared with those forming part of an evaluation framework proposed for tools to support systematic reviews in software engineering. However, the systematic review methodology is employed in many different domains. Consequently, many of the stages of a systematic review, regardless of the field it is undertaken, are similar. Therefore, reviewers in other domains also suffer many of the challenges faced by software engineering reviewers and seek tools to support them (Cohen *et al*., 2010; Tsafnat *et al*., 2014; Elliott *et al*., 2014; O'Mara-Eves *et al*., 2015). Despite this project being grounded in software engineering, the contribution of this work (and, in particular, the work reported in Chapters Five and Six) is considered to generalise across domains. The influence and contribution to knowledge by this work, therefore, extends beyond the original scope of the project.

[6] http://systematicreviewtools.com

1.4 Thesis Outline

A short description of each chapter is now given. See Figure 1-2 for this information in a diagram.

In ***Chapter Two***, a mapping study, which identifies and classifies tools that can help to automate part or all of the systematic review process in software engineering and establishes the degree to which those tools have been evaluated, is reported. The systematic review methodology has been used together with a supplementary search and review of the literature to update the background information. Work reported establishes the potential to evaluate a selection of candidate tools that provide support for the overall systematic review process in software engineering.

Motivated by the results of the mapping study, ***Chapter Three*** presents a feature analysis. This study aimed to compare and, independently, evaluate a selection of candidate tools, which are intended to provide support to the whole systematic review process (or at least the majority of stages within the process) in software engineering. The work also investigated the feasibility of an evaluation framework for such tools. An initial framework including a set of features, weightings and scoring apparatus was developed to perform this activity. Details of the evaluation process are provided and the results of the study discussed. Implications for the evaluation framework are also given. The conclusions influenced the research that followed.

Chapter Four introduces a novel resource, which allows reviewers to identify appropriate tools to support their systematic reviews based on their particular needs. *Systematic Review Toolbox* is a web-based catalogue of tools that support systematic reviews. Details of the motivation and development of the resource are reported.

In ***Chapter Five***, the background and study design of an interview-based survey is presented. The rationale for and a discussion of the appropriateness of the design are discussed. An introduction to the study, which aimed to explore the scope and practice of tool support for systematic reviews in domains outside of software engineering, is provided. An overview of the structure and content of the survey, including the procedures used for the interviews and selection of participants is given.

In ***Chapter Six***, details of the execution and results of the survey, introduced in Chapter Five, are presented. Qualitative and quantitative data were collected through semi-structured interviews and were analysed using an inductive approach. 13 researchers with experience performing systematic reviews in healthcare and social science were interviewed. Data was collected about participant's opinions on systematic reviews, their experience with systematic review tools and the importance of tool features. The findings of the study and implications for the evaluation framework are discussed. In addition, limitations of the survey and lessons learned from using semi-structured interviews are reported.

In ***Chapter Seven,*** the findings generated from all of the research activities performed and reported in this thesis are brought together and discussed in relation to the original research questions and objectives. The most recent version of the evaluation framework is presented, validated and used in a final comparative evaluation.

Chapter Eight presents a summary and conclusions of the research undertaken. Recommendations on the use and development of tools, final thoughts on the evaluation framework and suggestions for future work, are provided.

Chapter Two

Literature Review

Details the mapping study and supplementary literature review

Chapter Three

Feature Analysis

A feature analysis to compare and evaluate a selection of candidate tools, which provide support for the overall systematic review process in software engineering, is presented.

Chapter Four

Systematic Review Toolbox

Details of a novel web-based resource, which aims to help reviewers to identify appropriate tools to support their systematic reviews based on their needs, are outlined.

Chapter Five

Cross-Domain Survey: Background and Study Design

The background and study design of an interview-based survey, undertaken to explore the scope and practice of systematic review tool support in domains outside of software engineering, is provided.

Chapter Six

Cross-Domain Survey: Results, Discussion and Conclusions

The results of the interview-based survey, introduced in Chapter Five, are presented and discussed

Chapter Seven

Discussion

Findings from all of this work are brought together and discussed. The latest version of the evaluation framework is presented and validated and used to perform another comparative evaluation.

Chapter Eight

Summary and Conclusions

A summary of the work and conclusions are provided. Recommendations and guidance for related future work are also presented.

Figure 1-2. Summary of thesis content.

Chapter Two

Literature Review

In Chapter One, the motivation for the research project was established. In particular, the need for an in-depth investigation into the usefulness and development of tool support for systematic reviews in software engineering is discussed. In this chapter a mapping study of the literature is reported. The aim of the study was to identify and classify tools that can help to automate part or all of the systematic review process in software engineering and establish the degree to which they had been evaluated. An automated search strategy, plus snowballing, was used to locate relevant papers. A set of known papers was used to validate the search string. After applying the inclusion/exclusion criteria, 14 papers were accepted into the final set. A variety of approaches and support tools developed to assist the conduct of a systematic review in software engineering were found. Eight of the papers presented text mining tools and six papers discussed the use of visualisation techniques. The systematic review stage most commonly targeted by tools was study selection. Only two papers reported an independent evaluation of the tool presented. The majority were evaluated through small experiments and examples of their use. Four papers did not include any evaluation of the tool. Two years after the completion of the mapping study, a supplementary review of the literature was performed. This was done to ensure all relevant information, published following the mapping study, had been located. Work reported in this chapter established the potential to further investigate tool support for the systematic review process in software engineering.

2.1 Introduction to the Mapping Study

As discussed in Section 1.1.2, systematic reviews are a critical component of evidence-based software engineering. They are a useful research process providing a rigorous method for the location and analysis of evidence relating to a particular topic. They follow a pre-defined strategy beginning with the creation of a comprehensive review protocol, which details the nature and intended execution of the review, followed by the actual review process (that is comprised of several stages), and ends with the aggregation and documentation of results (see Figure 1-1).

A mapping study is intended to "map out" the research that has been undertaken, rather than to answer a detailed research question (Budgen *et al*., 2008). Mapping studies are more concerned with the exploration of a topic of interest rather than a rigorous evaluation and analysis of the related literature. Although a mapping study and systematic review share commonalities in their process, they are different in terms of their goals and approaches to data analysis (Peterson et *al*., 2015). Whereas a full systematic review aims at synthesising empirical evidence, a mapping study is more concerned with structuring a research area (Peterson *et al*., 2015). Therefore, the stages of a mapping study (although similar to a systematic review) are often broader in context to adequately address the wider scope of such a study (Budgen *et al*., 2008). A deeper explanation of the differences between the two methods is given in the following section.

2.1.1 Differences between mapping studies and systematic reviews

Kitchenham *et al*. contrasts the differing characteristics of a mapping study and a systematic review. A number of differences, regarding the research question, search process, quality assessment and analysis of results, are highlighted (Kitchenham *et al*., 2010).

The research questions in a mapping study, for example, can be more general in scope. This is as opposed to a full systematic review, which aims to answer specific research questions about a particular topic. Furthermore, the search strategy can be less stringent for mapping studies compared with systematic reviews. Wohlin *et al*. suggests that when undertaking a mapping study,

having a good sample and representation of studies is more important than identifying a large number of articles (Wohlin *et al*., 2013). This point is further discussed (and agreed upon) by Peterson *et al*. in their updated guidelines for undertaking mapping studies in software engineering (Peterson *et al*., 2015). In a systematic review, quality assessment is an important stage to determine the rigour and relevance of a primary study. In mapping studies, however, quality assessment is not essential, but may still be useful to ensure that sufficient information is available for data extraction (Peterson *et al*., 2015). In mapping studies, the data extraction and synthesis stages have an emphasis on classification and categorisation of evidence, over a more rigorous style demonstrated in a conventional systematic review. Papers presenting proposals or examples are largely excluded in a systematic review, as they lack empirical evidence. However, when mapping a research area; particularly, an area in its infancy, these types of articles should be included, as they can be important for identifying research trends and topics being worked on (Peterson *et al*., 2015).

2.1.2 Related work

A number of studies have investigated tools to support the systematic review process. Within healthcare, work was undertaken to identify and compare a range of data extraction tools (Elamin *et al*., 2009). Furthermore, a similar study, although more focused, was undertaken to evaluate evidence about the use of 'visual data mining' as a tool to support the systematic review process (Felizardo *et al*., 2012).

Following the completion (and publication) of the mapping study, further work has been undertaken in this topic area by other researchers. As discussed in Section 1.1.5, Tsafnat *et al.* performed a literature review, which identified and described a range of tools that automate part of the systematic review process in healthcare (Tsafnat *et al*., 2014). In addition, O'Mara-Eves et al. have since conducted a review of text mining approaches to support the study selection stage of systematic reviews in healthcare and social science (O'Mara-Eves et al., 2015).

The work reported in this chapter was the first study to investigate all types of tools, and is not limited to a specific type of approach or stage. Furthermore, this study focuses on tools that support the conduct of systematic reviews within the software engineering domain.

The findings of the study reported in this chapter have been published in the proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement (Marshall & Brereton, 2013).

2.2 Method

This mapping study was based upon the guidelines proposed by Kitchenham and Charters (Kitchenham & Charters, 2007). The study follows the stages outlined in Section 1.1.3 and visualised in Figure 1-1. A protocol was developed and presented to members of the software engineering research group[1] for feedback. The review panel included three experienced researchers (Prof Pearl Brereton, Prof Barbara Kitchenham and Mr Steve Linkman) and one previous PhD student (Dr Louis Major). In this section, the research questions are presented and the conduct of the study is described.

2.2.1 Research questions

The aims of the study were to identify and classify tools that can help to automate part or all of the systematic review process within the software engineering domain and to determine the extent to which they have been evaluated. In particular, special-purpose tools which had been designed or adapted specifically for supporting systematic reviews, were investigated. General purpose tools, such as productivity tools, reference managers and statistics packages, were not considered.

Four research questions were created to address the aims of the study:

RQ1) What tools to support the systematic review process in software engineering have been reported?

[1] http://www.keele.ac.uk/scm/research/compsci/softwareengineering/

RQ2) Which stages of the systematic review process do the tools address?

RQ3) To what extent have the tools been evaluated?

RQ4) What evidence is there about the usefulness of the tools?

2.2.2 Search process

The search process used was an automated keyword search of electronic databases. A snowballing strategy (i.e. pursuing the references of included papers) was also employed to identify papers of relevance that were not located via the automated search. Three electronic resources were used:

- ACM Digital Library - *(http://dl.acm.org)*
- IEEE Xplore - *(http://ieeexplore.ieee.org)*
- Google Scholar – (*https://scholar.google.co.uk)*

These resources sufficiently covered the relevant conferences and journals from the known papers (see Section 2.2.2.1). The search start date was 2004. This was the year that Evidence-based Software Engineering was first defined by Kitchenham, Dybå and Jørgensen (Kitchenham *et al*., 2004). The end date for the search was the end of 2012.

The following search strings were used to retrieve relevant papers:

- (tool OR support OR approach OR supporting) AND ("systematic literature review" OR "systematic review" OR "systematic literature reviews" OR "systematic reviews" OR SLR OR SR) AND ("software engineering")
- (tool OR support OR approach OR supporting) AND ("systematic literature review" OR "systematic literature reviews" OR "systematic review" OR "systematic reviews" OR SLR OR SR) AND (automatic OR automated OR automation)
- (tool OR support OR approach OR supporting) AND ("mapping study" OR "systematic mapping study") AND ("software engineering") AND (automatic OR automated OR automation)

2.2.2.1 Validating the search

Following the advice reported by Kitchenham *et al*. the search process was assessed for completeness by "obtaining a large and varied set of known studies based either on personal or manual search of important sources (e.g. journals and specialist conferences)" (Kitchenham *et al*., 2012). A relevant set of known papers were obtained from concurrent research being undertaken to investigate the application of systematic reviews within software engineering (Kitchenham & Brereton, 2013). Trial searches were performed in an attempt to retrieve all of 11 known papers (see Appendix A1). All but one paper was identified using the automated search. However, the paper in question was referenced in a number of articles in the known set. Since it was intended to use snowballing as part of the search strategy, the overall approach was concluded as adequate.

2.2.3 Inclusion and exclusion criteria

Inclusion and exclusion criteria were established to ensure that only relevant literature was accepted into the mapping study:

Inclusion Criteria:

- The publication must report on a tool that supports a systematic review, mapping study or both within the software engineering domain
- The tool reported in the paper can support any stage of the systematic review/mapping study procedure.
- The paper can report on any stage of development of the tool (i.e. proposal, prototype, conduct etc.)

Exclusion Criteria:

- Papers that are not written in English.
- Abstracts and PowerPoint presentations.

The inclusion and exclusion criteria were applied in two stages. In the first stage, papers located by the initial search were assessed for inclusion based upon analysis of their title and abstract. This

stage was carried out by me (referred to as CM) only and papers that were clearly of no relevance were discarded. In the second stage, the remaining papers were assessed against the inclusion and exclusion criteria by both members of the review team (i.e. CM and the lead supervisor PB), using the full text.

2.2.4 Data extraction

In order to answer the research questions, the following data was extracted from each paper. *EndNote* was used to store the abstracts and bibliographic information.

- Abstract and bibliographic information
- Study type (e.g. experiment, case study or discussion paper)
- Aims and objectives
- Type of approach underlying the tool (e.g. visualisation, text mining etc.)
- Name and a short description of the reported tool
- The particular stage of the systematic review process that the tool has been developed to support (e.g. study selection, data extraction etc.)
- Benefits of the tool
- Overheads or costs associated with the tool
- Whether tool has been independently evaluated.

The data except for abstracts and bibliographic information was extracted independently by both CM and PB. Disagreements were resolved through discussion.

2.2.5 Quality assessment

It is acknowledged in Section 2.1.1 that quality assessment is not always a necessary activity for a mapping study. For this study, however, quality assessment was required in order to adequately address RQ4 (see Section 2.2.1).

Each of the papers that included an evaluation of some sort was assessed for its quality. The quality assessment procedure was performed in tandem with the data extraction. The quality instrument developed by Dybå and Dingsøyr was employed (Dybå & Dingsøyr, 2008) as it has been used successfully in past systematic reviews to assess the quality of a range of study types. This instrument specifies 11 criteria used to assess quality:

1. Is the paper based on research or is it a "lessons learned" report based on expert opinion?
2. Is there a clear statement of the aims of the research?
3. Is there an adequate description of the context in which the research was carried out?
4. Was the research design appropriate to address the aims of the research?
5. Was the recruitment strategy appropriate to the aims of the research?
6. Was there a control group with which to compare treatments?
7. Was the data collected in a way that addressed the research issue?
8. Was the data analysis sufficiently rigorous?
9. Has the relationship between researcher and participants been considered to an adequate degree?
10. Is there a clear statement of the findings?
11. Is the study of value for research and practice?

The answers to each question for each paper were recorded in a spreadsheet and assigned either a value of 1 ('Yes'), 0.5 ('Partly'), or 0 ('No'). To ensure the validity of the quality assessment, both CM and PB conducted the assessment independently. Scores were compared and disagreements were resolved through discussion.

2.3 Results

2.3.1 Search results, data extraction and quality assessment

After the initial stage of the study selection process (i.e. applying the inclusion/exclusion criteria to the titles and abstracts only), 21 papers were included. The full text of each of these was considered by both reviewers and 16 were judged relevant. Two of these papers were subsequently excluded during data extraction, bringing the final set to 14 papers (see Table 2-1). Each paper in the study (included or excluded) can be identified by their Paper ID. Where papers report an evaluation study, an additional identifier (Study ID) was used. As shown in Table 2-1, P05, P08, P10 and P16, do not have a corresponding Study ID. This is because there were no studies reported in these papers which included an empirical element. P04, however, presents findings from two studies and, therefore, has two Study ID's (see Row 4 of Table 2-1). For a full list of the papers excluded from the mapping study, see Appendix A2.

Row No.	Paper ID	Study ID	Title	Paper Ref.
1	P01	S01	A Visual Text Mining Approach for Systematic Reviews	Malheiros *et al*., 2007
2	P02	S02	An Approach Based on Visual Text Mining to Support Categorization and Classification in the Systematic Mapping	Felizardo *et al*., 2010
3	P03	S03	Using Visual Text Mining to Support the Study Selection Activity in Systematic Literature Reviews.	Felizardo *et al*., 2011
4	P04	S04 S05	A Visual Analysis Approach to Validate the Selection Review of Primary Studies in Systematic Reviews	Felizardo *et al*., 2012
5	P05	-	SLR-Tool – A Tool for Performing Systematic Literature Reviews	Fernández-Sáez *et al*., 2010
6	P07	S06	Using Context Distance Measurement to Analyze Results across Studies	Cruzes *et al*., 2007
7	P08	-	Automated Information Extraction from Empirical Software Engineering Literature: Is that Possible?	Cruzes *et al*., 2007
8	P09	S07	Automatic Results Identification in Software Engineering Papers. Is it Possible?	Torres *et al*., 2012
9	P10	-	SLuRp: A Tool to Help Large Complex Systematic Literature Reviews Deliver Valid and Rigorous Results	Bowes *et al*., 2012
10	P11	S08	Analysing the Use of Graphs to Represent the Results of Systematic Reviews in Software Engineering	Felizardo *et al*., 2011
11	P12	S09	Towards Evidence-Based Ontology for Supporting Systematic Literature Review	Sun *et al*., 2012
12	P14	S10	Linked Data approach for selection process automation in Systematic Reviews	Tomassetti *et al*., 2011
13	P15	S11	Using GQM and TAM to Evaluate StArt - A Tool that Supports Systematic Review	Hernandes *et al*., 2012
14	P16	-	A Federated Search Approach to Facilitate Systematic Literature Review in Software Engineering	Ghafari *et al*., 2012
Excluded Papers: *P06, P13, P17, P18, P19, P20 and P21 were excluded (see* Appendix A2*)*				

Table 2-1. Set of included papers

Generally, there was a good level of agreement between the two sets of extracted data. However, there was disagreement about study types. Following discussion, a consensus was reached about what constituted an example and what should be considered a small experiment. A study that involved applying a tool to elements of a published systematic review and then discussing the outcomes in relation to those of the published study was classified as an example. An experiment involving only a very small number of participants was considered a small experiment (and in practice the maximum number of participants in the small experiments is five). There was also some discussion about what constituted an independent evaluation. It was agreed that where no author of a paper reporting an evaluation study had been involved in developing the tool, the evaluation should be considered independent.

The average scores (across the two reviewers) for the 11 evaluation studies reported in the 10 papers that include an empirical element are presented in Table 2-2. The full results for quality assessment can be viewed in Appendix A3.

Paper ID	Study ID	Avg.
P01	S01	**0.77**
P02	S02	**0.9**
P03	S03	**0.77**
P04	S04	**0.59**
	S05	**0.81**
P07	S06	**0.86**
P09	S07	**0.8**
P11	S08	**0.9**
P12	S09	**0.68**
P14	S10	**0.85**
P15	S11	**0.6**

Table 2-2. Quality assessment results

2.3.2 A Summary of included papers

This section provides a summary of each included paper identified and included in the mapping study. 14 papers were included in the final set. A list of these papers is presented in Table 2-1.

P01 - *A Visual Text Mining Approach for Systematic Reviews*

The authors describe some of the key challenges of performing a systematic review. They affirm that special-purpose tools to assist with undertaking a systematic review are needed. A Visual Text Mining (VTM) approach to support study selection is presented. An example, which compares the proposed VTM approach against a manual approach, is reported. Based on the results, the authors suggest that VTM has the potential to support the systematic review process.

P02 - *An Approach based on Visual Text Mining to Support Categorisation and Classification in the Systematic Mapping*

In this paper, a VTM tool has been developed and used to support the data extraction and data synthesis stages of a mapping study. By using the tool, the authors suggest that the time and effort required to perform these two activities, will be reduced. An example, which compares the results of two systematic reviews (one performed manually and one performed using the tool), is reported. The results show a significant reduction of time and effort when using the tool compared with a manual approach.

P03 – *Using Visual Text Mining to Support the Study Selection Activity in a Systematic Literature Review*

The authors discuss the importance and maturity of systematic reviews in software engineering and highlight the difficulties associated with their undertaking. In this paper, a VTM tool to support the study selection stage of a systematic review, is proposed. The authors compare the performance between a manual and VTM-based approach to study selection. Results indicate that using the tool speeds up this activity.

P04 – *A Visual Analysis Approach to Validate the Selection Review of Primary Studies in Systematic Reviews*

The paper highlights the importance and acceptance of the systematic review methodology in software engineering and discusses some of the key drawbacks to the process. The authors propose an approach using a VTM tool to assist with study selection; specifically, an aspect of this activity called "selection review". Results show that using the tool helps speed up this aspect of the study selection stage.

P05 – *SLR-Tool – A Tool for Performing Systematic Literature Reviews*

The paper presents a new tool (*SLR-Tool*), which aims to support the majority of stages in a systematic review. The authors discuss the features and functionality of the tool and share their experiences (and the experiences of several PhD students) using the system to support a systematic review. The authors call for feedback on the tool from other researchers within the community.

P07 – *Using Context Distance Measurement to Analyse Results across Studies*

In this paper, an interactive visualisation approach to compare contextual information across studies is presented. The tool uses clustering algorithms to assist with the exploration of similarities and differences between empirical studies. The tool aims to provide support to, primarily, the data synthesis stage of a systematic review. The authors report an example of its application, which shows promising results when compared with a manual approach.

P08 – *Automated Information Extraction from Empirical Software Engineering Literature: Is that Possible?*

The authors begin by discussing the rapid growth of empirical software engineering literature and the importance of systematic reviews for extracting and analysing evidence. It is argued that using information extraction tools could provide support for data extraction and data synthesis within a systematic review. An example, where the authors apply an entity recognition tool (*Site Content*

Analyzer), is reported. Results show that applying the tool helped automatically group documents within a systematic review.

P09 – *Automatic Results Identification in Software Engineering Papers: Is it Possible?*

In this paper, an analysis of several text mining techniques; specifically, methods of sentence classification, is reported. The authors describe an example they performed, where each method is applied to a corpus of unstructured software engineering papers during the data extraction stage of a systematic review. The results showed that the methods were not effective. The authors call, however, for new tools to assist with data extraction.

P10 – *SLuRp: A Tool to Help Large Complex Systematic Literature Reviews Deliver Valid and Rigorous Results*

In this paper, the authors discuss the issues associated with undertaking a systematic review. A tool (*SLuRp*), which aims to provide support for each stage of the systematic review process in software engineering, is presented. The functionality of *SLuRp*, and how it provides support for each stage of a systematic review, is described. The author's experiences of using *SLuRp* to perform a systematic review are also reported. They claim using *SLuRp* speeds up a systematic review's undertaking and increases the confidence in its results.

P11 – *Analysing the use of Graphs to Represent the Results of Systematic Reviews in Software Engineering*

The authors discuss the traditional approaches for reporting and visualising data in a systematic review (e.g. tables). They propose the use of other novel forms of graphical representation, in an attempt to improve comprehension of results. In this paper, the authors investigate whether graphs provide better comprehensibility than tables when presenting results. An experiment, which compares the effectiveness of using tables against graphs when presenting the results of a systematic review, is described. Experts and students were asked to analyse and understand the results of a systematic review, presented in both formats (tables or graphs). Results showed that graphical visualisation of results led to a reduction in time for analysis and comprehension.

P12 – *Towards Evidence-Based Ontology for Supporting Systematic Literature Reviews*

In this paper, the authors discuss the importance and usefulness of systematic reviews and the key challenges of their undertaking. An evolved ontology (*SLRONT*), which supports the automation of study selection and data extraction in a systematic review, is presented. An example of its application is reported. Results showed a significant reduction in the time and effort involved when compared with a manual approach.

P14 – *Linked Data Approach for Selection Process Automation in Systematic Reviews*

The authors discuss the difficulties associated with undertaking a systematic review. In this paper, an approach to semi-automate the study selection stage in the process, is presented. An example, which describes the implementation of a text mining tool, is reported. Results from the example showed a significant reduction in workload when compared with performing the study selection activity manually.

P15 – *Using GQM and TAM to Evaluate StArt – A Tool that Supports Systematic Review*

The drawbacks of performing a systematic review are described by the authors. A tool (*StArt*), which aims to provide support for each stage of a systematic review in software engineering, is presented. The tool's features and functionality are described. A survey, which aims to characterise the tool's usefulness using the "Goal Question Metric" and "Technology Acceptance Model" (GQM and TAM), is reported. Results indicate that tools to support the systematic review process, like *StArt*, would be very useful to researchers.

P16 – *A Federated Search Approach to Facilitate Systematic Literature Reviews in Software Engineering*

The authors discuss the importance of systematic reviews and evidence-based research. A key issue surrounding the search process when performing a systematic review is described; specifically, that digital libraries in software engineering do not provide adequate support for systematic reviewers. A federated search tool, developed to support the search process when performing a systematic

review, is presented. The tool aims to provide an automatic integrated search mechanism, which maps to well-known software engineering databases. An example using the tool is reported. Preliminary results suggest that the tool reduces the time to perform the search and, in addition, improves the reliability of its results.

2.3.3 Tools and underlying approaches

This section addresses RQ1: **What tools to support the systematic review process in software engineering have been reported?**

Eight papers present tools based on text mining, which is the largest 'support tools' cluster. Three are based on Project Explorer (*PEx*[2]) and two on *ReVis*[3]. *PEx* is a flexible visualisation tool providing several text handling facilities. *ReVis* is another visualisation and interaction tool providing a framework for different projection techniques to construct mappings. The remaining support tools are each reported in a single paper. P08 presents an entity recognition tool (*Site Content Analyzer*[4]) to support automated information extraction from empirical software engineering literature. The *UNITEX*[5] tool is reported in P09 to assist automatic results identification in software engineering papers. P14 discusses *DBpedia*[6], a resource description framework repository to support automated selection of primary studies.

The second biggest cluster is of tools based on methods of visualisation. Six papers fall into this category with four of these being based on visual text mining (and using *PEx* or *ReVis*). The remaining papers concern the use of *Hierarchical Cluster Explorer* (*HCE*[7]), a tool to identify patterns in multi-dimensional data sets (P07), and the use of an extension of *PEx* (*PEx-Graph*[8]) to provide graphical representations of results (P11).

[2] http://infoserver.lcad.icmc.usp.br/infovis2/PEx
[3] http://ccsl.icmc.usp.br/pt-br/projects/revis
[4] http://www.cleverstat.com/en/sca-website-analysis-software-index.htm
[5] http://www-igm.univ-mlv.fr/~unitex/index.php?page=0
[6] http://dbpedia.org/About
[7] http://www.cs.umd.edu/hcil/hce/
[8] http://infoserver.lcad.icmc.usp.br/infovis2/PExGraph

Underlying approach	Paper ID	Total
Visualisation	P01; P02; P03; P04; P07; P11	6
Text mining	P01; P02; P03; P04; P05; P08; P09; P14	8
Visual text mining (VTM)	P01; P02; P03; P04	4
Tools that support the whole systematic review process	P05; P10; P15	3
Ontology	P12	1
Search tool	P16	1

Table 2-3. Underlying approaches

The third largest cluster refers to tools that support the whole systematic review process. This grouping refers to stand-alone applications that aim to assist all (or at least most of) the stages of the systematic review process. Three papers are in this category. They present *SLR-Tool*[9] – which incorporates the use of text mining techniques and is freely available (P05), *SLuRp*[10] – an open source web-enabled database (P10) and *StArt*[11] – which provides support for most of the stages of a systematic review with the exception of the searching process (P15).

The remaining two papers present a federated web-based search tool, developed using Python, a document based database called *MongoDB*[12] (P16) and *SLRONT*, an evidence-based ontology supporting systematic reviews (P12).

Table 2-3 shows a mapping between the paper/study and the reported underlying approach. Table 2-4 lists the support tools identified.

2.3.4 Stages addressed by the tools

This section addresses RQ2: **Which stages of the systematic review process do the tools address?** The results are summarised in Table 2-5.

[9] http://alarcosj.esi.uclm.es/SLRTool/
[10] https://codefeedback.cs.herts.ac.uk/SLuRp/
[11] http://lapes.dc.ufscar.br/tools/start_tool
[12] http://www.mongodb.org/

Support tool	Paper ID	Total
Project Explorer (PEx)	P01; P02; P11	3
ReVis	P03; P04	2
SLR-Tool	P05	1
Hierarchical Cluster Explorer (HCE)	P07	1
Site Content Analyzer	P08	1
UNITEX	`	1
SLuRp	P10	1
SLRONT	P12	1
StArt	P15	1
DBpedia	P14	1
Unnamed tool	P09; P14	2

Table 2-4. Support tools identified

Most of the papers (11 of 14) present tools that address the conduct phase of the systematic review process and three papers describe tools that aim to support the overall review process. One of the tools addressing the conduct phase also provides support for the reporting phase (P11). Of the papers addressing the conduct phase, the study selection stage is the most commonly targeted by the tools (five papers). Three papers present visual text mining (VTM) tools (using *PEx* and *ReVis*) to support this stage (P01, P03, P04). One paper (P12) discusses constructing ontology (*SLRONT*) and another (P14) presents a text mining approach (using *DBpedia*).

Systematic review phase	Systematic review stage	Paper ID	Total
Planning the review	Identification of the need for a review	-	-
	Development of protocol	-	-
Conducting the review	Identification of research	P16	1
	Study selection	P01; P03; P04; P12; P14	5
	Study quality assessment	-	-
	Data extraction	P02; P09; P12	3
	Data synthesis	P02; P07; P08; P11	4
Reporting the review		P11	1
Whole process		P05; P10; P15	3

Table 2-5. Systematic review stage targeted by tool

Data synthesis is the next most commonly targeted stage of the systematic review process (four papers). Two papers present visualisation tools (using *PEx* and *HCE*) to support this stage (P07, P11). One paper (P02) presents a VTM tool (*PEx*) and a final paper (P08) describes a text mining approach using an entity recognition tool (*Site Content Analyzer*).

Type of study	Study ID	Total
Small experiment	S01; S03; S04; S05; S07; S09	5
Experiment	S08	1
Example	S02; S04; S05; S06; S09; S10	6
Survey	S11	1

Table 2-6. Method of evaluation

Three papers describe tool support for the data extraction stage. One paper presents a VTM tool (*PEx*) to support this stage (P02). Another paper (P09) discusses a text mining approach (using *UNITEX* and two unnamed tools). A final paper (P12) discusses using an ontology-based tool (*SLRONT*) to assist this activity.

Three papers present details of tools that aim to support researchers throughout the systematic review process; namely, *SLR-Tool* (P05), *SLuRp* (P10) and *StArt* (P15). These tools are described in a later chapter (see Section 3.3.1).

2.3.5 Evaluation of the tools

This section addresses RQ3: **To what extent have the tools been evaluated?**

As indicated in Table 2-1, the 14 papers include 11 evaluation studies. A classification of the studies by method of evaluation is shown in Table 2-6. Most of the evaluation studies are either examples or small experiments, with examples (6) forming the largest cluster.

Examples compare the outcomes of using the tool with the results of a published systematic review. Five studies report a small experiment with a sample size between three and five. S08 is a full-scale

experiment with 24 participants. The participants had a range of levels of experience regarding research and systematic reviews, and most were PhD or Masters students. A survey (consisting of two questionnaires) was reported in S11. 14 students took part in the first evaluation, and 35 in the second. In both occurrences, the participants were Computer Science graduate students.

Benefits	Paper ID	Type of approach	Total
Faster/reduction of effort	P01; P02; P03; P04; P05; P07; P10; P11; P12; P14; P15	Whole process; Visualisation; Text mining; Ontology; VTM	11
Improves clarity	P01; P02; P03; P05; P10; P11	Whole process; Visualisation; Text mining; VTM	6
Improves accuracy	P01; P02; P03; P12; P16	Search tool; Ontology; VTM	5
Improves validity	P03; P10; P14	Whole process; Text mining; VTM	3
Improves collaboration	P10; P12	Whole process; Ontology	2
Improves organisation	P10; P12; P16	Whole process; Search tool; Ontology	3
Improves presentation	P02, P03, P11	Visualisation; VTM	3

Table 2-7. Benefits associated with reported tools

Most (9) of the evaluation studies were carried out by the tool developers or by researchers who had adapted or applied generic tools to support the proposed approach. Two studies report independent evaluations of the tool (S07, S11).

2.3.6 Usefulness of the tools

This section addresses RQ4: **What evidence is there about the usefulness of the tools?**

Table 2-7 summarises the benefits associated with the tools expected by authors or participants of empirical studies and Table 2-8 the costs (or overheads).

Most of the tools presented were, at the time of review, in the early stages of development and usage. This led to limited primary data about the usefulness of tools, as most papers provide only examples of use, small experiments or no evaluation (P11, P15). Two of the papers do include more substantial evaluation studies. The results of the experiment reported in P11, which has one of the highest quality scores (see Table 2-2), suggests that *PEx* is an effective tool for presenting the results of systematic reviews and reduces the time taken for their analysis. The survey reported in P15, which has a relatively low quality score, suggests that *StArt* is a "useful" tool to support the systematic review process. This conclusion is based on the opinions of 49 graduate students who responded to questions about the tool.

Costs (Overheads)	Paper ID	Type of approach	Total
Compatibility issues	P01; P03; P04; P05; P14; P16	Whole process; Text mining; Search tool; VTM;	6
Setup time	P01; P03; P04; P12	Ontology; VTM	4
Training required	P01; P02; P03	VTM	3

Table 2-8. Costs (or overheads) associated with reported tools

These results suggest that to-date there is very little evidence about the usefulness of the tools described. As a consequence of this, data was extracted about what benefits might be expected from the tools (by the authors or participants of studies) and what overheads (or costs) might be associated with them.

Seven types of benefit were reported. The most common is *faster/reduction in effort*, which is reported in 11 papers (and refers to visualisation, text mining, ontology, VTM and tools to support the whole systematic review process). Six papers present tools which report to *improve clarity*, the second most common benefit. The third most commonly reported benefit is *improves accuracy/precision* (five papers). Three papers report that the tool *improves validity* through "improve rigour" or "reduce bias" (P03, P10, P14). Three papers indicate that the tool described

improves presentation of the results of a systematic review (and hence their understandability). Two papers suggest that the tool *improves collaboration* (P10, P12).

Overheads (or costs) associated with the tools include compatibility issues (six papers), setup times (four papers) and training requirements (three papers).

2.3.7 Limitations of the mapping study

There are two main threats to the validity of this study:

- The possibility that relevant papers were missed during the identification of research.
- Bias in the process of agreeing data values and quality assessment scores.

An automated search of three electronic resources was carried out. Due to a low number of resources and the absence of any manual search, there is the possibility that not all relevant papers were located. However, as reported by Kitchenham *et al*., the search requirements are less stringent for a mapping study than for a full systematic review (Kitchenham *et al*., 2010). Furthermore, a large set of known papers (see Appendix A1) was used to validate the search strings, as recommended by Kitchenham *et al*. (Kitchenham *et al*., 2012). In addition, the search strategy successfully identified all relevant papers found by the broader mapping study undertaken by Felizardo *et al*. (Felizardo *et al*., 2012). A supplementary literature update has also been undertaken to ensure all relevant literature has been considered between the completion of the mapping study and the submission of this thesis (see Section 2.4).

As discussed in Section 2.3.1, there were a number of differences in the data extracted and the quality assessment scores between CM and PB. Although the disagreements were discussed until a joint set was agreed, the fact that one of the researchers is a PhD student and the other their supervisor might have influenced the outcomes. Every effort was made to avoid this by including comments with the scores where it was considered useful to justify these.

2.4 Supplementary Literature Update

The mapping study includes studies published from 2004 to 2012. To ensure all relevant work (published between January 2013 and June 2015) is considered in this thesis, an additional search of the literature was undertaken. Whilst the search strategy outlined in Section 2.2 was not fully repeated, previous experience gained when conducting the original mapping study influenced the process for this supplementary search. The same search terms and electronic resources were used. This is in addition to the snowballing technique being adopted when analysing retrieved literature.

As shown in Table 2-9, five additional papers were found:

Title	Paper Ref.
SESRA: A Web-based Automated Tool to Support the SLR Process	Molléri & Benitti, 2015
Semi-automatic Selection of Primary Studies in Systematic Literature Reviews: Is it Reasonable?	Octaviano *et al*., 2014
Towards Supporting Systematic Mapping Studies: An Automatic Snowballing Approach	Bezerra *et al*., 2014
A Visual Analysis Approach to Update Systematic Reviews	Felizardo *et al*., 2014
Automatically Locating Results to Support Systematic Reviews	Torres *et al*., 2013

Table 2-9. Additional papers identified from the supplementary search

As was the case during the mapping study, most papers identified during the supplementary search reported visualisation and text mining approaches to support the systematic review process in software engineering. Torres, Cruzes and Salvador define a new text mining method (*Textum*) to automate the task of locating results in unstructured software engineering papers (Torres *et al*., 2013). The work reported in this paper builds upon the authors' earlier research, which was identified and included in the mapping study (see Table 2-1: P07, P08 and P09). Octaviano, Felizardo, Maldonado and Fabbri present another text mining approach. Their technique combines two previously identified tools (*StArt* and *ReVis*) to form a "Score Citation Automatic Selection" (SCAS) strategy (Octaviano *et al*., 2014). Similar to the work reported by Torres *et al*., 2013, this

work builds on previous research undertaken by the authors (see Table 2-1: P03, P04 and P15). Felizardo, Nakagawa, MacDonnell and Maldonado present a visualisation and text mining (VTM) approach. The technique (labelled "USR-VTM") is implemented using *ReVis* (a visualisation tool) and aims to support updates of existing systematic reviews (Felizardo *et al*., 2014). The concept of VTM to support systematic reviews and, in addition, the use and development of *ReVis*, has been investigated in previous work undertaken by the authors (see Table 2-1: P01, P03 and P04). Bezzerra, Favacho, Souza and de Souza present an automatic snowballing approach to support mapping studies. This approach is a particular feature of a more substantial tool (*Ramani*) developed by the authors. *Ramani* is described as a web-based collaborative application for supporting mapping study projects. Lastly, Molléri and Benitti introduce *SESRA*[13], a web-based tool that aims to support all phases of a systematic review (Molléri & Benitti, 2015).

The majority of papers found by the supplementary search reported on tools/approaches which aimed to support activities within the conduct phase of a systematic review (see Figure 1-1). The text mining approach presented by Torres *et al*. (*Textum*) aims to support data extraction and data synthesis. The SCAS strategy reported by Octaviano *et al*. aims to assist study selection. The additional VTM approach proposed by Felizardo *et al*. aims to provide support for the search process and study selection stages of a systematic review; specifically, when updating a previous review. The automated snowballing technique presented by Bezerra *et al*. aims to support the search and selection of relevant papers; particularly within mapping studies. Unlike the majority of tools identified, however, which only target support for one particular stage (or activity), *SESRA* (introduced by Molléri and Benitti) targets support for all phases of the systematic review process in software engineering. *SESRA* joins three other previously identified tools; namely, *SLuRp*, *StArt* and *SLR-Tool* (see Table 2-4), which all offer similar support.

As with the majority of the papers identified in the mapping study, the papers found during the supplementary search reported limited evaluations of the tools (or approaches) they presented. In

[13] http://sesra.net/

particular, none of the tools have been independently evaluated. Torres *et al*. report a small feasibility study to assess the effectiveness of *Textum* compared with a previous approach they proposed in P09 (see Table 2-1). The results of *Textum* showed an improvement in precision and recall over the previous tool. Octaviano *et al*. report the results of an example using the SCAS strategy. In this example, the authors compare the accuracy and effort of performing study selection manually and using their proposed approach. Using the tool, results showed a significant reduction in effort required to perform the task; however, a loss in accuracy was reported. Felizardo *et al*. performed an experiment to evaluate their USR-VTM approach. A sample of 12 graduate students were recruited and split into two groups. Group One performed the selection activity using the traditional approach (i.e. manually) and Group Two performed the same activity using the tool. Results showed Group Two achieved higher performance than Group One. Bezerra *et al*. discuss an example using their automatic snowballing approach. The authors applied their approach to a selection of papers, which had been identified as part of a previously undertaken mapping study. The goal was for the tool to return these papers and, in addition, any relevant papers that had been missed or published after the completion of the original study. Results showed that using the tool helped identify a selection of new papers to include. Molléri and Benitti report two small-scale evaluation activities for their tool (*SESRA*). A GQM-based questionnaire was circulated to a small number of researchers (local to the authors) for feedback on the tool. In addition, the authors designed a series of tests to evaluate various features of *SESRA*. Results from both activities suggest that *SESRA* can improve the reliability and productivity of a team-based systematic review.

2.5 Discussion

The results of the mapping study and supplementary literature review provide a platform for the programme of research reported in this thesis. In this section, the implications of the findings of the literature review are discussed. To recap, the aims of the mapping study were:

- To identify and classify tools that can help to automate part or all of the systematic review (and mapping study) process in software engineering
- To establish the degree to which these tools have been evaluated.

Several types of tools were found. Text mining (eight papers) and visualisation (six papers) were the most common underlying approaches of tools, with four of these papers describing visual text mining tools. The predominance of visualisation and text mining tools is further reflected by the literature update reported in Section 2.4. During this supplementary review, four additional papers were found that that also described visualisation or text mining approaches.

The three tools to support the whole process all claim the benefit *improves collaboration.* Collaboration is an important aspect of conducting a systematic review and only one other tool (*SLRONT*) claims to provide support for this (P12). However, only one of the tools to support the whole process has been the subject of an evaluation study (S11), indicating that, at that time, there was very little evidence about their effectiveness. Furthermore, an additional tool (*SESRA*), which was identified during the supplementary review, also aims to support the whole systematic review process. However, as was the case with the majority of 'Whole Process' tools found in the mapping study, *SESRA* has yet to be subjected to a rigorous evaluation.

The most commonly reported benefit claimed by authors or as a result of an evaluation study is *faster/reduction of effort*. This benefit addresses two of the most widely reported problems associated with undertaking a systematic review, in that they are time consuming and labour intensive (Babar & Zhang, 2009). This is further reflected in the three stages most commonly

targeted by tools; namely, study selection, data extraction and data synthesis. These stages are considered some of the more demanding activities of the systematic review process, and hence are likely to benefit from automated support (Riaz *et al*., 2010). However, it is interesting to note that no dedicated tool to support the quality assessment stage was found. Quality assessment is an important stage of a systematic review and is also considered a difficult and time-consuming task (Zhou *et al*., 2015).

Only 19 relevant papers were found overall (14 in the original mapping study and 5 more in the supplementary review of the literature), the majority of which included only preliminary investigations, often describing an example of the tool in use, or a small experiment to assess its effectiveness. In addition, only two studies reported that an independent evaluation of a tool had been carried out. These results, therefore, reflect the immaturity of the research area and provide foundations for future empirical work.

2.6 Summary

The mapping study and supplementary review reported in this chapter explored what tools were available to help automate part or all of the systematic review process within software engineering, and established the degree to which they had been evaluated. The findings of the literature review aim to address RQ1, Objective 1 (see Section 1.2.1 and 1.2.2).

Three electronic resources were searched for research concerning tool support for systematic reviews in software engineering. Following the initial stage of the study selection process, 21 papers were included. This was reduced to 16 papers after each article was read in full. Upon closer analysis during data extraction, two more papers were excluded, resulting in a final set of 14 papers. 11 evaluation studies were reported in 10 of the included papers. A supplementary search for relevant literature published between January 2013 and June 2014 identified an additional five papers (see Section 2.4).

Results show a small but encouraging growth of tools to support the systematic review process in software engineering. A predominance of visualisation and text mining techniques, to support the study selection, data extraction and data synthesis stages in a systematic review, were found.

This chapter has identified how further research into tools, to support the systematic review process in software engineering, is required. The results of the mapping study and supplementary literature update have provided a platform for future work to be undertaken. Based upon the evidence gathered, most of the tools identified were in the early stages of development and usage. This has led to very little primary data regarding their usefulness and, generally, only speculation over their potential. In the remainder of this thesis, a programme of research to investigate the usefulness and development of systematic review tools is presented. This work involves a series of studies and activities to develop and validate an evaluation framework for an overall tool to support systematic reviews in software engineering.

Other literature that has significantly influenced this research project, including that related to systematic reviews (and tools) in other domains such as healthcare and social science (see Section 1.1.1, Section 1.1.5 and Chapter Four) and literature related to methods for evaluating tools (see Section 3.2), is described elsewhere in this thesis.

Chapter Three

Feature Analysis

The literature review identified that current tools to support systematic reviews in software engineering had received limited evaluation. In particular most tools had received no independent evaluation, with generally only speculation over their potential. This was particularly the case for a selection of tools that aimed to support the whole systematic review process. In this chapter, details of a multi-criteria decision analysis activity, which compares and evaluates a selection of overall systematic review support tools (*SLuRp*, *StArt*, *SLR-Tool* and *SLRTOOL*), is reported. This work also serves as a feasibility study of an evaluation framework to evaluate tools that support the whole systematic review process in software engineering. The framework comprises a set of features, weightings and scoring instruments. Results showed that each of the candidate tools presented some strengths and weaknesses. *SLuRp* scored highest, whilst *SLRTOOL* has the lowest overall score. *SLuRp* scored well on process management features such as support for multiple users and document management and less well on ease of installation. The results of the study provided new insight into tools that support systematic reviews in software engineering and led to a refined version of the evaluation framework. Work to continue this investigation and further refine and validate the framework is discussed.

3.1 Introduction

In Chapter Two, a variety of tools that provide support for systematic reviews in software engineering were identified in the literature review. This work found a predominance of visualisation and text mining techniques to support, primarily, the study selection, data extraction and data synthesis stages of a systematic review undertaken in software engineering. (Marshall & Brereton, 2013). The literature review, organised as a mapping study, also identified a selection of tools (three in total) that aimed to support all (or at least the majority of) stages in a systematic review. Whilst promising, however, there was little primary data regarding the effectiveness of these types of tools with, generally, only speculation over their potential. In particular, the majority of these tools had received no independent evaluation.

The study reported in this chapter aims to compare and evaluate these three 'whole process' tools together with a further candidate system (introduced in Section 3.3.1), which also aims to support the overall systematic review process in software engineering. The reported study is a multi-criteria decision analysis (MCDA) activity, which takes the form of a feature analysis. It is the first step toward the development of a rigorous evaluation framework for tools that support systematic reviews in software engineering. A set of initial features, which such tools should include, is generated and each tool is scored against each feature. The strengths and weaknesses of each tool, in terms of how well it provides support for each feature, are discussed.

This chapter is organised as follows. Section 3.2 provides an overview of software evaluation, MCDA, the DESMET methodology and the feature analysis approach. Section 3.3 introduces the candidate tools and describes the set of features used as the basis for the evaluation, the scoring process and the method used to calculate an overall score for each tool. Section 3.4 presents the results of the feature analysis. This is followed by a discussion of the study in Section 3.5, including refinements made to version 1.0 of the evaluation framework in Section 3.5.2. Section 3.6 provides a summary of this work and implications for the later work reported in this thesis. The findings of this feature analysis have been reported as a conference paper (Marshall *et al*., 2014).

3.2 Method

This section describes the approach taken to evaluate the four candidate tools, which uses a form of multiple criteria decision analysis (MCDA). An overview of software evaluation and MCDA methods is followed by a description of the DESMET methodology and feature analysis approach.

3.2.1 Evaluating software

Software evaluation is the problem of determining the extent to which a piece of software satisfies a set of requirements (Dujmović & Kadaster, 2002). The increasing size, complexity and demand for software are some of the most important issues within the software engineering domain (Ali Babar *et al.,* 2004). To achieve software quality, it is important to establish the key required features for the software and perform an evaluation as early as possible (Ali Babar *et al.*, 2004). Software quality is defined as the degree to which the software includes a desired and effective combination of different features (Ali Babar *et al.*, 2004). Brown and Wallnau (1996) define two main types of technology evaluation:

1. *Product-oriented* – focuses the evaluation on selecting the best product (i.e. software) from a range of products offering similar functionality.
2. *Process-oriented* – an evaluation to assess the impact of a new technology on existing practices and to understand how it will improve performance and/or increase quality.

Software evaluations can be formulated as a multiple criteria decision making (MCDM) problem. An MCDM problem refers to making a preference decision over the available alternatives that are characterised by multiple, usually conflicting, attributes (Jadhav & Sonar, 2009). Jadhav and Sonar identify three goals for evaluating software:

1. To help decision makers choose the best alternatives of those studied.
2. To help sort out alternatives that seem good among the set of alternatives studied.
3. To help rank the alternatives in decreasing order of performance.

An evaluation of software might be undertaken for a number of reasons such as risk assessment, maintenance, cost prediction, architecture comparison or trade-off analysis (Ali Babar *et al.*, 2004). Statemlos and Tsoukias analysed different situations where evaluations of software might be required (Stamelos & Tsoukias, 2003). They classified two main "problem situations" as:

- *Keep or change* - an evaluation may be needed when a tool, which is already in wide use, begins to be questioned as to whether or not it still meets the needs of its users. If a decision is taken to 'Keep' a particular tool, it is ideal for all necessary additional features (required by the users) to be implemented within the existing system. If a decision is taken to 'Change' tools, a new decision based on the next category will need to be made.
- *Make or buy* – when a new tool is needed by an organisation (or community of users), a decision must be made as to whether there is an adequate option already available (i.e. there is an existing tool, which already meets the needs of users) or whether a new tool will need to be developed from scratch.

Technology evaluations are typically performed in an ad hoc way and are heavily reliant on the skills and intuition of those carrying out the evaluation (Brown & Wallnau, 1996). To help guide and provide structure to the evaluation process, a variety of tool-specific frameworks (or models) to assess systems for their suitability, have been developed. Generally, creating an evaluation framework should involve "gathering objective data on the target technology, subjective opinions and experiences with the new technology and comparing the new technology with existing practices" (Brown & Wallnau, 1996). For many years, software engineering researchers have developed and applied these frameworks to various tools. For example:

- Blanc and Korn developed a structured approach to assist with the evaluation and selection of **computer-aided software engineering (CASE) tools** (Blanc & Korn, 1992). Similar frameworks were also developed by du Plessis (du Plessis, 1993) and Misra (Misra, 1990) to evaluate this technology.

- Cochran and Chen developed a fuzzy multi-criteria selection of object-oriented **simulation software** (Cochran & Chen, 2005). Similar work was undertaken (earlier) by Hlupic and Mann (Hlupic & Mann, 1997) and Nikoukaran and Paul (Nikoukaran & Paul, 1999).
- A framework for the assessment of **knowledge management tools** was developed by Patel and Hlupic (Patel & Hlupic, 2002). Such tools have been evaluated in many other studies using similar frameworks (Tyndale, 2002; Ngai & Chan, 2005).
- Collier *et al* produced a model for evaluating and selecting data mining software (Collier *et al*., 1999). **Data mining tools** have since been evaluated by many other researchers (Khalifelu & Gharehchopogh, 2012; Dejaeger *et al*., 2012).
- Blanc & Jelassi developed a multiple criteria decision framework to assist evaluating and selecting **decision support software** (Blanc & Jelassi, 1989). Other studies have evaluated DSS using similar models (Johnston *et al*., 2004; Leslie *et al*., 2006).

Evaluation frameworks are highly useful for supporting the selection of a particular tool based on a comparative assessment of existing systems. There is also a need, however, for a framework to serve as a requirements specification for a new or enhanced tool (Iyer & Richards, 2004).

3.2.2 Multiple criteria decision analysis (MCDA) and related techniques

MCDA methods can support decision makers faced with evaluating alternatives by taking into account multiple criteria in an explicit manner (Belton & Stewart, 2002). MCDA techniques provide a structured and transparent approach to identify a preferred alternative by clear consideration of the relative importance of the different criteria and the performance of the alternatives on the criteria (Thokala & Duenas, 2012). The degree to which one decision is preferred over another is often represented by constructing and comparing numerical scores (Thokala & Duenas, 2012).

According to Thokala & Duenas, the main aspects which define a MCDA method are:

1. The alternatives (or candidates) to be appraised,

2. The criteria, attributes or features against which the alternatives are appraised,
3. Scores that reflect the value of an alternative's expected performance on the criteria, and
4. Criteria weights that measure the relative importance of each criteria.

In other domains, MCDA has been used to inform healthcare decisions (Baltussen *et al*., 2010), health technology assessments (Husereau *et al*., 2010) and other governmental issues (Nutt et al., 2010). As mentioned in the previous section, evaluating software is also referred to as a multiple criteria decision making (MCDM) problem (Jadhav & Sonar, 2009). Therefore, a number of common evaluation techniques used in software engineering can be described as a form of MCDA. Typical evaluation approaches used in software engineering include:

- *Analytic hierarchy process (AHP)* – AHP has been widely used for evaluating software. Developed by Saaty, AHP is a theory of measurement through pairwise comparisons and relies on the judgement of experts to derive priority scales (Saaty, 1988; Saaty, 2008). The process is based on a hierarchical framework of data (Jadhav & Sonar, 2009). AHP has been applied to evaluate a wide range of tools; including, customer relationship management systems (Colombo *et al*., 2004), automated manufacturing systems (Davis & Williams, 1994) and e-commerce software (Sarkis & Talluri, 2004).
- *Weighted scoring method* – using this technique, weights, ratings and scales are developed and assigned to each criteria (or feature) of a tool. The weights reflect the relative importance of a particular feature, while the ratings/scales determine how easily each tool is able to provide support for that feature. A score is generated for each feature for each tool. Using these values, scores for each category of features can be calculated. These scores are then combined to determine a total score for each tool. The weighted scoring method has been used to evaluate a variety of tools; including, decision support software (Blanc & Jelassi, 1989), data mining software (Collier *et al.*, 1999) and workflow applications (Perez & Rojas, 2000). Feature analysis (see Section 3.2.4) is a specific type of evaluation activity, which is based on the weighted scoring method (Kitchenham, 1996).

- *Fuzzy-based approach* – this technique can be used to evaluate software when performance ratings and weights cannot be easily (and precisely) determined. Although similar to the weighted scoring method, this approach is more flexible and accommodates a more subjective assessment. A fuzzy-based approach to evaluation can be particularly useful when dealing with more complex systems, where the needs for features evolve at an increasing rate. This technique has been used to evaluate simulation software (Cochran & Chen, 2005), manufacturing systems (Bozdag *et al.*, 2003) and large-scale database warehouse applications (Lin *et al.*, 2006) amongst many other types of software.

As discussed by Jadhav and Sonar, each of these techniques presents their own strengths and weaknesses (Jadhav & Sonar, 2009). AHP is a flexible technique, which is able to handle both qualitative and quantitative evaluations of software. It can, however, be time consuming and involves a repetitive process. The main strength of the weighted sum method is its ease of use. However, weightings are assigned subjectively, which makes it difficult to assign weights to a high number of criteria. Therefore, this method is not particularly well-suited to evaluating large-scale applications. A fuzzy-based evaluation approach is flexible and can accommodate the vagueness and ambiguity that occurs during the decision making process. However, allowing for increased subjectivity limits the rigour of the method and can make ranking tools difficult.

It is important to note that (to-date) there is no one method which is the most suitable to use for any evaluation. This is because there is still little consensus on the technical and non-technical issues that an evaluation technique should focus on (Ali Babar *et al.*, 2004). There is a case made by Jadhav and Sonar to develop a framework that can support the evaluation and selection of any kind of software (Jadhav & Sonar, 2009). For now, however, evaluation methods should be optimised based on the goals of the evaluation, which should be explicitly defined beforehand, and the technology under investigation (Ali Babar *et al.*, 2004). There are some characteristics that all evaluation techniques are recommended to include as standard. For example, it is considered, that a method should provide a set of standards, guidelines and heuristics which provide detailed information on all aspects of the evaluation activities (Ali Babar *et al.*, 2004).

3.2.3 DESMET

DESMET is a methodology for comparing and evaluating methods or tools (Kitchenham, 1996). Although the method can be used to evaluate software engineering methodologies; in this section, DESMET is discussed in the context of evaluating tools (i.e. software) only. DESMET is intended to help an evaluator plan and perform an evaluation exercise that is unbiased and reliable. When adopting the DESMET methodology, the first stage is to select an evaluation type. DESMET defines **nine different evaluation types** (see Table 3-1).

Evaluation type	Description
Quantitative experiment	*An evaluation organised as an experiment which aims to investigate the quantitative impact of a selection of tools.*
Quantitative case study	*The case study methodology is employed to investigate the quantitative impact of a tool.*
Quantitative survey	*A survey is undertaken to investigate the quantitative impact of a selection of tools.*
Qualitative screening	*A feature based evaluation which involves developing a set of features and scoring instruments to evaluate a selection of tools. It can be organised in a variety of ways, including an initial screening (see Section 3.2.4).*
Qualitative experiment	*An experiment performed by a group of potential users who try out the tools on a number of typical tasks.*
Qualitative case study	*A feature-based evaluation organised as a case study, where tools are evaluated based on their performance in a real life project.*
Qualitative survey	*A survey is performed to evaluate tools based on the user-experiences reported by participants*
Qualitative effects analysis	*Involves a subjective assessment of the quantitative effect of the tools based on expert opinion.*
Benchmarking	*An assessment is undertaken of the relative performance of a number of tools based on a series of tests.*

Table 3-1 Nine DESMET evaluation types (Kitchenham *et al.*, 1996)

Selecting an evaluation type is an important decision and can be influenced by a number of factors. To assist this decision, DESMET includes selection criteria to help evaluators choose an appropriate evaluation type based on their needs (Kitchenham & Jones, 1997): **Seven selection criteria** are outlined:

1. *Evaluation context* – the circumstances that form the setting of the evaluation. In software engineering, the context of an evaluation is often industrial, although it can be academic as well. In the study reported in this chapter, the context of the evaluation is an academic one.
2. *Nature of impact* – Quantitative impact can be measured by improvements to productivity, maintainability and quality. Impact, however, can also be qualitative; for example, better visibility of progress, better usability and improved tool integration.
3. *Nature of evaluation object* – the particular tool to be evaluated, which automates a well-defined activity. It is noted by Kitchenham that if the purpose is to evaluate and compare a selection of tools, then a feature analysis is likely to be the most appropriate evaluation type (Kitchenham, 1996)**.**
4. *Scope of impact* – the extent of the impact the tool will have on its targeted community of users and the task it aims to automate/support.
5. *Maturity* – the amount of information about the tool which is readily available. If, for example, the tool has been newly developed or is still in development, there will most-likely be limited details available about the system.
6. *Learning curve* – the time required for the evaluator(s) to become familiar enough and proficient with the tool to be able to assess its capabilities.
7. *Evaluation maturity* – the evaluation capability of the evaluators based on their experience, resources and the context of the evaluation.

Using these criteria, the types of evaluation available can be categorised according to the aspects of the tool that are to be examined. For example, if the primary aspects of a tool to be evaluated are the effect it has within an organisation, then quantitative methods of evaluation are deemed most appropriate. If, however, the objective of the evaluation is more concerned with the suitability of a

tool in a given setting, then this can be better determined using a qualitative form of evaluation. Both categories of evaluation (i.e. quantitative or qualitative) can be organised as a formal experiment, case study or survey. Qualitative forms of evaluation, however, can also be organised as a feature analysis. Furthermore, a DESMET evaluation is context-dependent. This means it is not used to rank tools in terms of effectiveness, but instead to retrieve information on which to base a decision about a tool's suitability in a particular context (Kitchenham & Jones, 1997). Due to the context of the study reported in this chapter, a qualitative form of evaluation has been selected; notably, the feature analysis approach. The context, in this case, is academic; specifically, where researchers are undertaking a systematic review within the software engineering domain.

3.2.4 Feature analysis

Feature analysis is a qualitative form of evaluation and involves the subjective assessment of the relative importance of different features plus an assessment of how well each of the features is implemented (Kitchenham, 1997; Jadhav & Sonar, 2009). Traditionally, the main objective of the evaluation is to provide input into a decision about whether an organisation should purchase a tool. However, a feature analysis is also particularly well-suited for evaluating new technology and providing insights into its use (Zelowitz & Wallace, 1998). Feature analysis is an established evaluation method in software engineering (Grimán *et al*., 2006; Hedberg & Lappalainen, 2005) and draws influence from other commonly used techniques within the domain (see Section 3.2.2). According to Kitchenham, a feature analysis should, as a minimum, provide information in the following areas:

- *Suitability for purpose* – does the tool provide appropriate and relevant support?
- *Economic issues* – is the tool affordable to obtain and maintain?
- *Advantages* – what are the main strengths of the tool?
- *Drawbacks* – what are the tool's key weaknesses?

Undertaking a feature analysis can provide a baseline for relevant features of a new technology (Zelowitz & Wallace, 1998). Feature sets (i.e. categories of features) are generated based on the

requirements that users have for the particular tasks that they expect the tool to support (Kitchenham, 1997; Kitchenham & Jones, 1997). The main stages involved in carrying out a feature analysis are (Kitchenham, 1996):

1. Select a set of candidate tools to evaluate.
2. Generate a set of required features for the type of tool being evaluated.
3. Prioritise these features with respect to the requirements of the tool users.
4. Develop a scoring system that can be applied for all features.
5. Carry out the evaluation to determine how well the tools meet the criteria set.
6. Analyse, interpret and present the results.

This process shares similarities to Jadnav and Sonar's five reported activities for undertaking a technology evaluation (Jadnav & Sonar, 2009). These activities include:

1. *Identifying criteria (i.e. features) to be considered for evaluation.*

   (Relates to **step two** of the feature analysis process).

2. *Assigning weights to each criteria,*

   (Relates to **step three** of the feature analysis process).

3. *Setting up a rating scale for each criteria,*

   (Relates to **step four** of the feature analysis process).

4. *Calculating the score,*
5. *Ranking the alternatives and selecting the best one.*

   (Relates **to steps five and six** of the feature analysis process)

Feature analysis is an extremely flexible method for evaluating any type of tool and its process can be tailored to any required level of detail (Kitchenham, 1996). Simple screening evaluations, for example, can be performed quickly and cheaply. This is because a feature analysis is, primarily, a paper-based study with less focus on the need for any implementation of the tools being evaluated (Zelowitz & Wallace, 1998). It is recommended, however, to include more detailed evaluation

elements, which involve assessment based on a tool's implementation and use (Kitchenham, 1996). A feature analysis can also evaluate tools where there are no process metrics currently available to undertake more rigorous quantitative evaluations (Kitchenham, 1996). This makes the feature analysis technique particularly suited to evaluating new, emerging tools, where there is limited evaluative work.

The feature analysis reported in this chapter follows the steps shown in Figure 3-1. This process is based on the main stages for carrying out a feature analysis (Kitchenham, 1996) and important activities for undertaking a technology evaluation (Jadnav & Sonar, 2009). For the most part, the study has been organised as an initial screening. An initial screening focuses on evaluating simple features, which relate to aspects of a tool that are either present, partially present or absent (Kitchenham & Jones, 1997).

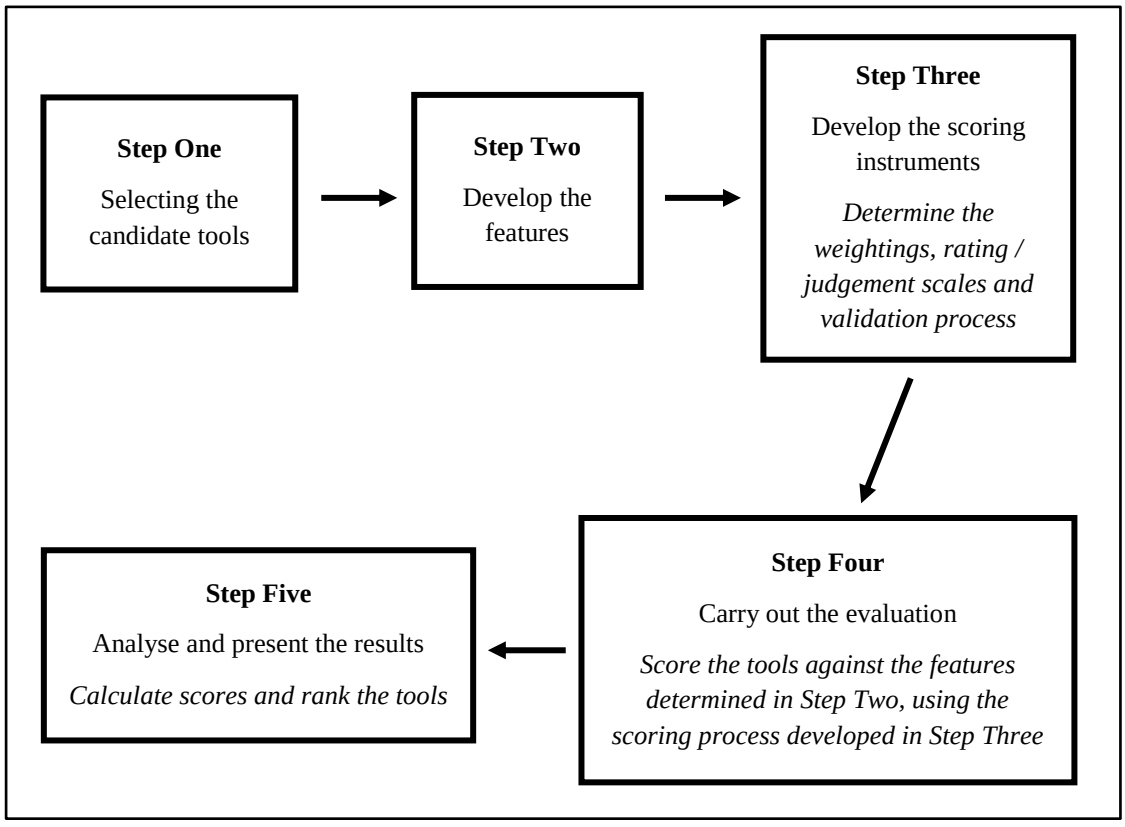


Figure 3-1. Feature analysis process

3.3 Candidates, Features and Scoring

In this section, details are given on how the evaluation was performed. In particular, the first four steps of the feature analysis process (as shown in Figure 3-1) are covered. An overview of the four candidate tools, the features against which the tools are evaluated and the approach taken to scoring the candidates (which includes a series of weightings and rating scales), is provided.

3.3.1 Selecting the candidate tools - (Step One)

The four candidate tools subject to evaluation are:

a) ***SLuRp* (Systematic Literature unified Review program)** which is described as an open source web-enabled database that supports the management of systematic reviews (Bowes *et al*., 2012). The tool has been developed using Java and SQL. *SLuRp* was identified by the literature review reported in Chapter Two.

b) ***StArt* (State of the Art through Systematic Review)** which aims to provide support for each stage of the systematic review process in software engineering (Hernandes *et al*., 2012). *StArt* was identified by the literature review.

c) ***SLR-Tool*** developed in Java, which is described as a freely-available tool to support each stage of the systematic review process in software engineering (Fernández-Sáez *et al*., 2010). *SLR-Tool* was identified in the literature review reported in Chapter Two.

d) ***SLRTOOL***[1], which aims to support the systematic review process in software engineering, amongst other disciplines. The developers state that the guidelines, established by Kitchenham and Charters, underpin its design. *SLRTOOL* was not identified by the literature review reported in Chapter Two as there is no supporting publication.

At the beginning of the study, the developers of each tool were contacted and informed that their tool had been selected as a candidate for the feature analysis. They were asked if they could provide the most up-to-date version of their tool plus any relevant literature or documentation that

[1] www.slrtool.org

supports it. A developer of *SLuRp*, based at the University of Hertfordshire, invited the evaluation team to attend a demonstration of the tool. The team that developed *StArt* provided a recently updated version of their tool, a related publication and a link to a video tutorial. The developers of *SLR-Tool* provided an updated version of their tool plus a user manual and installation guide. A developer of *SLRTOOL* responded that the query had been forwarded to a more suitable member of the team. Following this initial interaction, no further response was received.

3.3.2 Set of features – (Step Two)

As well as covering technical aspects, features should also include economic, cultural and quality aspects (Kitchenham *et al*., 1997). A feature can be decomposed into subfeatures and further broken down into subsubfeatures if required. Doing so, however, risks generating too many features for the evaluation. Having too many features can make evaluating tools very time consuming and can complicate analysis of scores (Kitchenham, 1996). Balance is needed between the depth of understanding required to achieve a desired level of confidence in the evaluation, and the practical difficulties in dealing with a large number of features (Kitchenham, 1996).

The features for this study, presented in Table 3-2, were based on the following factors:

- *Experiences of performing systematic reviews reported in the literature and relevant research regarding tool support for the process.*

As discussed in Section 1.1.5, there are many reported issues about problems associated with undertaking systematic reviews in software engineering, along with calls for tools to support the process. Predominantly, systematic reviews are a highly manual, error prone and labour intensive activity to perform. There are particular challenges concerning certain stages (or aspects) of a systematic review; including, study selection, data extraction, automated searching, collaboration and support for novices (Brereton *et al.*, 2007; Babar & Zhang, 2009; Riaz *et al.*, 2010; Imitaz *et al.*, 2013; Carver *et al.*, 2013). In particular, Carver *et al.* identified some of the primary areas of the systematic review process in software engineering that require automated support (see Section

1.1.5). These factors fed into the development of the initial set of features used for the evaluation. In particular, features most influenced were those that address support for systematic review activities (see Section 3.3.2.3) and process management features (see Section 3.3.2.4).

- *The results of the literature review undertaken and reported in Chapter Two*

The literature review identified the need for an independent evaluation of tools that aim to support the overall systematic review process in software engineering. Its results have also, however, helped influence the set of features developed and used for the feature analysis. In particular, one of the aims of the literature review was to locate any available evidence about the current usefulness of tools. This resulted in a number of classified benefits and costs (or overheads) associated with reported tools (see Section 2.3.6). These factors particularly helped develop features addressing the ease of introduction and setup of a tool (see Section 3.3.2.2) and collaboration.

- *Generic factors from the literature about software/tool evaluation.*

As discussed throughout Section 3.2, many evaluations have been performed to compare and evaluate different tools. Whilst an evaluation's process will largely be tailored based on the context and nature of the candidate tool (i.e. area of support, size, complexity etc.), there is still usually a set of common criteria (or features), which relate to all tools. Features concerning economic factors such as price, maintenance and upgradability, are often investigated in an assessment and comparison of any tool. Therefore, when developing the features for the feature analysis, these generic factors were also considered and included in the initial set (see Section 3.3.2.1).

- *Discussions between members of the evaluation team.*

All of these factors were discussed by the evaluation team to consider a final set of features. The evaluation team consisted of me (the lead evaluator) and both of my PhD supervisors. Both of my supervisors (Prof Pearl Brereton and Prof Barbara Kitchenham) are experienced researchers in software engineering, evaluation and systematic reviews (in particular, the application of the systematic review methodology within the software engineering domain).

id	Feature Set	id	Feature	Feature Level of Importance	Interpretation of Judgement Scale	Feature set Importance Weighting
F1	Economic	F1-F01	No financial payment	HD	JI1	0.1
		F1-F02	Maintenance	HD	JI1	
F2	Ease of introduction and setup	F2-F01	The tool has reasonable system requirements.	M	JI1	0.2
		F2-F02	Simple installation and setup.	HD	JI2	
		F2-F03	There is an installation guide.	HD	JI1	
		F2-F04	There is a tutorial.	HD	JI1	
		F2-F05	The tool is self-contained.	HD	JI1	
F3	Systematic review activity support	F3-F01	Protocol development	D	JI3	0.4
		F3-F02	Protocol validation	D	JI3	
		F3-F03	Supports automated searches	HD	JI3	
		F3-F04	Study selection and validation	HD	JI3	
		F3-F05	Quality assessment and validation	HD	JI3	
		F3-F06	Data extraction and validation	HD	JI3	
		F3-F07	Data synthesis	HD	JI3	
		F3-F08	Text analysis	N	JI1	
		F3-F09	Meta-analysis	N	JI1	
		F3-F10	Report write-up	N	JI3	
		F3-F11	Report validation	N	JI3	
F4	Process management	F4-F01	Support for multiple users	M	JI1	0.3
		F4-F02	Document management	M	JI1	
		F4-F03	Security	D	JI1	
		F4-F04	Management of roles	HD	JI1	
		F4-F05	Support for multiple projects	M	JI1	

Table 3-2 Features, assigned weightings and interpretation of judgement scale used in the feature analysis (version 1.0)

The features developed for this evaluation were divided into four sets. Feature sets related to economics, ease of introduction, systematic review activity support and process management (see Table 3-2). The following subsections describe the features in each of these sets and the criteria suggested for their assessment.

3.3.2.1 Feature set 1: Economic

This set concerns economic factors relating to the initial cost of the tool and subsequent support for maintaining and upgrading the tool. Features (and suggested assessment criteria) include:

- **No financial payment (F1-F01)**
    - Suggested assessment criterion:
        - *The tool does not require any financial payment in order to use the tool.*
- **Maintenance (F1-F02)**
    - Suggested assessment criteria:
        - *The tool is well and freely maintained by its developers.*
        - *The tool is regularly updated with new features.*
        - *There is a single point of contact to obtain support if needed.*

3.3.2.2 Feature set 2: Ease of introduction and setup

This feature set focuses on the level of difficulty inherent in setting up and using the tool for the first time. Features (and suggested assessment criteria) include:

- **Reasonable system requirements (F2-F01)**
    - Suggested assessment criterion:
        - *The tool does not require any advanced hardware or software to function.*
- **Simple installation and setup procedure (F2-F02)**
    - Suggested assessment criterion:
        - *The tool is considered by the evaluator simple to install and setup.*
- **Installation guide (F2-F03)**
    - Suggested assessment criterion:
        - *There is an effective installation guide included with the tool.*
- **Tutorial (F2-F04)**
    - Suggested assessment criterion:
        - *There is an effective, preferably interactive tutorial available.*

- **Self-contained (F2-F05)**
    - Suggested assessment criterion:
        - *The tool is able to function, primarily, as a stand-alone application with minimal requirements for other external technologies.*

3.3.2.3 Feature set 3: Systematic review activity support

These features relate to how well the tool supports each of the three main phases of a systematic review and the steps (or stages) within these phases.

To support the planning phase, features (and suggested assessment criteria) include:

- **Protocol development (F3-F01)**
    - Suggested assessment criteria:
        - *Provides a collaborative template to develop the protocol.*
        - *Supports automated version control to keep track of changes.*
- **Protocol validation (F3-F02)**
    - Suggested assessment criterion:
        - *Supports automated evaluation checklists, which are distributed internally and/or externally for review.*

For the conduct phase, features (and suggested assessment criteria) include:

- **Supports automated searches (F3-F03)**
    - Suggested assessment criteria:
        - *Able to perform an automated search of various electronic resources from within the tool.*
        - *Handles search string format conversion depending on the requirements of digital libraries.*
        - *The tool identifies any duplicates and handles them accordingly*

- **Study selection and validation (F3-F04)**
    - Suggested assessment criteria:
        - *Supports a multi-stage selection process (i.e. title/abstract then full paper).*
        - *Allow multiple users to apply the inclusion/exclusion criteria independently.*
        - *Provides a facility to resolve disagreements.*
- **Quality assessment and validation (F3-F05)**
    - Suggested assessment criteria:
        - *Enables the use and development of suitable quality assessment criteria.*
        - *Allow multiple users to perform the scoring independently.*
        - *Provides a facility to solve disagreements.*
- **Data extraction (F3-F06)**
    - Suggested assessment criteria:
        - *Supports the extraction and storage of qualitative data using classification and mapping techniques.*
        - *Supports the extraction of quantitative data, which manages specific numerical data reported in a study.*
- **Data synthesis (F3-F07)**
    - Suggested assessment criterion:
        - *Supports automated analysis of extracted data.*
- **Text analysis (F3-F08)**
    - Suggested assessment criterion:
        - *Supports text analysis techniques.*
- **Meta-analysis (F3-F09)**
    - Suggested assessment criterion:
        - *Supports meta-analysis (i.e. statistical analysis).*

To support the reporting phase, features (and suggested assessment criteria) include:

- **Report write-up (F3-10)**
    - Suggested assessment criteria:
        - *Provides a collaborative template to assist the write-up.*
        - *Supports automated version control to keep track of changes.*
- **Report validation (F3-11)**
    - Suggested assessment criterion:
        - *Automated evaluation checklists, which are distributed internally and/or externally for review.*

3.3.2.4 Feature set 4: Process management

This set of features relates to the management of a systematic review. Features (and suggested assessment criteria) include:

- **Support for multiple users (F4-F01)**
    - Suggested assessment criterion:
        - *Allows multiple users work on a single systematic review.*
- **Document management (F4-F02)**
    - Suggested assessment criteria:
        - *Able to manage large collections of papers.*
        - *Can manage the relationship between papers and studies.*
- **Security (F4-F03)**
    - Suggested assessment criterion:
        - *The tool is secure and includes a log-in or similar system.*
- **Role management (F4-F04)**
    - Suggested assessment criterion:
        - *Supports different users performing specific tasks (e.g. study selection, quality assessment, data extraction) and allocates resources accordingly.*

- **Support for multiple projects (F4-F05)**
    - Suggested assessment criterion:
        - *Users can perform multiple systematic reviews using the tool.*

3.3.3 Scoring process – (Step Three and Step Four)

In this section, the following three elements of the scoring process are described:

1. Scoring each tool against each feature to produce a raw score,
2. Assigning a level of importance to each feature which is used as a weighting (i.e. a multiplier) to convert raw scores to weighted scores for each feature.
3. Determining scores for each feature set and an overall score for each candidate tool.

As mentioned in Section 3.2.4, a feature analysis is primarily considered a paper-based evaluation study (Zelowitz & Wallace, 1997). However, as recommended by Kitchenham, a feature analysis may also include assessment based on a tool's implementation and use (Kitchenham, 1996). For this study, features of tools were assessed by:

- Examining relevant documentation associated with the tool (e.g. papers, manuals, webpages etc.).
- Trying out the tool. In particular, this involved attempting to reproduce parts of a previously completed systematic review (specifically, the literature review organised as a mapping study, reported in Chapter Two), using the tool.

Each tool was initially scored against each feature by the lead evaluator (referred to as CM). The scores were then discussed by all members in the evaluation team (i.e. CM and their PhD supervisory team) to produce a set of validated raw scores. A spreadsheet was used to record raw scores, weighted scores and overall scores.

The following sections provide some more details about the judgement scale and its interpretation for specific features, the assignment of a level of importance weighting to each feature and the approach taken to calculating overall scores for each of the tools.

3.3.3.1 Judgement scale and its interpretation

A single simple judgement scale was used to score the features. Where a feature was considered fully present or strongly supported it was awarded a score of 1, where it was partly present or partially supported it was awarded a score of 0.5 and where it was absent or minimally supported it was awarded a score of 0.

The judgement scale was interpreted for each of the features in one of three ways, labelled JI1, JI2 and JI3 in Table 3-2. The interpretations are shown in Tables 3-3, 3-4 and 3-5.

Is the feature present?	Score
Yes	1
Partly	0.5
No	0

Table 3-3 JI1 interpretation of judgement scale

3.3.3.2 Level of importance

An effective tool is one that includes features that are most important to its target users (Kitchenham *et al*., 1997). Kitchenham *et al*. state that if a tool fails to include a mandatory feature, then it is, by definition, unacceptable (Kitchenham *et al*., 1997). Non-mandatory features allow the evaluator to judge the relative merit of a group of otherwise acceptable tools (Kitchenham *et al*., 1997). For this study, a feature can be considered Mandatory (M) or one of three gradations of desirability; namely, Highly Desirable (HD), Desirable (D) or Nice to have (N). Table 3-6 shows the multiplier (i.e. the weighting) associated with each level of importance. The level of importance assigned to each feature is shown in Table 3-2. The importance levels were determined by the same process used to develop the initial set of features (see Section 3.3.2).

Is the tool simple to install and setup?	Score
Yes	1
Some difficulties OR The tool could be installed, but there were a number of slight difficulties throughout the process.	0.5
No - The tool could be installed but the process was very difficult. OR No - The tool could not be installed.	0

Table 3-4 JI2 interpretation of judgement scale

Is the activity supported?	Score
Yes - Fully	1
Partly (Support is limited. Some aspects of the activity are not supported.)	0.5
No	0

Table 3-5 JI3 interpretation of judgement scale

3.3.3.3 Feature set and overall scores

As indicated above, a weighted score for each feature is calculated by multiplying the raw score by the importance weighting for that feature. These weighted scores can be combined to determine a percentage score for each feature set (as shown for example in column 8 of Table 3-8).

The percentage score for a feature set is determined as follows:

$$Percentage\ Score = \frac{Sum\ of\ Weighted\ Score}{Maximum\ Score} \times 100\%$$

The maximum score for a feature set is assumed to be the sum of the weighted scores where all features in the set are fully present (or fully supported).

For example, Feature Set 1 (F1) has two features (F1-F01 and F1-F02). F1-F01 has been classified as highly desirable (HD). This means the maximum weighted score for this feature is three. Similarly, F1-F02 has been classified as HD so its maximum score is also three. Therefore, the maximum score for F1 is six. Similarly, the maximum scores for the remaining feature sets are 16 for Feature set 2, 23 for Feature set 3 and 17 for Feature set 4.

Importance	Multiplier
Mandatory (M)	*4
Highly Desirable (HD)	*3
Desirable (D)	*2
Nice to have (N)	*1

Table 3-6 Level of importance of a feature

An overall percentage score for each tool can be determined by taking a (weighted) average of the percentage scores for each feature set. Since there are a number of different features in each of the feature sets, it is necessary to use normalised scores (i.e. the percentage scores) for this. For this calculation, the feature set weighting shown in Table 3-7 is used. The values selected for this evaluation emphasise support for systematic review activities (F3) and for process management (F4). However, other weightings could be used, perhaps to emphasise usability, as tools to support systematic reviews become more mature.

Feature Set	Weight
F1	0.1
F2	0.2
F3	0.4
F4	0.3

Table 3-7 Feature set weighting

The overall score for each tool can be determined using the following equation:

$$Overall\ score = \frac{\sum_{i=1}^{4}(w_i TP_i)}{\sum_{i=1}^{4}(w_i)} \quad (3.1)$$

Where w_i is the weighting for the i^{th} feature and TP_i is the percentage score for the i^{th} feature set.

This study was intended to assess the potential of systematic review support tools from the viewpoint of the tasks that are undertaken during a collaborative systematic review. For this reason, the values selected for the overall weights emphasised the feature sets that provide functions needed by a systematic review research team performing a systematic review (i.e. Feature sets 3 and 4). Weightings for feature sets relating to economic and installation issues (i.e. Feature set 1 and Feature set 2) were reduced as these are considered generic tool issues.

3.4 Results – (Step Five)

This section reports the post-validation scores, indicating which of these were modified by the validation process, and the overall scores for each of the candidate tools. A summary of the results for all of the candidate tools are summarised in Table 3-12.

3.4.1 Results for *SLuRp*

Table 3-8 presents the scores for *SLuRp.*

3.4.1.1 Feature set 1

SLuRp is free to use and can be accessed from the development team's website[2] (Bowes *et al*., 2012). The tool is well maintained, regularly updated and provides a single point of contact for a user to obtain help if needed. *SLuRp* scored full marks for this features set.

[2] https://codefeedback.cs.herts.ac.uk/SLuRp/

3.4.1.2 Feature set 2

SLuRp can be used at the developer's website on request. However, it is likely most users will opt to download, install and implement the tool locally. *SLuRp* has a complex setup. To configure a full version of *SLuRp*, a number of external technologies must also be installed; namely, Tomcat, MySQL, LaTeX and R. *SLuRp* can be used without installing LaTeX and R, but doing so will remove some of its features. Some installation instructions can be found at the tool's website. At the time of evaluation, there was no tutorial. *SLuRp* scored 6.5 out of 16 marks for this feature set.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	6/6	100%
	F1-F02	HD	JI1	1	3		
F2	F2-F01	M	JI1	1	2	6.5/16	41%
	F2-F02	HD	JI2	0	0		
	F2-F03	HD	JI1	1	3		
	F2-F04	HD	JI1	0	0		
	F2-F05	HD	JI1	0.5	1.5		
F3	F3-F01	D	JI3	0	0	10/23	43%
	F3-F02	D	JI3	0	0		
	F3-F03	HD	JI3	0	0		
	F3-F04	HD	JI3	0.5	1.5		
	F3-F05	HD	JI3	1	3		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	0.5	1.5		
	F3-F08	N	JI1	1	1		
	F3-F09	N	JI1	0.5	0.5		
	F3-F10	N	JI3	1	1		
	F3-F11	N	JI3	0	0		
F4	F4-F01	M	JI1	1	4	17/17	100%
	F4-F02	M	JI1	1	4		
	F4-F03	D	JI1	1	2		
	F4-F04	HD	JI1	1	3		
	F4-F05	M	JI1	1	4		

Total Score	Overall % Score Using Feature Set Weightings
39.5/62	65.4%

Table 3-8 Scores for *SLuRp*

3.4.1.3 Feature set 3

To-date, *SLuRp* does not support protocol development or automated searches. Most other stages, however, are supported by the tool. To assist quality assessment and study selection, *SLuRp* allows users to, independently, define and apply multiple criteria (see Figure 3-2 and 3-3), throughout a multi-stage selection process (see Figure 3-4). In addition, *SLuRp* identifies disagreements between quality scores, inclusions and exclusions. To resolve disputes, *SLuRp* supports moderation whereby a user, outside of the conflict acts as a mediator (see Figure 3-5).

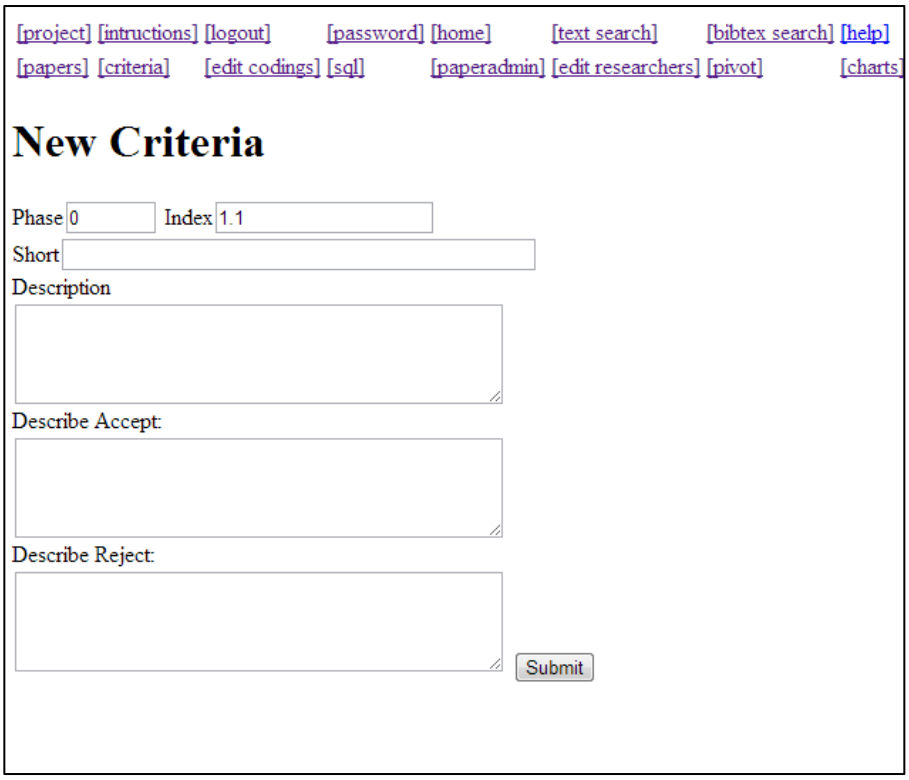


Figure 3-2. Screenshot of the form for defining new criteria (*SLuRp*)

To assist data extraction, *SLuRp* allows users to design and apply two types of data extraction form; namely, a coding form and a performance form. The coding form allows the user to extract and record qualitative data about each paper (see Figure 3-6 and 3-7). It is particularly useful for classification and mappings. The performance form allows users to extract more specific quantitative data from a study (see Figure 3-8). In the supporting paper, a number of features that support automated analysis are described (Bowes *et al.*, 2012). *SLuRp* provides facilities for text analysis using an embedded SQL editor (see Figure 3-9). In addition, meta-analysis is supported

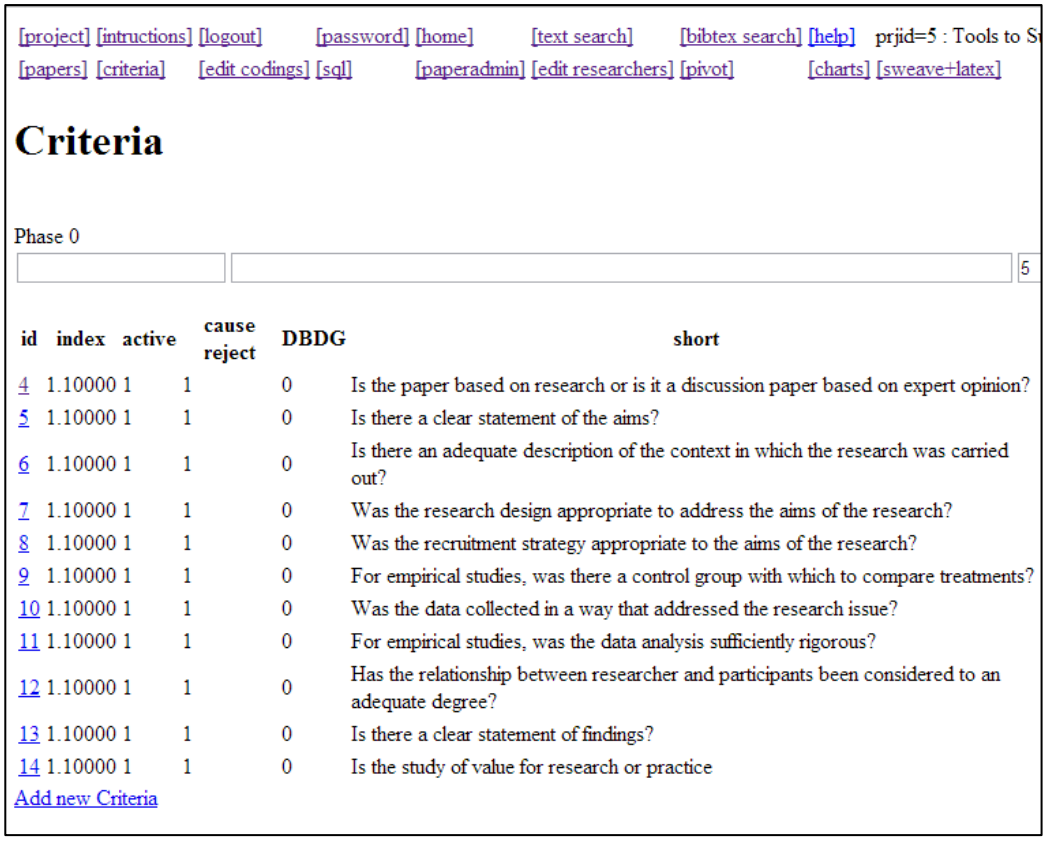


Figure 3-3. Screenshot of the quality assessment criteria page (*SLuRp*)

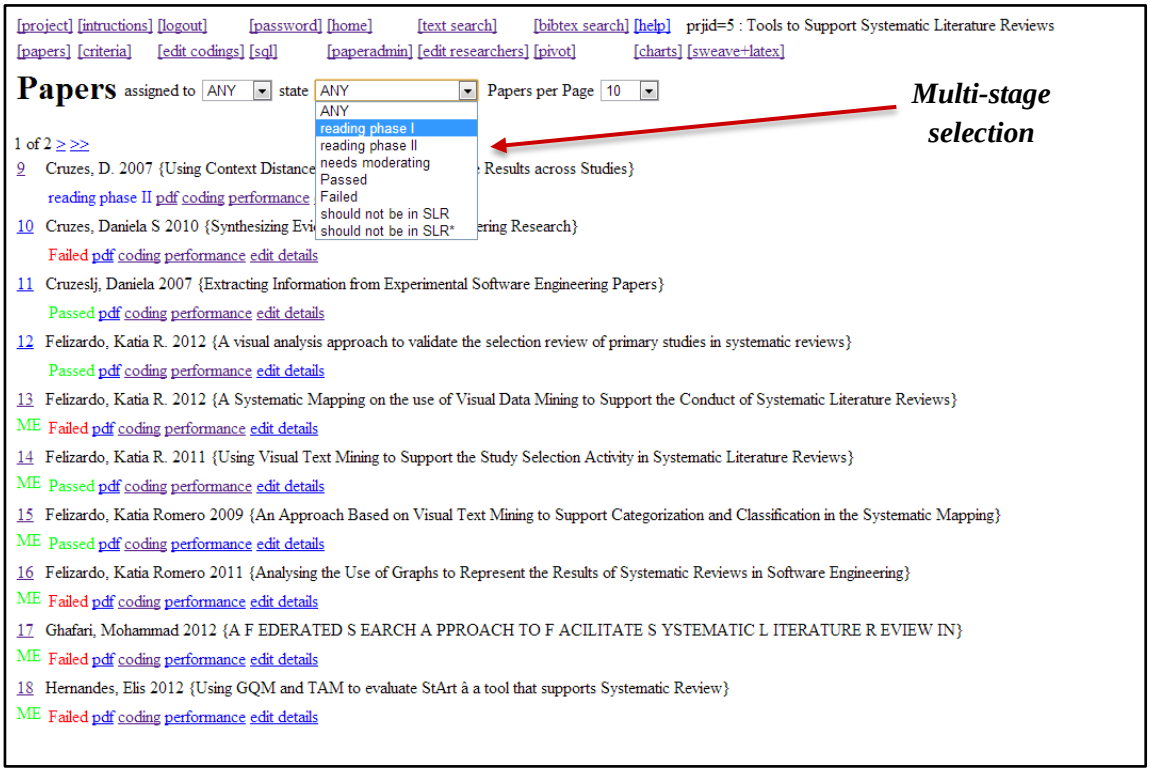


Figure 3-4. Annotated screenshot showing a multi-stage selection process (*SLuRp*)

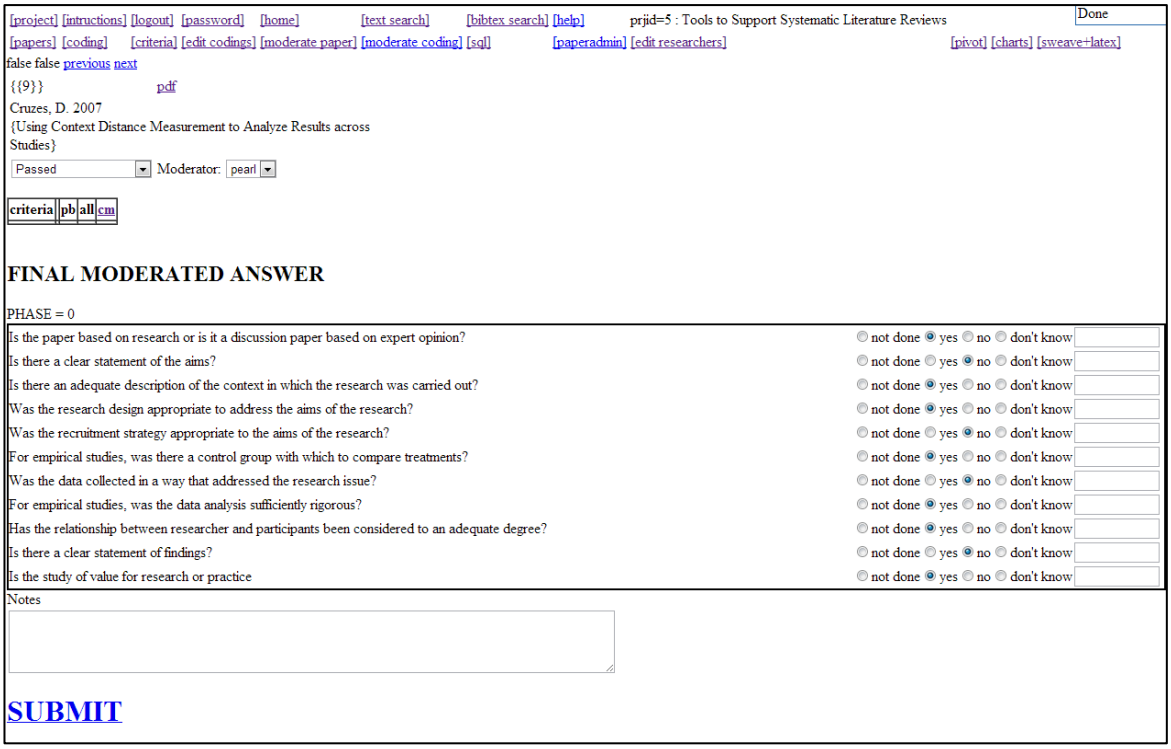


Figure 3-5. Screenshot of the tool's facility for resolving a conflicting quality assessment score, inclusion or exclusion (*SLuRp*)

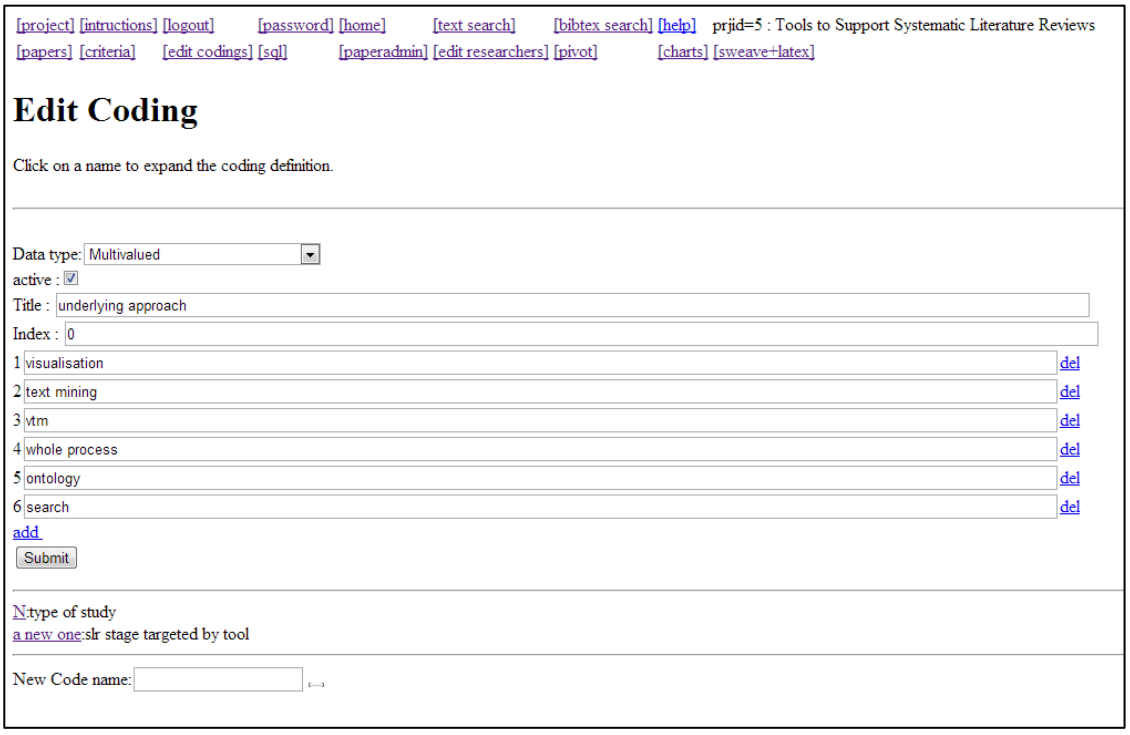


Figure 3-6. Screenshot of the 'coding form' to extract qualitative data (*SLuRp*)

(providing R has been included in the installation). The final report can be written (in full) within the tool providing LaTeX is installed. In total, *SLuRp* scored 10 out of 23 marks for this feature set.

3.4.1.4 Feature set 4

SLuRp allows multiple users to work on a single review (see Figure 3-10) and allows multiple projects to be undertaken. The tool contains a number of useful document management features. Papers can be imported into the tool using BibTeX. As part of the import process, *SLuRp* will attempt to attach a full copy of a paper, automatically. If an attachment fails, a link to its location is provided. *SLuRp* assists with the management of roles. The "super-user" can manage various levels of access for other users and assign them to undertake particular activities (see Figure 3-11).

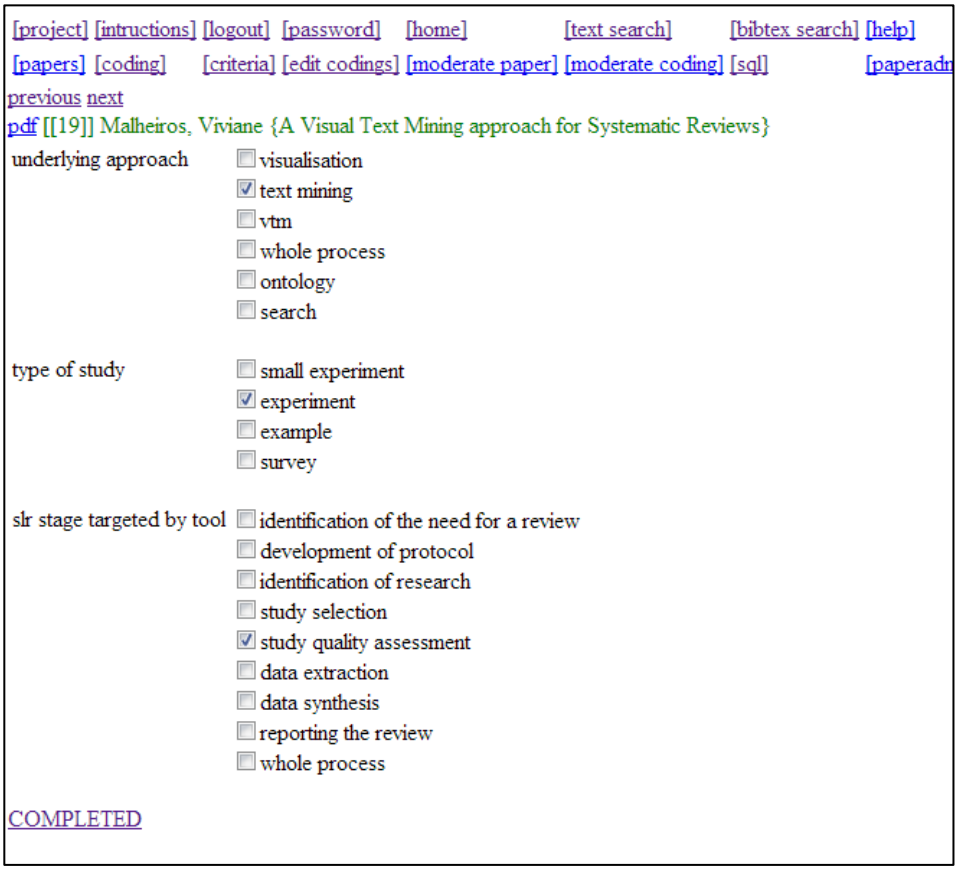


Figure 3-7. Screenshot showing the application of the 'coding form' (*SLuRp*)

SLuRp implements a secure login system which requires a username and password on access. *SLuRp* scored full marks for this feature set.

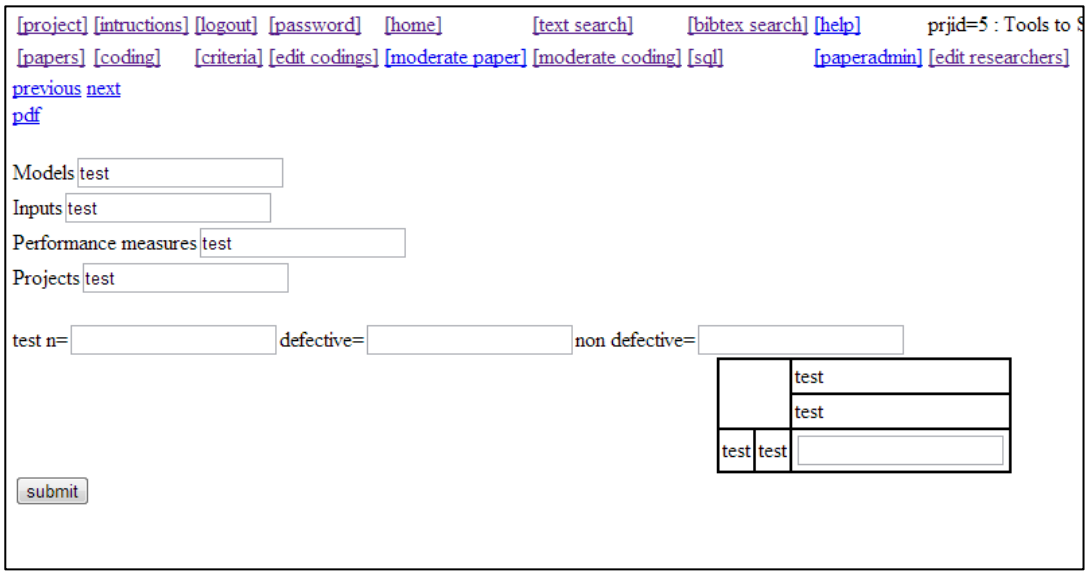


Figure 3-8. Screenshot of the 'performance form' to extract quantitative data (*SLuRp*)

Figure 3-9. Screenshot of the embedded SQL editor (*SLuRp*)

3.4.1.5 Modifications of scores

One score was modified as a result of the validation process. The original score allocated for *SLuRp's* support of quality assessment has been modified. Initially, it was considered that the tool only provided partial support for this stage. However, it was decided that *SLuRp* supported this activity better than first thought. In particular, allowing multiple users to apply the criteria independently and resolve conflicts though moderation helped to increase its score.

Figure 3-10. Screenshot showing the facility to add and remove users (*SLuRp*)

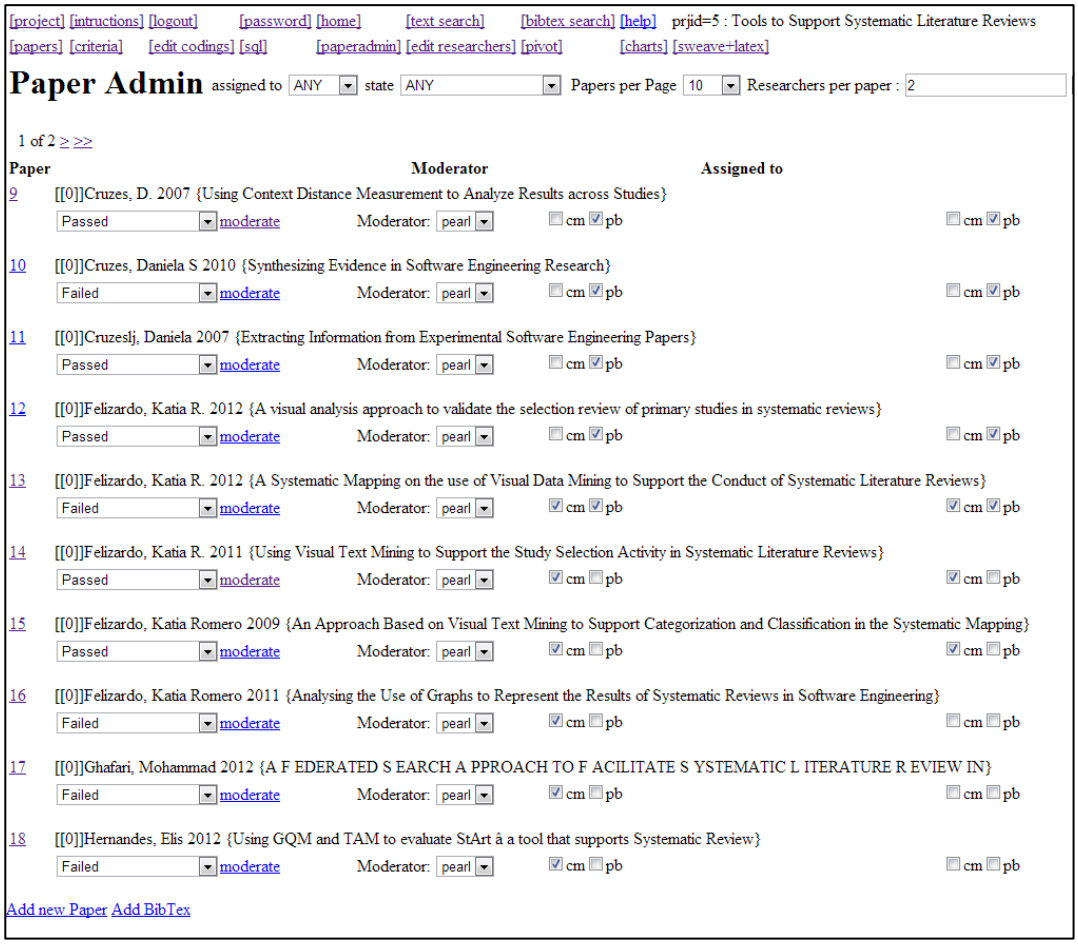


Figure 3-11. Screenshot showing the ability to assign users to different tasks (*SLuRp*)

3.4.1.6 Overall score

As indicated in Section 3.3.3.3, the overall score for the tool is calculated using Equation 3.1 and the feature set weightings shown in Table 3-7. For *SLuRp*, the overall score is **65.4%**

3.4.2 Results for *SLRTOOL*

Table 3-9 presents the scores for *SLRTOOL*.

3.4.2.1 Feature set 1

SLRTOOL can be accessed from the developer's website, free of charge. The tool, however, does not seem to be well maintained. During the study, the tool's website was not always available and, although new features have been planned, the tool has not been updated. *SLRTOOL* scored 3 out of 6 marks for this feature set.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	3/6	50%
	F1-F02	HD	JI1	0	0		
F2	F2-F01	M	JI1	1	4	11.5/16	72%
	F2-F02	HD	JI2	0.5	1.5		
	F2-F03	HD	JI1	1	3		
	F2-F04	HD	JI1	0	0		
	F2-F05	HD	JI1	1	3		
F3	F3-F01	D	JI3	0	0	4.5/23	20%
	F3-F02	D	JI3	0	0		
	F3-F03	HD	JI3	0	0		
	F3-F04	HD	JI3	0	0		
	F3-F05	HD	JI3	0.5	1.5		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	0.5	1.5		
	F3-F08	N	JI1	0	0		
	F3-F09	N	JI1	0	0		
	F3-F10	N	JI3	0	0		
	F3-F11	N	JI3	0	0		
F4	F4-F01	M	JI1	0.5	2	10/17	59%
	F4-F02	M	JI1	0.5	2		
	F4-F03	D	JI1	1	2		
	F4-F04	HD	JI1	0	0		
	F4-F05	M	JI1	1	4		
Total Score					**Overall % Score Using Feature Set Weightings**		
29/62					**45.1%**		

Table 3-9 Scores for *SLRTOOL*

3.4.2.2 Feature set 2

Following a registration process, the tool can be used online at the developer's website. Alternatively, the source-code and database script can be downloaded (from the same website) for local installation. This setup requires an apache web server, PHP and MySQL database. The installation process, although not entirely straight forward, was considered reasonable. Brief installation instructions were found at the website. At the time of this evaluation, there was no tutorial. *SLRTOOL* scored 11.5 out of 16 for this feature set.

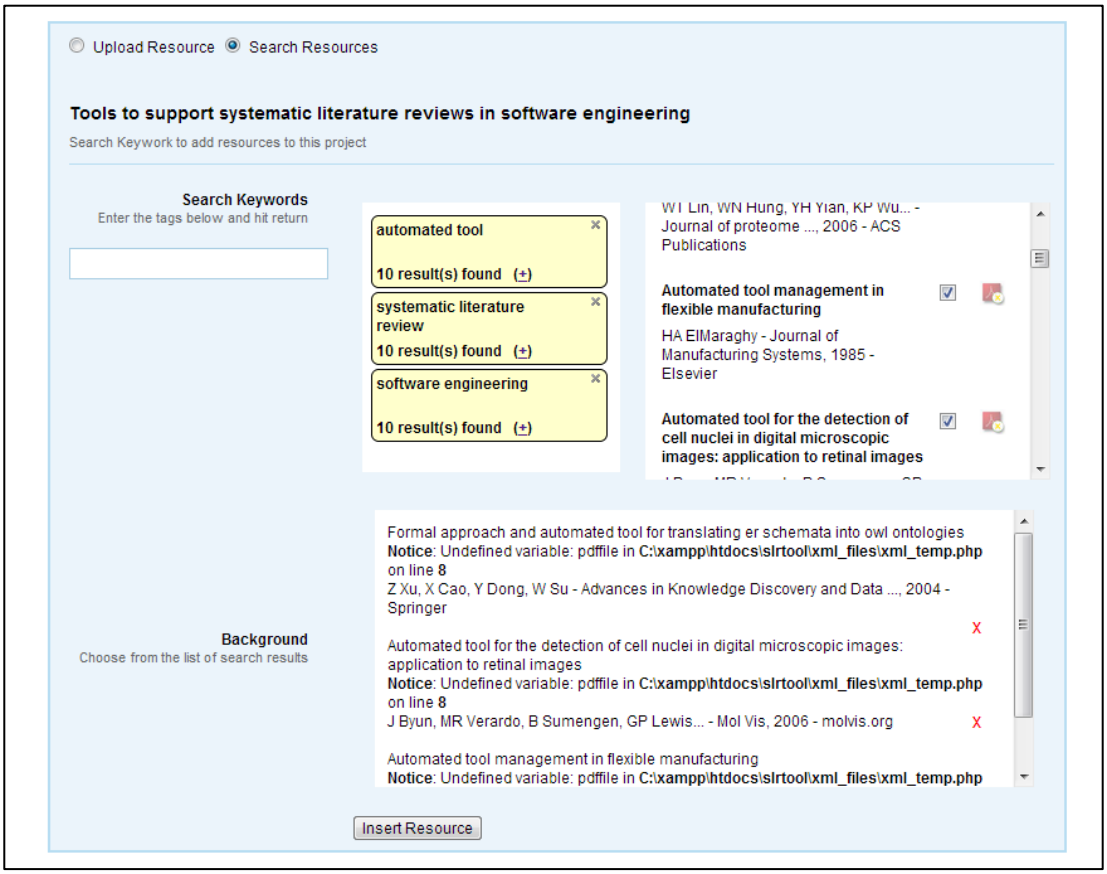


Figure 3-12. Screenshot showing the internal search feature (*SLRTOOL*)

3.4.2.3 Feature set 3

SLRTOOL does not support protocol development or validation. A facility, which allows the user to perform an internal, automated search, has been developed. However, whilst initially promising, the feature is rather limited and only allows for informal, ad-hoc keyword searches for Google Scholar (see Figure 3-12). Whilst there is potential, in its current state this feature does not provide

adequate support. *SLRTOOL* aims to support study selection. However, the support is limited. In particular, users are unable to apply inclusion/exclusion criteria using a multi-stage selection process and multiple users cannot perform their selections independently (see Figures 3-13 and 3-14). Quality assessment is partially supported by the tool (see Figure 3-15). Users can design a three-tier classification form to assist with extraction (see Figure 3-16). Analysis of the data is, however, limited. *SLRTOOL* can perform analysis on certain aspects of the review; such as, study selection, quality assessment, publisher and year of publication. The tool produces a number of charts to visualise these findings (see Figure 3-17). However, automated analysis of extracted data is not performed by the tool. *SLRTOOL* scored 4.5 out of 23 for this feature set.

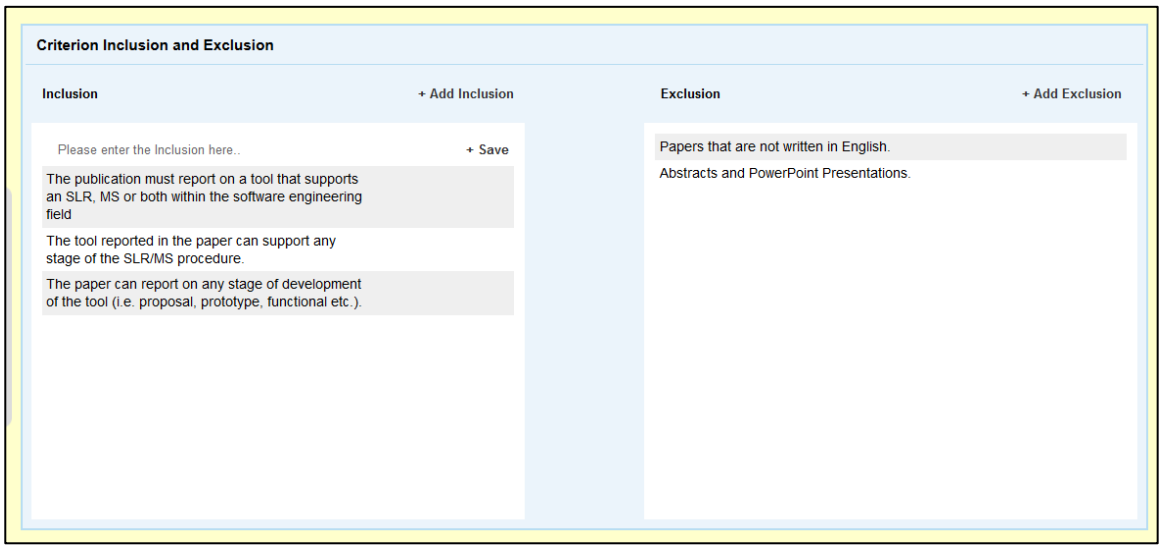


Figure 3-13. Screenshot of the tool's facility to create inclusion/exclusion criteria (*SLRTOOL*)

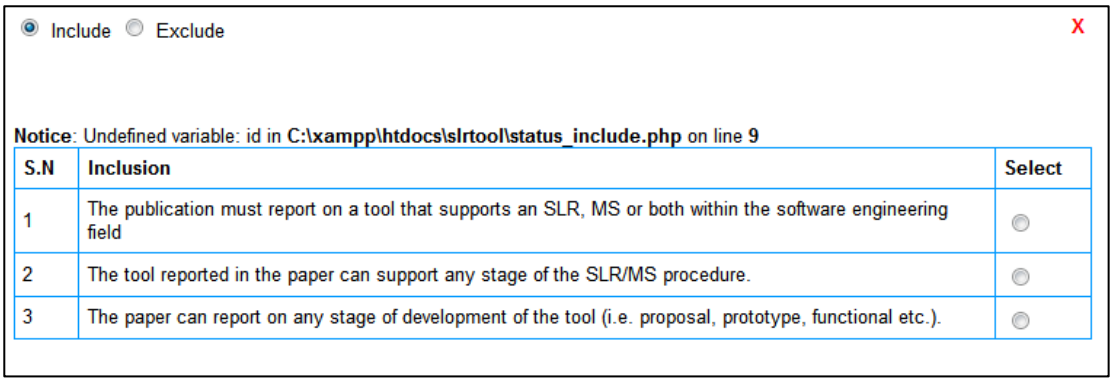


Figure 3-14. Screenshot showing the application of the inclusion/exclusion criteria (*SLRTOOL*)

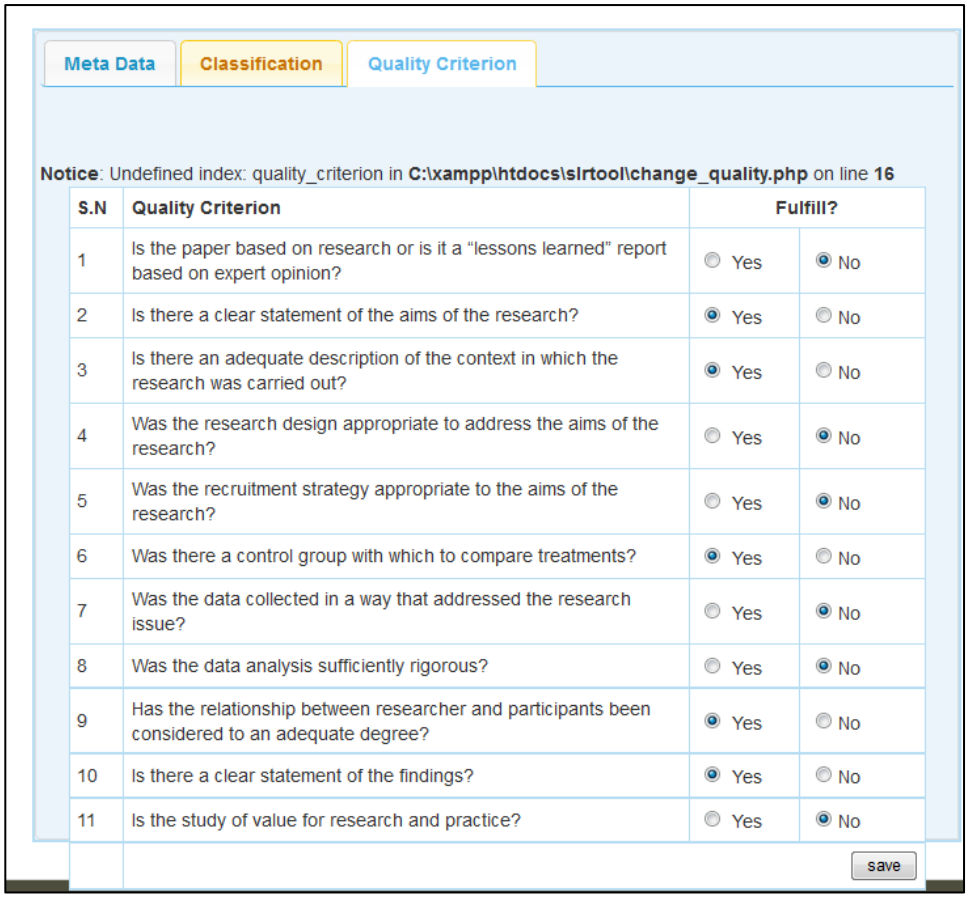


Figure 3-15. Screenshot of applying the quality assessment criteria (*SLRTOOL*)

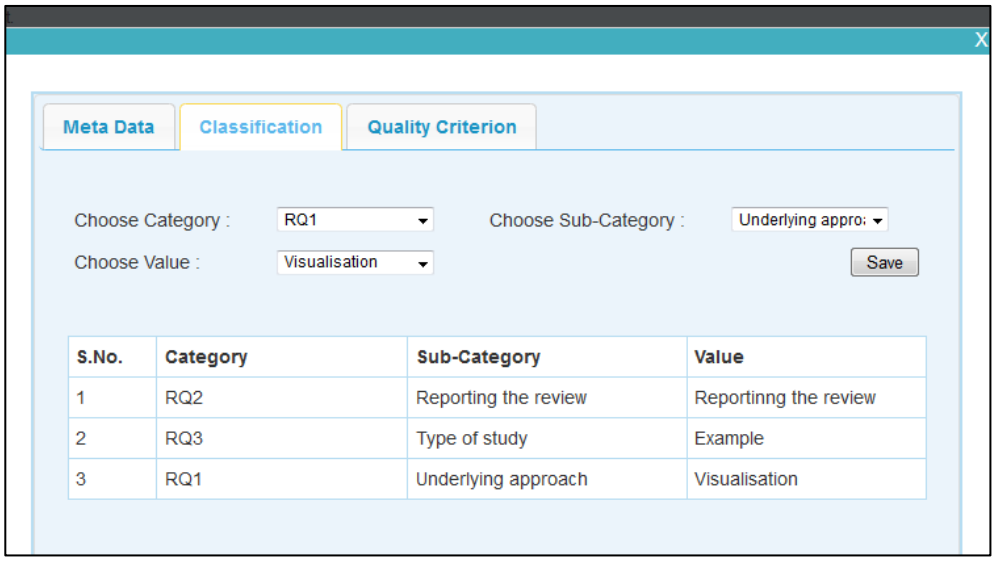


Figure 3-16. Screenshot of designing a classification form for data extraction (*SLRTOOL*)

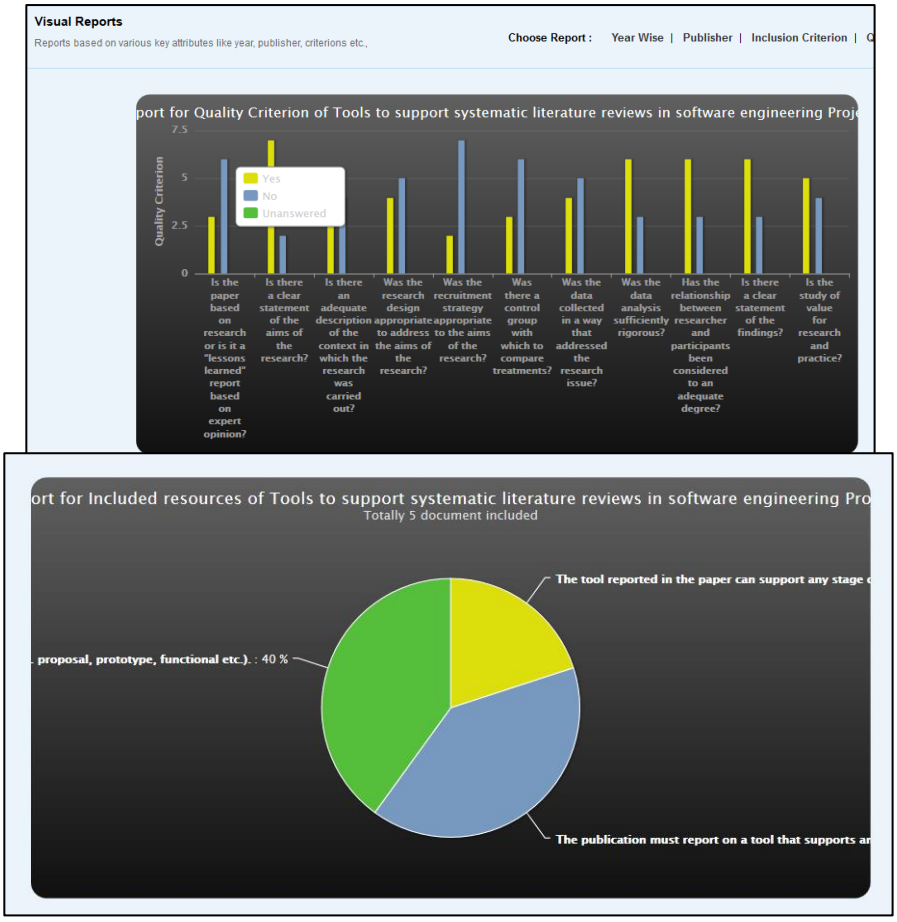


Figure 3-17. Screenshot of a bar chart and pie chart generated by the tool (*SLRTOOL*)

3.4.2.4 Feature set 4

The tool partially supports multiple users. Once a project has been created, new users labelled "collaborators" can be added. Providing a new user has been registered, they can be located and added to a project using the tool's user-search facility. Each user can be a "collaborator" for multiple projects (i.e. systematic reviews) and, at the same time, the "lead-user" of their own projects. *SLRTOOL* does not, however, support management of roles within a project. Support for document management is also limited. Although papers can be exported from within the tool as a BibTeX file, they cannot be imported in bulk using the same method. Papers have to be manually imported one at a time. Once papers are stored, however, the tool provides reasonable facilities to manage and organise them. *SLRTOOL* requires each user to register a username and password, which must be entered at each visit. *SLRTOOL* scored 13.5 out of 17 for this feature set.

3.4.2.5 Modifications of scores

Four scores were modified as a result of the validation process. Initially, partial marks were awarded for *SLRTOOL's* support of an automated search. However, this score was reduced. It was agreed by members of the evaluation team that, although there is potential, the tool does not provide enough support to fulfil the rigour required of a systematic review's search process. The score for *SLRTOOL's* support for study selection was also modified. Initially, partial marks were awarded for its support of the activity. However, once discussed, it was agreed that support was too limited. In particular, users were unable to perform a multi-stage selection process, independently. As a result, the score was reduced. Finally, the initial scores received for *SLRTOOL's* support of multiple users and management of roles, have been revised. The foundations for collaboration, amongst multiple users, are in place. Users can be easily located, added and removed from a project at any given time. However, the tool's support for what are considered collaborative aspects of a systematic review is, generally, quite limited. Due to this, both scores were reduced.

3.4.2.6 Overall score

Using Equation 3.1 (see Section 3.3.3.3) and the feature set weightings shown in Table 3-6, the overall score for *SLRTOOL* is **45.1%.**

3.4.3 Results for *StArt*

Table 3-10 presents the scores for *StArt*.

3.4.3.1 Feature set 1

StArt is free to use and can be downloaded from the developer's website (Hernandes *et al*., 2012). The tool is well maintained and regularly updated with new features and fixes. In addition, there exists a single point of contact for user assistance. *StArt* scored full marks for this feature set.

3.4.3.2 Feature set 2

StArt must be downloaded from the developer's website and installed locally. The tool's setup was simple and easy to perform using a full installation wizard. To assist users, the developers have created an introductory video, providing an overview of the tool and its key features. *StArt* is entirely self-contained and does not require any external applications to be installed. *StArt* scored 14.5 out of 16 for this feature set.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	6/6	100%
	F1-F02	HD	JI1	1	3		
F2	F2-F01	M	JI1	1	4	14.5/16	90%
	F2-F02	HD	JI2	1	3		
	F2-F03	HD	JI1	1	3		
	F2-F04	HD	JI1	0.5	1.5		
	F2-F05	HD	JI1	1	3		
F3	F3-F01	D	JI3	0.5	1	8.5/23	37%
	F3-F02	D	JI3	0	0		
	F3-F03	HD	JI3	0.5	1.5		
	F3-F04	HD	JI3	0.5	1.5		
	F3-F05	HD	JI3	0	0		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	0.5	1.5		
	F3-F08	N	JI1	1	1		
	F3-F09	N	JI1	0	0		
	F3-F10	N	JI3	0.5	0.5		
	F3-F11	N	JI3	0	0		
F4	F4-F01	M	JI1	0	0	6/17	35%
	F4-F02	M	JI1	0.5	2		
	F4-F03	D	JI1	0	0		
	F4-F04	HD	JI1	0	0		
	F4-F05	M	JI1	1	4		
Total Score					**Overall % Score Using Feature Set Weightings**		
35/62					**53.3%**		

Table 3-10 Scores for *StArt*

3.4.3.3 Feature set 3

StArt provides a reasonably detailed template for users to develop a protocol (see Figure 3-18). Its validation, however, is not supported. *StArt* cannot apply search stings directly to digital libraries

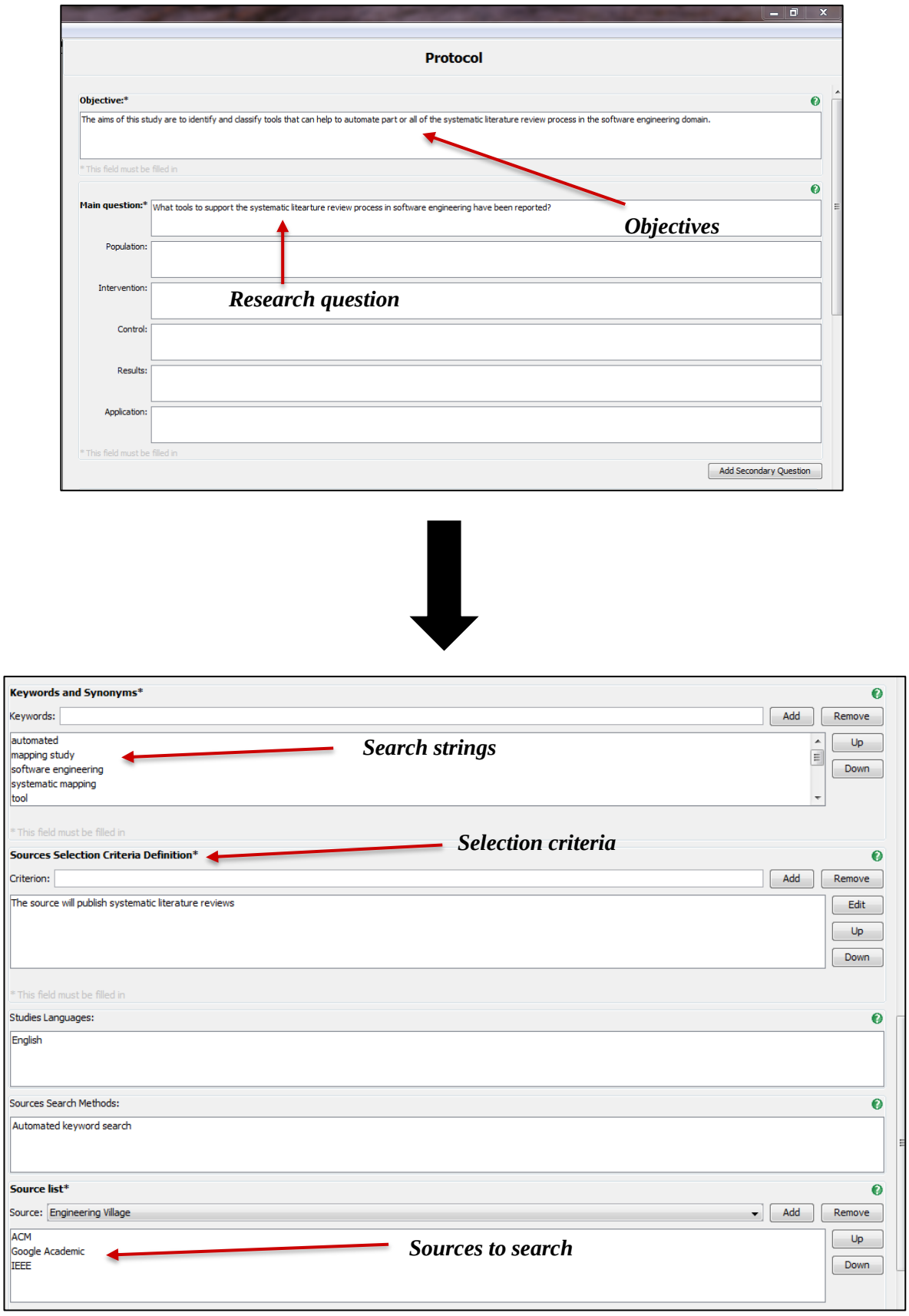


Figure 3-18. Annotated screenshot of the tool's template for developing the protocol (*StArt*)

and retrieve papers automatically. The developers claim this limitation is due to security rules imposed on the search strings (Hernandes *et al.*, 2012). However, *StArt* allows searches to be managed using "search sessions". For each search, a user defines a new "search session". Each "search session" corresponds to a particular resource (that is to be searched) and a search string. Once the user has performed the search, its results are imported and stored within the "search session" (see Figure 3-19). *StArt* provides support for a two-stage study selection process (see Figure 3-20). Quality assessment, however, is not supported by the tool. *StArt* provides partial support for data extraction. Classification forms, designed using the protocol template, can be used to assist this stage (see Figure 3-20). *StArt* can analyse extracted data. The tool employs a number of visualisations to present this analysis (see Figures 3-22 and 3-23).

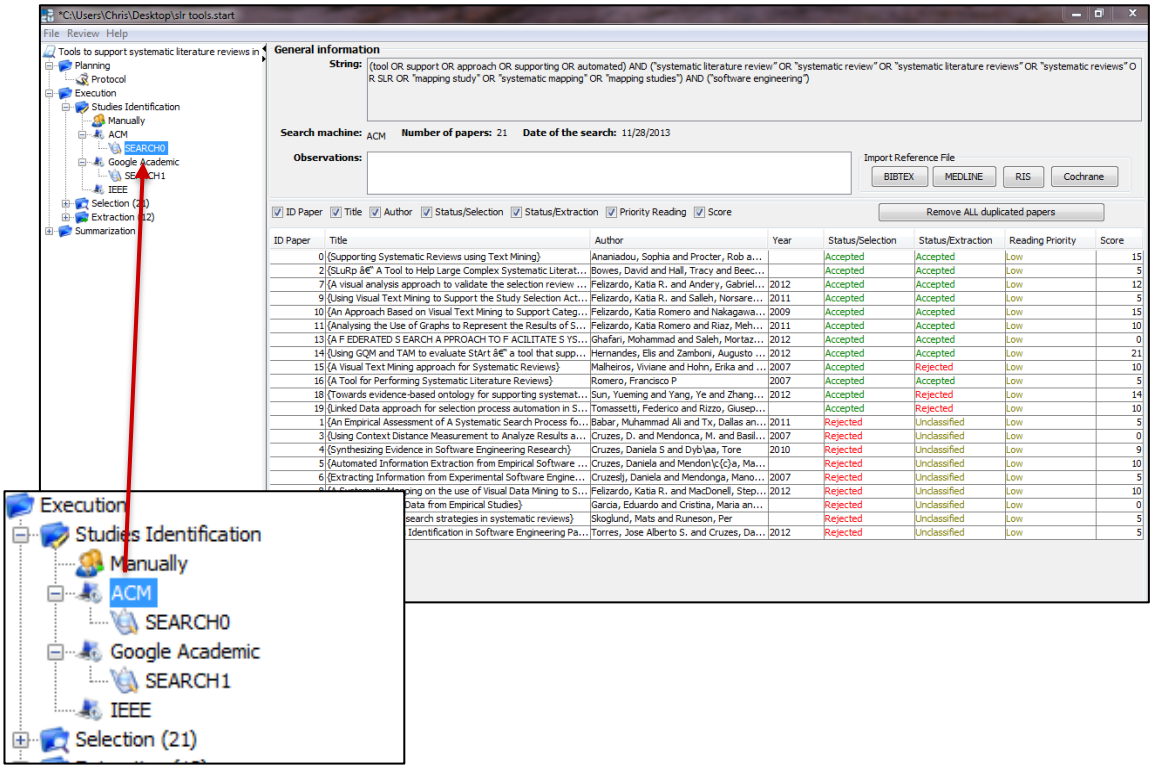


Figure 3-19. Screenshot of a "search session" (*StArt*)

Analysis for quantitative data is, however, limited. *StArt* includes an interesting text analysis feature. The tool generates a "score" for each paper. A score is calculated by matching keywords from a paper's title and abstract, with keywords defined in the protocol. In addition, using the same method, the tool calculates a similarity statistic between papers. Meta-analysis, however, is not

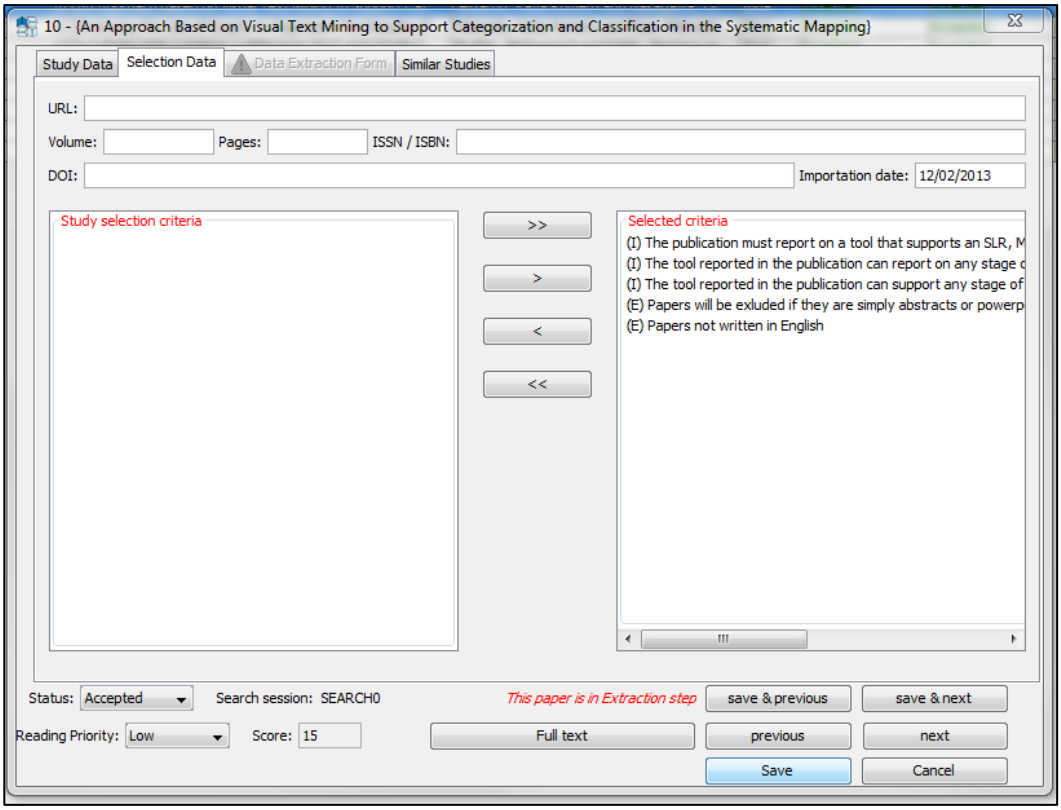


Figure 3-20. Screenshot of applying the study selection criteria (*StArt*)

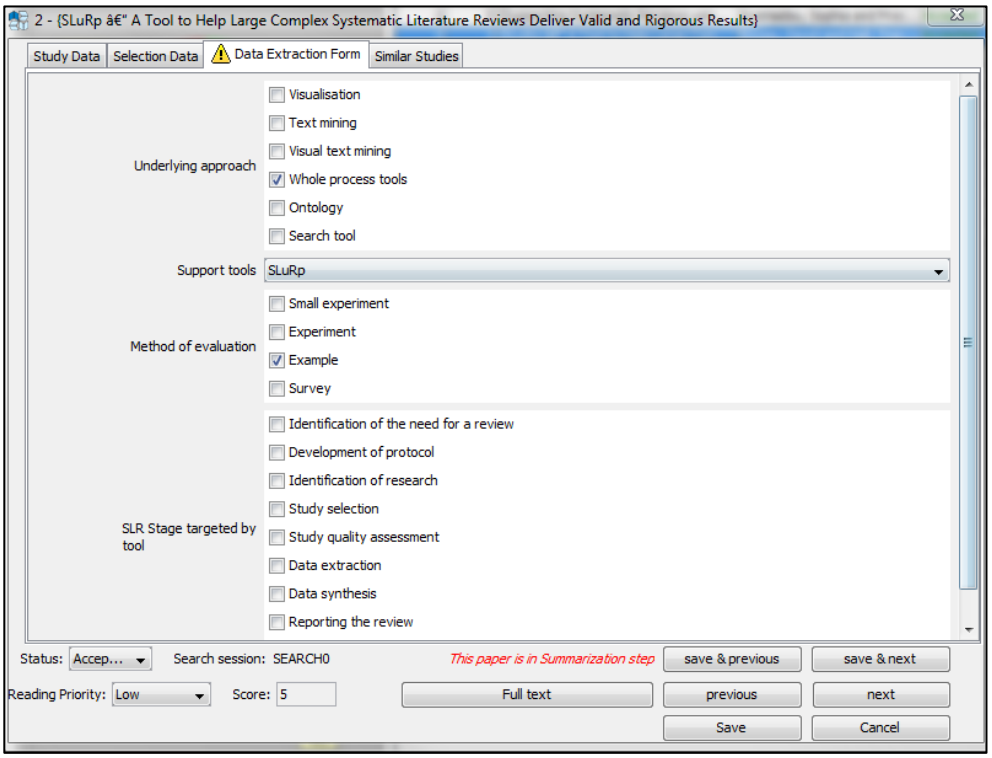


Figure 3-21. Screenshot showing the extraction of data using a classification form (*StArt*)

supported by the tool. *StArt* provides partial support for reporting the review. Tables and charts, produced by the tool, can be exported for use in the report. In addition, users can export the raw data direct to Excel for further analysis. *StArt* scored 8.5 out of 23 for this feature set.

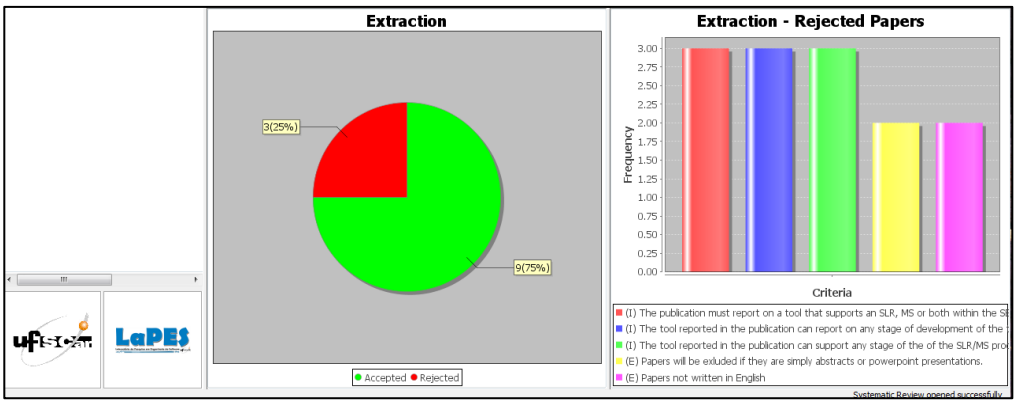


Figure 3-22. Screenshot of a pie chart and bar chart generated by the tool (*StArt*)

3.4.3.4 Feature set 4

StArt does not support multiple users and, therefore, management of roles. Document management is, however, partially supported by the tool. Papers can be imported into the tool in bulk. For this process, *StArt* supports a variety of reference file formats; including, BibTeX, MEDLINE, RIS and Cochrane. The tool provides useful facilities to manage, sort and organise papers. *StArt* does not

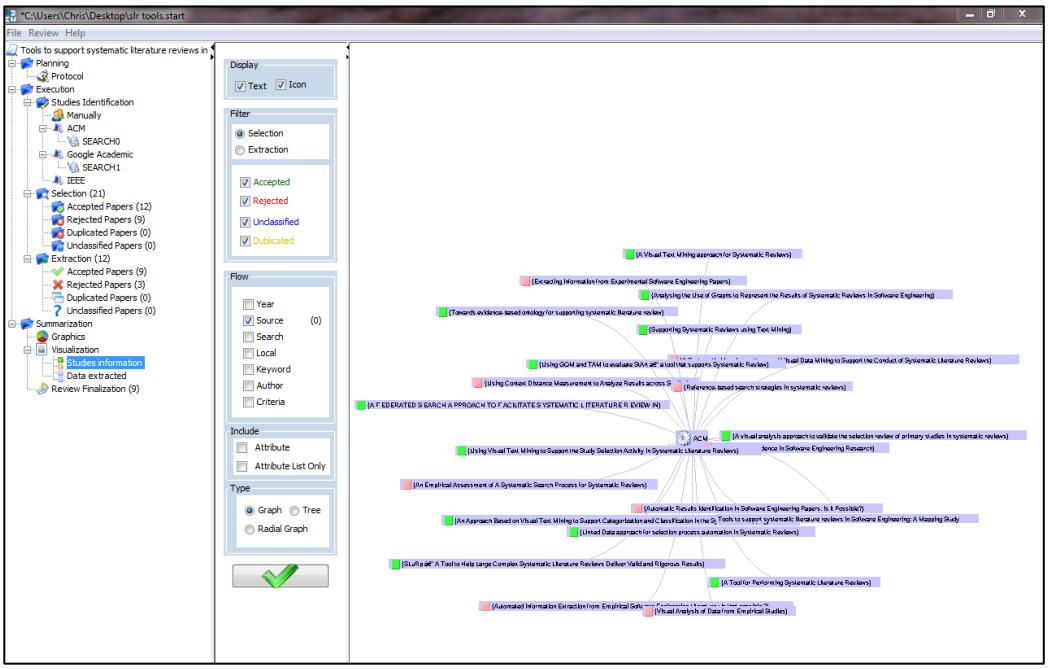


Figure 3-23. Screenshot of an interactive data visualisation generated by the tool (*StArt*)

store full papers. Only a paper's location (providing it is locally stored) can be managed by the tool. *StArt* does not include any features for security. The tool does, however, allow multiple projects to be undertaken. *StArt* scored 6 out of 17.

3.4.3.5 Modifications of scores

Five scores were modified as a result of the validation process. Initially, full marks were awarded for *StArt's* support for protocol development. During the validation process, however, it was highlighted that *StArt* does not provide support for version control. Therefore, the score was reduced. The original score awarded for quality assessment, was also modified. It was agreed that, although a "score" (described in Section 3.4.3.3) is generated by the tool, its calculation process does not reflect the proper procedure for quality assessment in a systematic review. Therefore, the feature's score was reduced. The initial score for data extraction was also reduced. Initially, full marks were awarded for *StArt's* support of this stage. During the validation process, however, it was agreed that support is primarily targeted toward mapping studies rather than full systematic reviews. In addition, data extraction is generally considered a collaborative activity. Since *StArt* does not support multiple users, support for this activity is, therefore, limited. Finally, the initial score awarded for *StArt's* support of automated analysis, was reduced. Originally, full marks were awarded for this feature. However, it was agreed during the validation process that *StArt* focuses, primarily, on analysing qualitative data. Support for quantitative data analysis is limited.

3.4.3.6 Overall score

Using Equation 3.1 (see Section 3.3.3.3) and the feature set weightings shown in Table 3-6. the overall score for *StArt* is **53.3%**.

3.4.4 Results for *SLR-Tool*

Table 3-11 shows the scores for *SLR-Tool.*

3.4.4.1 Feature set 1

SLR-Tool is free to use and can be downloaded from its developer's website. However, the website remains (to-date) in poor condition and it seems that the tool has not been updated for some time. *SLR-Tool* scored 4.5 out of 6 for this feature set.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	4.5/6	75%
	F1-F02	HD	JI1	0.5	1.5		
F2	F2-F01	M	JI1	1	4	14.5/16	90%
	F2-F02	HD	JI2	1	3		
	F2-F03	HD	JI1	0.5	1.5		
	F2-F04	HD	JI1	1	3		
	F2-F05	HD	JI1	1	3		
F3	F3-F01	D	JI3	1	2	10/23	43%
	F3-F02	D	JI3	0	0		
	F3-F03	HD	JI3	0	0		
	F3-F04	HD	JI3	0.5	1.5		
	F3-F05	HD	JI3	0.5	1.5		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	1	3		
	F3-F08	N	JI1	0	0		
	F3-F09	N	JI1	0	0		
	F3-F10	N	JI3	0.5	0.5		
	F3-F11	N	JI3	0	0		
F4	F4-F01	M	JI1	0	0	6/17	33%
	F4-F02	M	JI1	0.5	2		
	F4-F03	D	JI1	0	0		
	F4-F04	HD	JI1	0	0		
	F4-F05	M	JI1	1	4		
Total Score				**Overall % Score Using Feature Set Weightings**			
35/62				**53.2%**			

Table 3-11 Scores for *SLR-Tool*

3.4.4.2 Feature set 2

SLR-Tool requires an installation of MySQL to function. This component is relied on heavily by the tool. Its setup procedure is supported with a reasonably effective installation manual. During

the installation process, an option is available to load an example systematic review project into the tool. When combined with the user manual, this serves as an effective tutorial. *SLR-Tool* scored 14.5 out of 16 for this feature set.

3.4.4.3 Feature set 3

SLR-Tool provides a template for users to develop a protocol. The background, justification, research questions, search strategy (including multiple sources and search strings), quality criteria and study selection criteria can all be defined (see Figure 3-24).

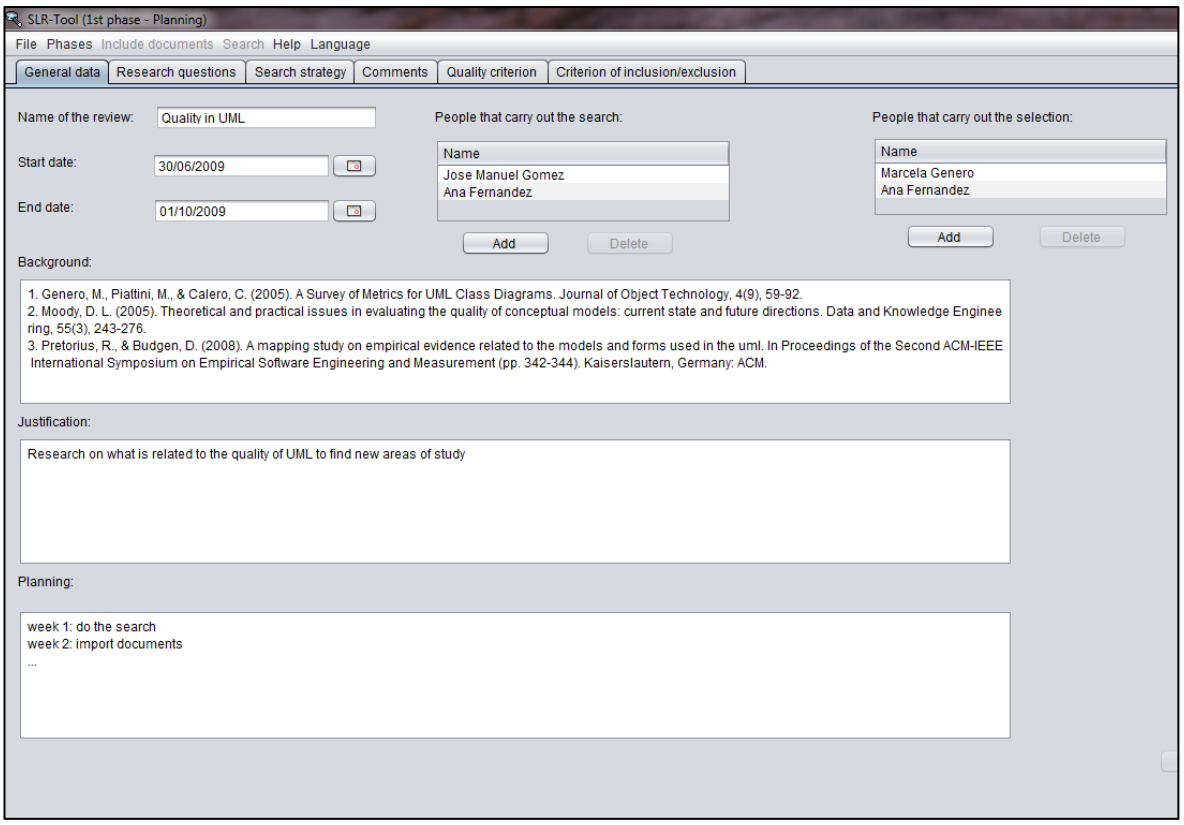


Figure 3-24. Screenshot of the tool's template for developing the protocol (*SLR-Tool*)

The protocol's validation, however, is not supported by the tool. In addition, the tool does not provide support for an automated search. *SLR-Tool* tries to support a multi-stage study selection process, with limited success. Users can apply the study selection criteria defined in the protocol. Papers can be included/excluded during a "first review" and "second review" (see Figure 3-25). In addition, *SLR-Tool* partially supports quality assessment. During the protocol's development, users

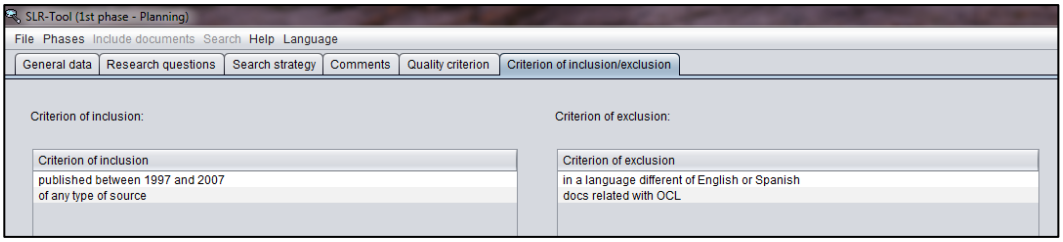


Figure 3-25. Screenshot of the inclusion exclusion criteria defined in the tool (*SLR-Tool*)

design a simple quality assessment questionnaire, which can be applied to included studies. Data extraction is also supported by the tool, all be it, in a limited capacity. Users design classification forms (see Figure 3-26) to extract, primarily, qualitative data. *SLR-Tool* provides effective support for data analysis. The tool generates a variety of charts to visualise findings (see Figure 3-27). Charts can be exported from the tool for use in written reports (see Figure 3-28). *SLR-Tool* scored 10 out of 23 for this feature set.

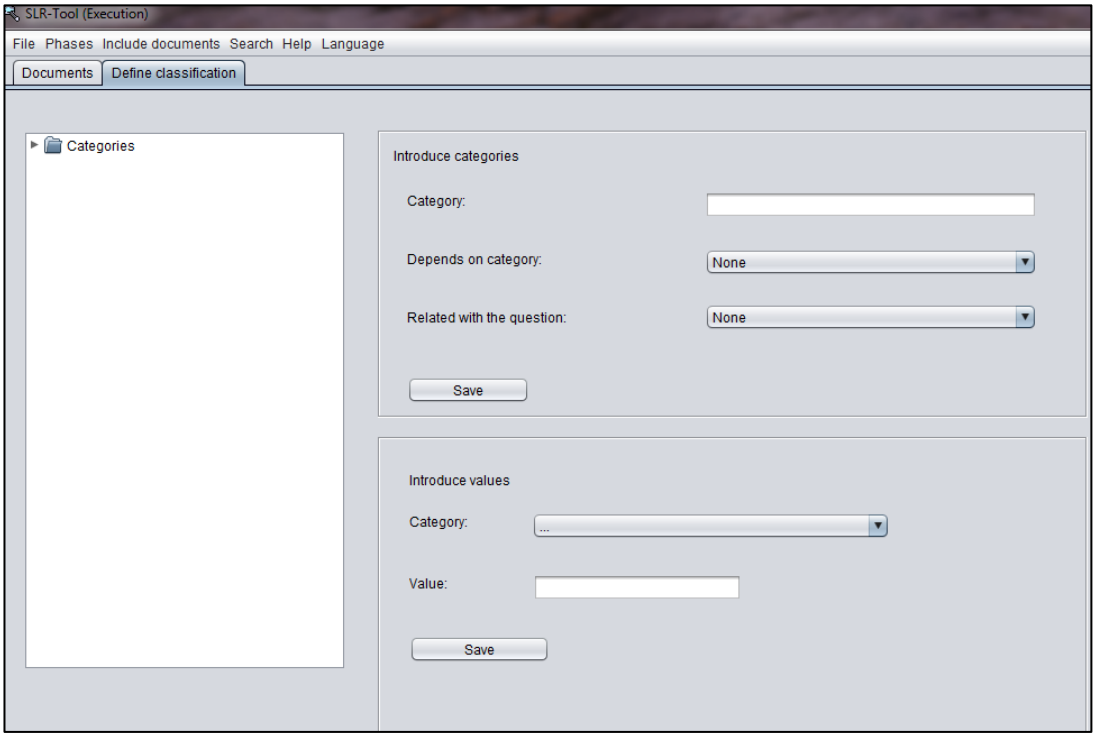


Figure 3-26. Screenshot of designing a classification form to extract data (*SLR-Tool*)

3.4.4.4 Feature set 4

SLR-Tool does not support multiple users nor, therefore, management of roles. Document management is, however, partially supported. The developers indicate that *SLR-Tool* is compatible

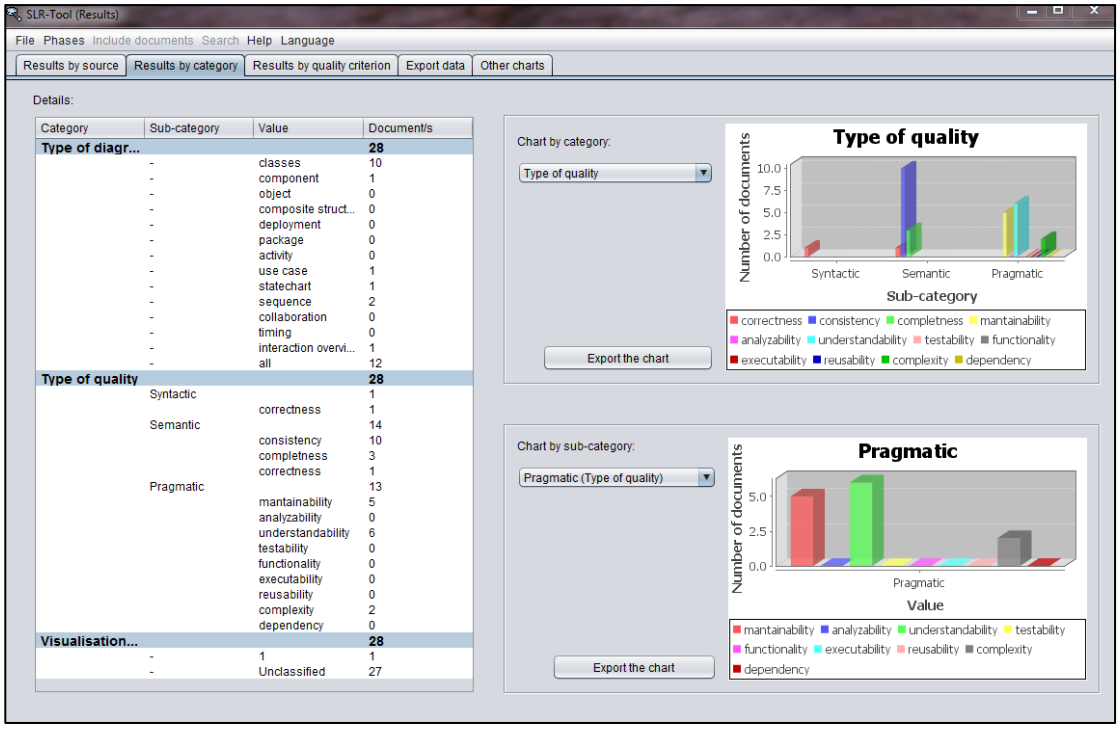


Figure 3-27. Screenshot of the tool's data analysis facilities (*SLR-Tool*)

with a range of reference file formats (including BibTeX, EndNote and RIS) which can be used to import collections of documents (Fernández-Sáez *et al.*, 2010). This feature, however, failed to work during the evaluation process and only allowed papers to be imported individually. Once papers are stored, *SLR-Tool* offers reasonable support for their organisation and management. The tool supports multiple projects to be undertaken. *SLR-Tool* scored 6 out of 17 for this feature set.

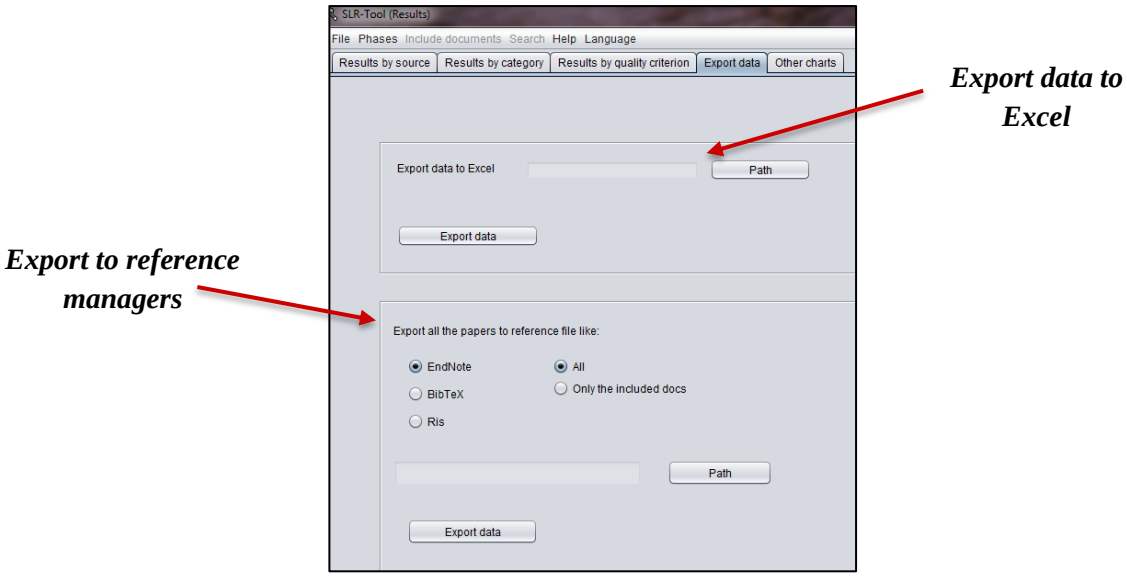


Figure 3-28. Annotated screenshot of the tool's export options (*SLR-Tool*)

3.4.4.5 Modifications of scores

Three scores were modified. Initially, full marks were awarded for *SLR-Tool's* installation guide. However, during validation, it was agreed that its content did not sufficiently cover how to setup the MySQL component. Therefore, the score was reduced. The original score awarded for *SLR-Tool's* tutorial, was increased. Initially, partial marks were awarded for this feature. However, it was agreed that the ability to load an example project into the tool is a highly useful feature (and more useful than initially thought). Finally, partial marks were originally awarded for *SLR-Tool's* support of role management. The tool allows the user to make note of who (i.e. which members of the review team) will perform certain activities; specifically, the search, study selection and quality assessment. However, since the tool does not support multiple users, it was agreed that management of roles cannot be supported effectively. Therefore, the score was reduced.

3.4.4.6 Overall score

Using Equation 3.1 and the feature set weightings shown in Table 3-6, the overall score for *SLR-Tool* is **53.2%**.

3.5 Discussion of the Feature Analysis

This section presents a discussion of the results of the feature analysis, highlighting the main strengths and weaknesses of each candidate tool, as well as implications for the evaluation framework. Limitations of the feature analysis are also discussed.

3.5.1 Discussion of results

As mentioned in Section 2.1.1, systematic reviews in software engineering usually take one of two forms. The 'standard' form, aims to address specific research questions relating to software engineering methods or procedures. The alternative form, termed a mapping study, aims to classify the literature on a specific software engineering topic (Kitchenham *et al.*, 2011). For mapping studies, the search strategy is often less stringent than for standard systematic reviews and quality assessment is not always required. Therefore, these slightly different requirements (in the context of tool support for a full systematic review or mapping study) are considered within this discussion.

As shown in Table 3-12, *SLuRp* achieves the highest overall score of **65.4%** and so within the constraints of this study can be considered the most suitable tool to support systematic reviews in software engineering. The tool's main strengths are:

- Provides full support for a team-based systematic review process.
- Can be used for standard systematic reviews as well as for mapping studies (good support for quality assessment).
- Actively supported by its developer.

SLuRp's main weaknesses are its complex installation, lack of support for protocol development and difficulties associated with the use of the 'performance form' feature.

StArt has an overall score of **53.3%**. Its main strengths are:

- Active support and maintenance by its developers.

- Its simple setup procedure.

StArt is the only tool that does not rely on the installation of any additional applications in order to function. One of *StArt*'s weaknesses is an absence of support for multiple users. As a consequence, many of the systematic review stages that are considered collaborative activities are only partially supported by the tool.

SLR-Tool has an overall score of **53.2%**. The tool's main strengths are:

- Strong support for developing a review protocol.
- Effective support provided to new users; notably, the ability to load an example project into the tool.
- Effective support for automated analysis.

Its main weakness is its lack of support for multiple users. Also, as indicated in Section 3.4.4.4, it was not possible to import collections of papers. This meant that papers had to be manually imported on a paper-by-paper basis.

	F1 (scores out of 6)		**F2 (scores out of 16)**		**F3 (scores out of 23)**		**F4 (scores out of 17)**		**Total (score out of 62)**		**Overall score**
SLuRp	6	100%	6.5	41%	10	43%	17	100%	39.5	64%	**65.4%**
StArt	6	100%	14.5	90%	8.5	37%	6	35%	35	56%	**53.3%**
SLR-Tool	4.5	75%	14.5	90%	10	43%	6	35%	35	56%	**53.2%**
SLRTOOL	3	50%	11.5	72%	4.5	20%	10	59%	29	46%	**45.1%**

Table 3-12 Feature set scores and overall scores

SLRTOOL has the lowest overall score of **45.1%**. The tool has a number of promising and potential features, yet fails to implement them effectively. In particular, it is clear that support for collaboration, amongst multiple users, was a primary design objective. The facility to add/remove users to and from on-going projects is impressive and, generally, works well. Unfortunately, *SLRTOOL* doesn't really allow users to collaborate in any meaningful way. Due to this, much of its support for the systematic review process is quite limited.

3.5.2 Refinements to the framework

The results of and experience gained from the feature analysis have led to some modifications to version 1.0 of the evaluation framework. The changes are:

1. **Reasonable system requirements (F2-F01), installation guide (F2-F03) and tutorial (F2-F04) features are consolidated.**

There are no longer separate features to assess a tool's system requirements, the presence and effectiveness of an installation guide and/or a tutorial. This change was made because, in the context of the evaluation framework, these features were considered too low-level. Instead, these characteristics will contribute toward a higher-level assessment of the simple installation and setup (F2-F02). Reasonable system requirements, an installation guide and/or tutorial are all provided as suggested assessment criteria when evaluating this feature.

2. **Support for multiple projects (F4-F05) is strengthened.**

Assessing a tool's support for multiple projects was considered rather vague. Therefore, the scope of this feature (F4-F05) has been expanded. This feature now focuses on evaluating the value of multi-project support; specifically, support for reusing data from past systematic review projects. At this stage, a level of importance was not determined due to limited experience of reusing past systematic review data in software engineering. Further research undertaken, however, has informed a suitable weighting (see Section 6.2.4.2).

Both of these refinements are reflected in an updated set of features and weightings presented in Table 5-1. Changes made to version 1.0 of the evaluation framework are discussed further in Section 7.3.1.1.

3.5.3 Limitations of the feature analysis

The main threats to validity arise from the subjective nature of many of the elements of the feature analysis process. The features (included in version 1.0 of the evaluation framework) are at this point a preliminary set based on the factors described in Section 3.3.2. However, this activity is intended to provide the foundations for further study of the features expected from a systematic review tool. Similarly, the levels of importance, both for individual features and feature sets, are based on experience. However, these components have been designed to be easily adjusted and re-calculated where requirements and priorities differ (more information about the flexibility of the framework is provided in later sections of this thesis). The scoring of features is also subjective. However, as independent evaluators, the members of the evaluation team have no vested interest in any of the candidate tools.

As described in Section 3.3.3, all scores were subjected to a validation exercise. This process involved all evaluators reviewing all scores for each tool. In particular, each score (and the justification for why it was given) was presented by the lead evaluator, as part of a group discussion with all members of the evaluation team. If a member of the evaluation team felt that a particular (initial) score did not reflect an accurate assessment of a certain tool feature, the score was discussed until a final (validated) score was agreed. The validation exercise aimed to mitigate any potential bias associated with the subjectivity of the scoring process. However, the approach used to validate scores presents additional threats to validity. For example (and similar to a limitation of the mapping study reported in Section 2.3.7), it is possible that an amended score may have been influenced by the fact that the lead evaluator was a PhD student, and the other members of the evaluation team their supervisors. On reflection, an alternative approach where *all* members

of the evaluation team score each tool independently, with the final score being determined through discussion or using some averaging process, may have helped to address this limitation.

In Section 3.3.1, it is noted that the evaluation team accepted an invitation to attend a live demonstration of a candidate tool (*SLuRp*) led by its developer. No such demonstration took place for any of the other tools included in the feature analysis (i.e. *StArt*, *SLR-Tool* and *SLRTOOL*). Therefore, it is possible that the scores for *SLuRp* may have been influenced, either positively or negatively, by this differential treatment. However, whilst this threat to validity is acknowledged, it was considered preferable to attend the demonstration of the tool, based on the aims of the feature analysis and the exploratory nature of the overall work.

3.6 Summary

The study reported in this chapter has evaluated four candidate tools; namely *SLuRp*, *StArt*, *SLR-Tool* and *SLRTOOL* using the feature analysis method. These tools aim to support the whole systematic review process in software engineering. A preliminary evaluation framework (version 1.0), which includes a set of features, their levels of importance and scoring instruments, were developed and used as the criteria against which to evaluate the candidate tools.

SLuRp received the highest overall score and is, therefore, based on the results of this study; the most suitable tool to support systematic reviews in software engineering. *SLRTOOL* received the lowest overall score, making it the least suitable.

The results of this study have provided new insight into tools that support the overall systematic review process in software engineering and; in particular, generated the first (version 1.0) and second version (version 1.1) of an evaluation framework to assess such systems. In the following chapter (Chapter Four), details of a novel resource developed to locate tools to support systematic reviews, are provided. Parts of its design were influenced by the feature analysis, as well as the literature review reported in Chapter Two. In Chapters Five and Six, work undertaken to investigate the use of tools to support systematic reviews in other domains, is presented. Specifically, a cross-domain, interview-based survey has been performed, which also serves as the next validation exercise for the evaluation framework.

Chapter Four

Systematic Review Toolbox

In this chapter, a novel resource, which allows systematic reviewers to identify appropriate tools to support their systematic reviews based on their particular needs, is presented. *Systematic Review Toolbox* is a community driven web-based catalogue of tools that support systematic reviews. Particular types of tools can be queried based on the selection of different criteria. These criteria have been influenced by the work reported in this thesis. Users can add their own tools which they have developed or other tools, which are not currently stored within the database. Details of the motivation behind the development of the resource, its key features and impact within the research community, are presented.

4.1 Introduction to the Systematic Review Toolbox

As discussed in Section 1.1.5, a number of studies have been undertaken which identify and investigate tools to support systematic reviews. In healthcare, a survey of current systems that provide support for systematic reviews identified a variety of tools (Tsafnat *et al*., 2014). Furthermore, a cross-domain mapping study of visual data mining (VDM) techniques identified a number of VDM tools to support data extraction and data synthesis (Felizardo *et al*., 2012). Within software engineering, a broader mapping study of systematic review tools was performed (see Chapter Two), which identified a predominance of visualisation and text mining tools used to support study selection, data extraction and data synthesis (Marshall & Brereton, 2013). Whilst studies like these are useful, however, it remained a challenge for reviewers to easily discover what tools are currently available to support the conduct of their systematic reviews.

Some effort has been made in other domains to provide information to researchers about available systematic review tools. For example, in healthcare, the Cochrane Collaboration provides a webpage on 'Other Software Resources[1]', maintained by the Cochrane Informatics and Knowledge Management Department. This page presents a list of available tools, in addition to their own system (*RevMan*), that offer support for undertaking a systematic review (see Figure 4-1). The list, however, is short, misses many potentially helpful tools and places a focus on support for Cochrane reviews (a very particular type of systematic review in healthcare). The following tools are listed by Cochrane:

- ***GRADEpro*** (GRADE profiler) –software used to create summary of findings tables within Cochrane Reviews.
- ***DistillerSR*** – a web-based tool that supports study selection and data extraction activities in a systematic review.
- ***EPPI-Reviewer*** – a comprehensive web-based application with a focus on managing and analysing data within a systematic review (this tool is explored further in Chapter Five).

[1] https://tech.cochrane.org/revman/other-resources

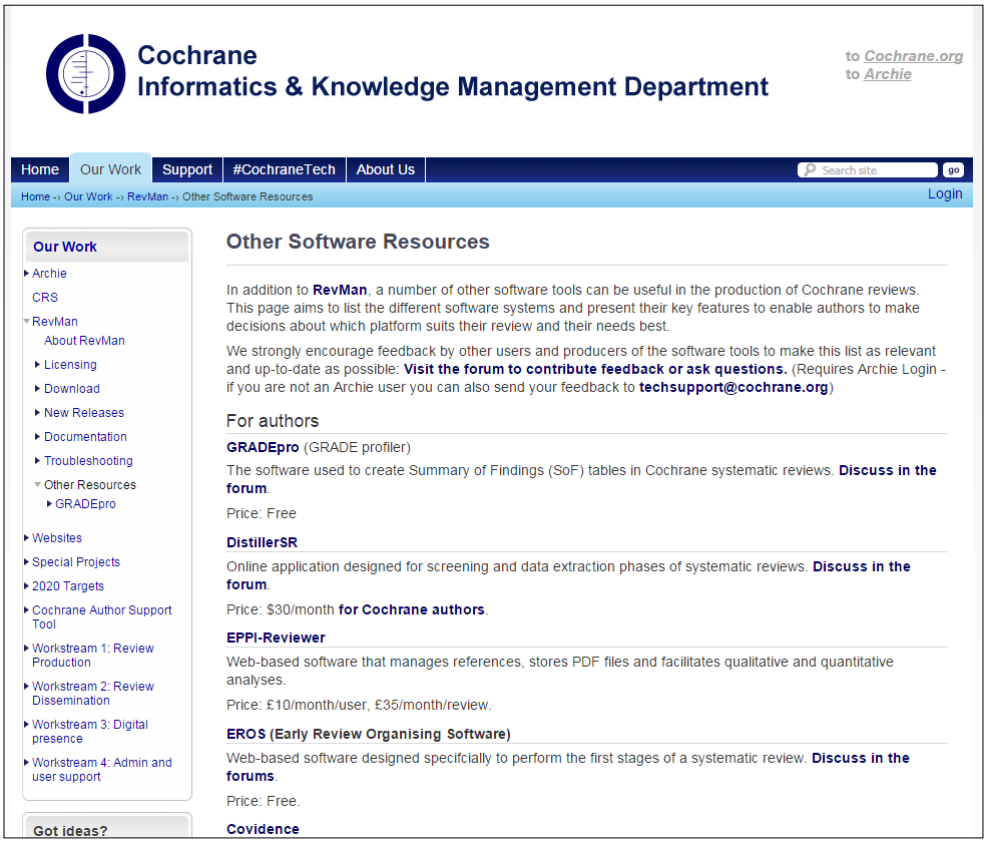


Figure 4-1. Screenshot of the information maintained by Cochrane on systematic review tools

- ***EROS*** (Early Review Organising Software) – web-based software designed to support the planning phase of a systematic review.
- ***Covidence*** – web-based tool that supports various stages of a systematic review.
- ***SRDR*** (Systematic Review Data Repository) – web-based tool that supports the extraction, management and search for systematic review data.
- ***SUMARI*** (System for the Unified Management, Assessment and Review of Information) – a suite of tools to support various aspects of the systematic review process.
- ***Rayyan*** – web-based application that, primarily, supports independent study selection and also includes a facility to solve disagreements.

Likewise, the EPPI-Centre also offer guidance on available tools that support systematic reviews (see Figure 4-2). Again, however, the list is short and very light on information. Tools listed by the EPPI-Centre include:

- ***EPPI-Reviewer*** – the EPPI-Centre's online systematic review support tool.
- ***MetaLight*** – a tool for performing, teaching and learning meta-analysis.
- ***RIS Export*** – search string conversion tool for importing search strings to various reference management systems.

This chapter presents the *Systematic Review (SR) Toolbox*[2]. *SR Toolbox* is a community-driven, searchable, online catalogue of tools that support the systematic review process, across multiple domains (see Figure 4-3). The resource aims to help reviewers find appropriate tools based on their particular needs. It uses a simple, yet flexible, classification system to classify tools based on how they provide support for the systematic review process. The classification criteria used to organise tools has been heavily influenced by the work reported in this thesis (e.g. the categories of tools used in the literature review and features used in the framework). *SR Toolbox* was developed using PHP, MySQL and JavaScript and uses social networking tools (specifically, Twitter) to manage its community. A short paper, which introduces the resource, has been published and presented at a leading empirical software engineering conference (Marshall & Brereton, 2015).

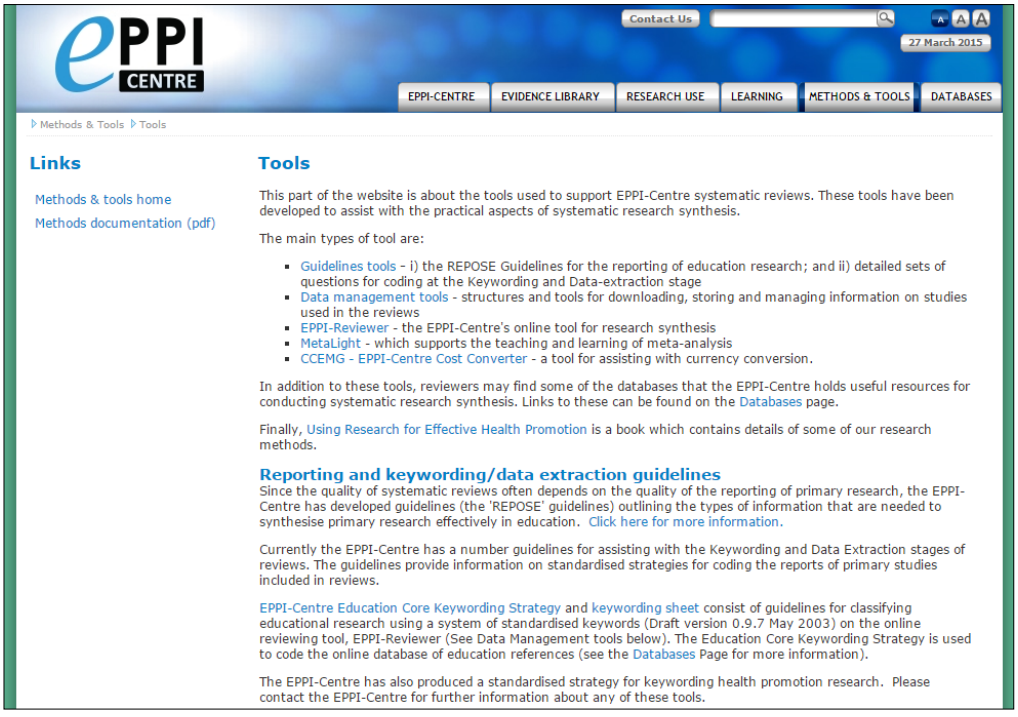


Figure 4-2. Screenshot of the information maintained by the EPPI-Centre on systematic review tools

[2] http://systematicreviewtools.com

4.2 Features

In this section, the key features of *SR Toolbox* are described; namely, executing a 'Quick Search', performing an 'Advanced Search' and submitting a new tool to the database.

4.2.1 Quick Search, Tool profile page and 'Other Tools'

Users can perform a simple keyword search (i.e. a Quick Search) to find tools. This type of search queries the 'tool_name' and 'tool_description' fields in the 'tool table' of the database and returns any matching results. A class diagram, which visualises the backend database, can be viewed in Figure 4-4. As shown in the example presented in Figure 4-5, a search for the term "Framework" has returned three 'software' tools; namely, *DBPedia* (a resource description framework), *Pimiento* (a framework for text mining) and *ReVis* (a visual text mining tool). If a user wishes to find out more about the tool, clicking the tool's name directs them to a dynamically generated profile page. This area (as shown in Figure 4-6) provides more information about the tool and some useful links. Details includes:

- *Description* - a short description of the tool.
- *More info* - a link to the tool's homepage.
- *Papers* - links to any relevant publications that focus on the tool.
- *Discipline* - the domain in which the tool is focusing support, the underlying approach (or technology) associated with the tool.
- *Cost* - the cost (or price) of the tool.
- *Supports* - the stages of a systematic review (or features) the tool supports.

Returning to the example presented in Figure 4-5, a search for "Framework" has also returned two 'Other tools'; namely, *Quality in Qualitative Evaluation: A Framework for Assessing Research Evidence* and *SQUIRE* (Standards for Quality Improvement Reporting Excellence). Depending on the context, the term "tool" can be considered quite vague. In software engineering, for example, it is quite common to interpret "tool" as a piece of software or something technology- related.

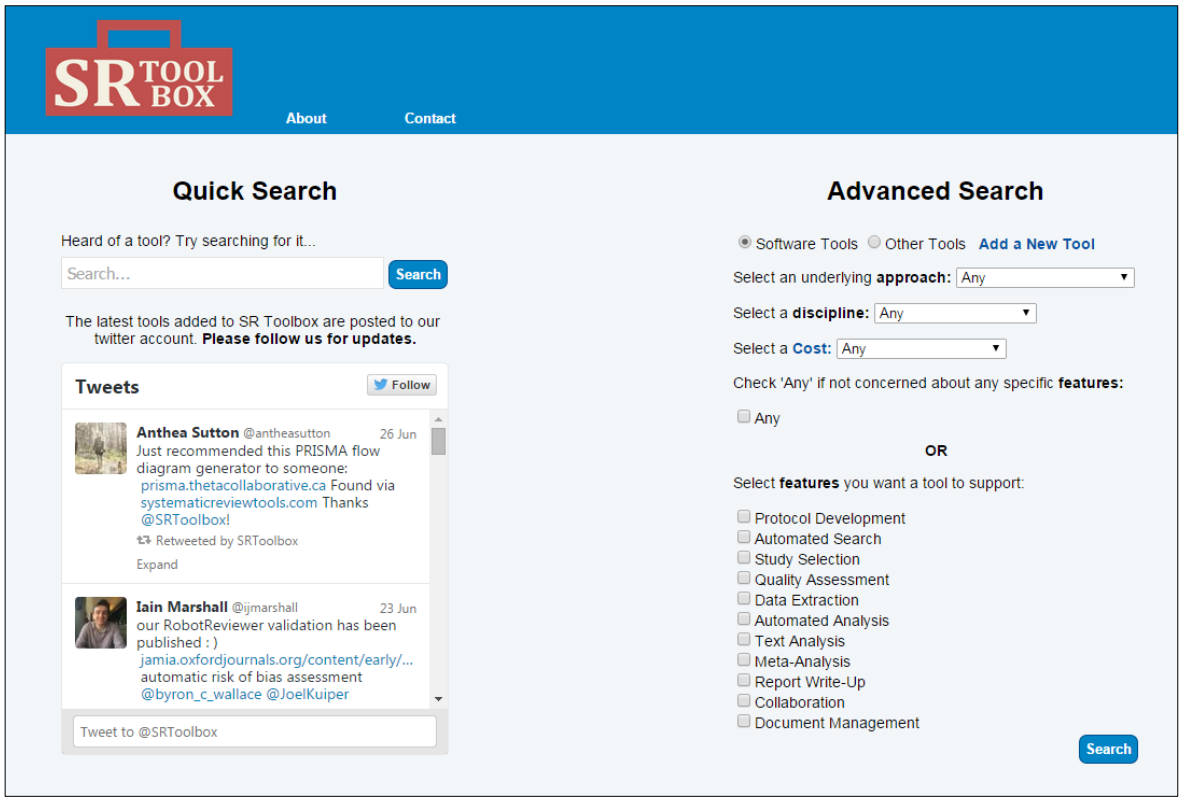


Figure 4-3. Screenshot of the *Systematic Review Toolbox* homepage

In other domains, however, "tool" can have a variety of meanings, particularly when concerning support for systematic reviews. In the early weeks after the launch of *SR Toolbox*, there were a number of requests for the resource to hold more 'paper-based' tools (i.e. checklists, guidelines etc.). Therefore, although the focus of *SR Toolbox* is still on identifying software tools to support systematic reviews, other tools or support mechanisms such as checklists, guidelines and reporting standards can also be found.

4.2.2 Advanced Search

Users may also perform an Advanced Search for tools (see Figure 4-7). Here, the user can specify what kind of tool they require to support their systematic review, based on their particular needs, using a number of different selection criteria to tailor their query. These criteria include:

1. **Type of tool** – located at the top of the Advanced Search form, the user can select whether they would prefer to search for software tools (selected by default) or 'Other Tools' (these types of tools are discussed in Section 4.2.1). Selecting the 'Other Tools' radio button

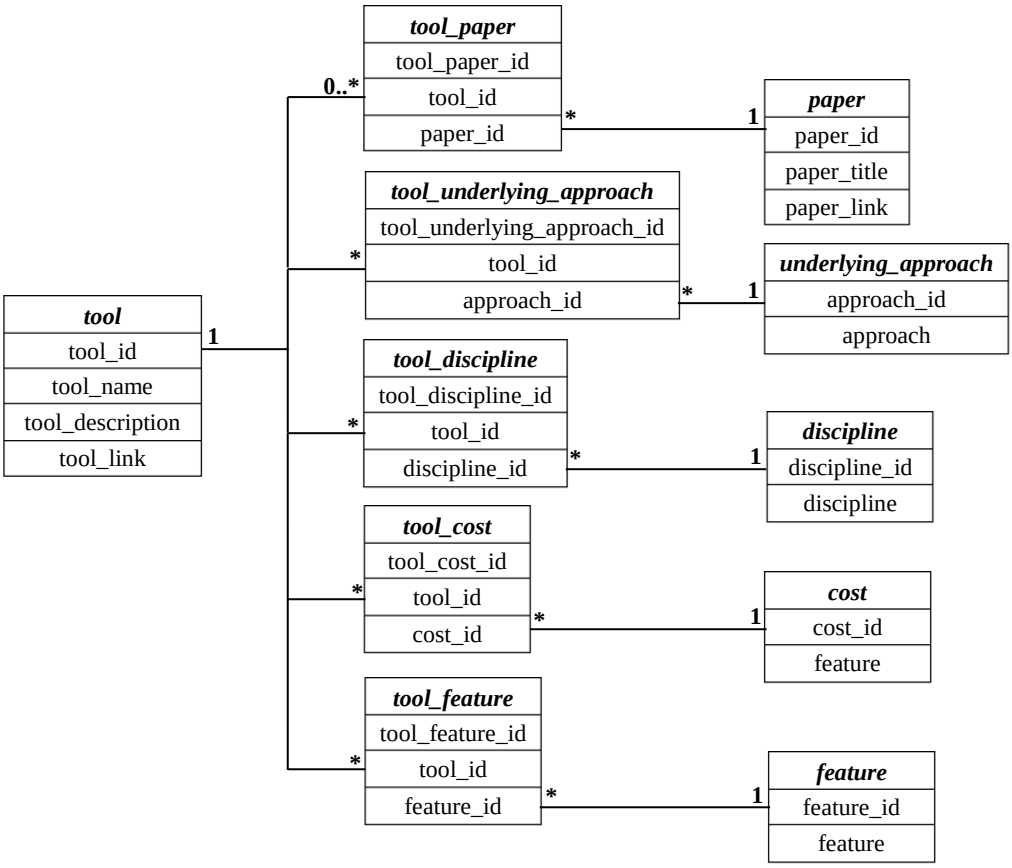


Figure 4-4. Class diagram visualising the database behind *Systematic Review Toolbox*

presents a new query form (see Figure 4-8). This form lets users search for paper-based tools, such as guidelines, quality checklists and reporting standards across healthcare, social science and software engineering domains.

2. **Underlying approach** - users can select a particular underlying approach (or technology) associated with a tool to filter their search. As of July 2015, the underlying approaches available are summarised in Table 4-1. These selection criteria have been influenced by the underlying approaches identified and used to classify tools in Chapter Two's literature review (see Table 2-3). This particular criteria set might be useful for users with particular knowledge or expertise about a certain kind of associated technology. Where a user is not concerned about the underlying approach (or doesn't mind), they can select 'Any'. Selecting 'Any' will include all underlying approaches in a search query.

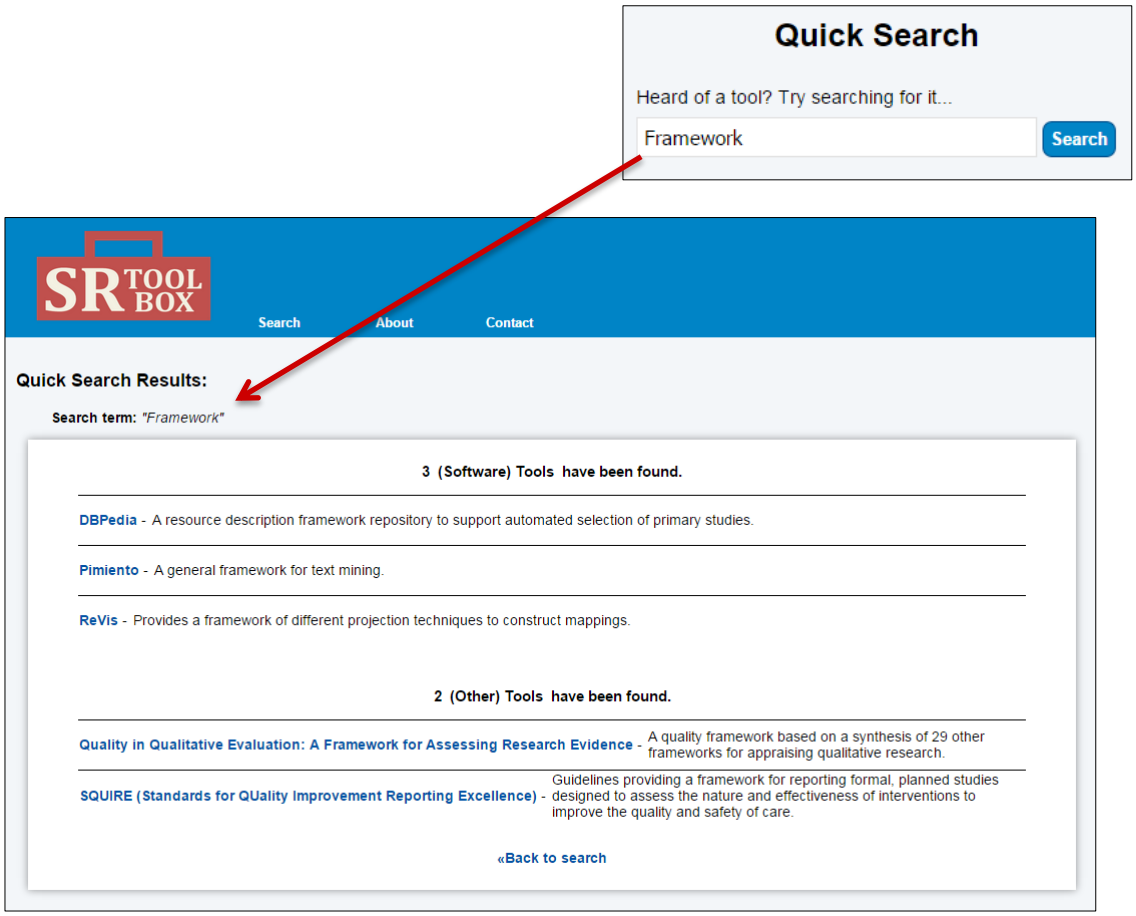


Figure 4-5. Screenshot of an example Quick Search for tools

Underlying approach	Comments
Visualisation	-
Text mining	-
Visual text mining	*A particular type of approach which combines elements of both visualisation and text mining. Such tools have been investigated by Malheiros, 2007 and Felizardo et al., 2011.*
Whole process	*Tools that aim to provide support for the whole systematic review process (or at least the majority of stages). Such tools were evaluated in the feature analysis reported in Chapter Three.*
Ontology	-
Search	-
Machine learning	-
Data mining	-
Visual data mining	*Tools combining elements of visualisation and data mining. This approach has been investigated by Felizardo et al., 2012.*
Reference management	-
Other	-

Table 4-1 Advanced search criteria for underlying approaches of tools (as of July 2015)

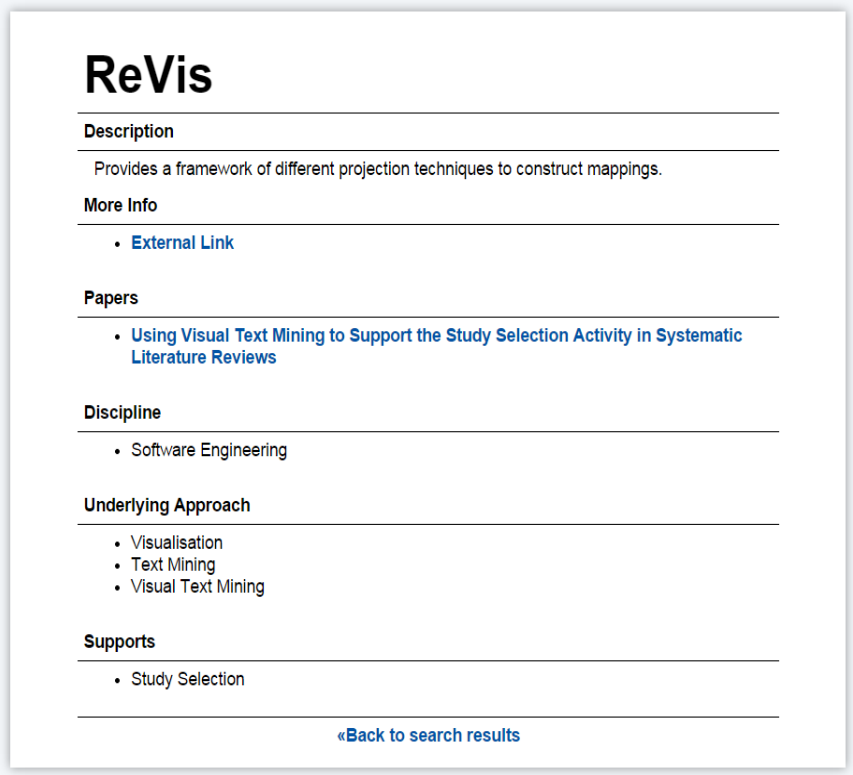


Figure 4-6. Screenshot of a tool's profile page

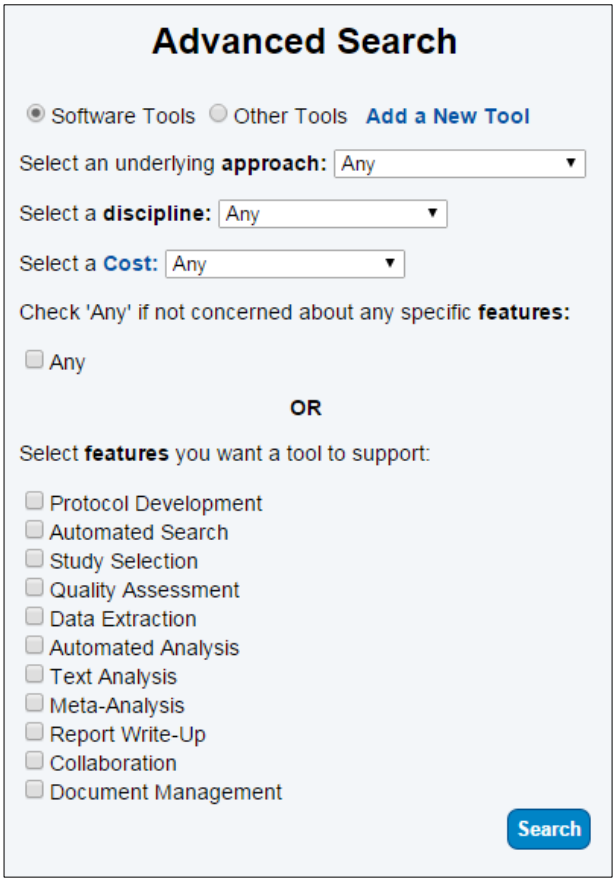


Figure 4-7. Screenshot of the form used to perform an Advanced Search for tools

3. **Domain** – users can specify the target domain in which they require support from a tool. As of July 2015, *SR Toolbox* includes tools that support systematic reviews in healthcare, areas of social science and software engineering. However, as mentioned in Section 1.1.5 many issues that arise when undertaking a systematic review are common across multiple domains. Work undertaken (and reported over Chapter Five and Chapter Six), for example, suggests that many problems relating to systematic reviews faced in certain disciplines are similar to those faced by researchers in other domains as well (Marshall *et al.*, 2015). Therefore, many tools, which might be considered domain-specific, may also be helpful to researchers in other fields. Where this is believed to be the case, tools have been classified appropriately as providing multi-domain support. Where users are not concerned about a particular domain, they may also select the 'Any' option. Doing so includes all domains when searching for tools.
4. **Cost** – users can select a particular cost for a tool to further refine their search query. As of July 2015, there are four different options for this criteria. Tools can be considered (and classified) as 'Completely Free' – i.e. the full version of the tool requires no financial payment to use, 'Free Version Available' – i.e. a free version of the tool exists as well as a paid version (selecting this option also includes 'Completely Free' tools in the search results), 'Free Trial' – i.e. a free trial of the tool can be downloaded and used before purchasing the full version and 'Payment Required' – i.e. a financial payment is required to use any version of the tool.

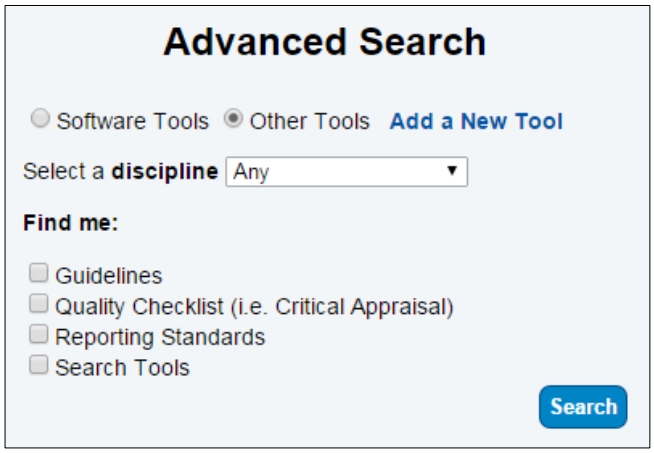


Figure 4-8. Screenshot of the form used to search for 'Other Tools' (i.e. paper-based tools)

5. **Feature** – users are able to select what aspects (or features) of the systematic review process they want supported by a tool. Features supported by tools, included in *SR Toolbox*, include protocol development, automated search, study selection, quality assessment, data extraction, automated analysis, text analysis, meta-analysis, report write-up, collaboration and document management. This criteria was influenced by (a previous version of) a set of features developed to evaluate a selection of candidate systematic review support tools (this work is reported in Chapter Three). These features, included as part of an early version of an evaluation framework for an overall support tool for systematic reviews in software engineering, have since been refined based on work reported in Chapters Five and Six When applying this criteria, it is important to note that feature selections stack. If, for example, a user checks the, 'Study Selection', 'Quality Assessment' and 'Data Extraction' boxes, only tools which include support for *all* of these features, will be retrieved from the database. In the case of this example, *SR Toolbox* returns six tools which fulfil the search criteria (*Covidence*, *JBI-SUMARI*, *SESRA*, *SLR-Tool*, *SLuRp* and *SRDB.PRO*) as shown in Figure 4-9. Where users are not concerned about particular features from a tool, selecting the 'Any' box disables the other feature checkboxes and performs a search for tools with any combination of features.

Although performing a Quick Search for tools may be useful, it is expected that most users will use the Advanced Search option to find appropriate tools. This feature (i.e. Advanced Search) is considered the most novel aspect of *SR Toolbox*. It is the first resource to provide such a service to researchers. Figure 4-9 visualises a complete example of an Advanced Search.

4.2.3 Adding a new tool

Since the launch of *SR Toolbox* in May 2014, several users have been in contact with suggestions for new functionality. One of the most frequently requested updates was the ability for users to add their own tools. Such tools could either be those that the user had developed, or those that the user had heard about (or experienced) which were not currently stored in the database. Initially, users

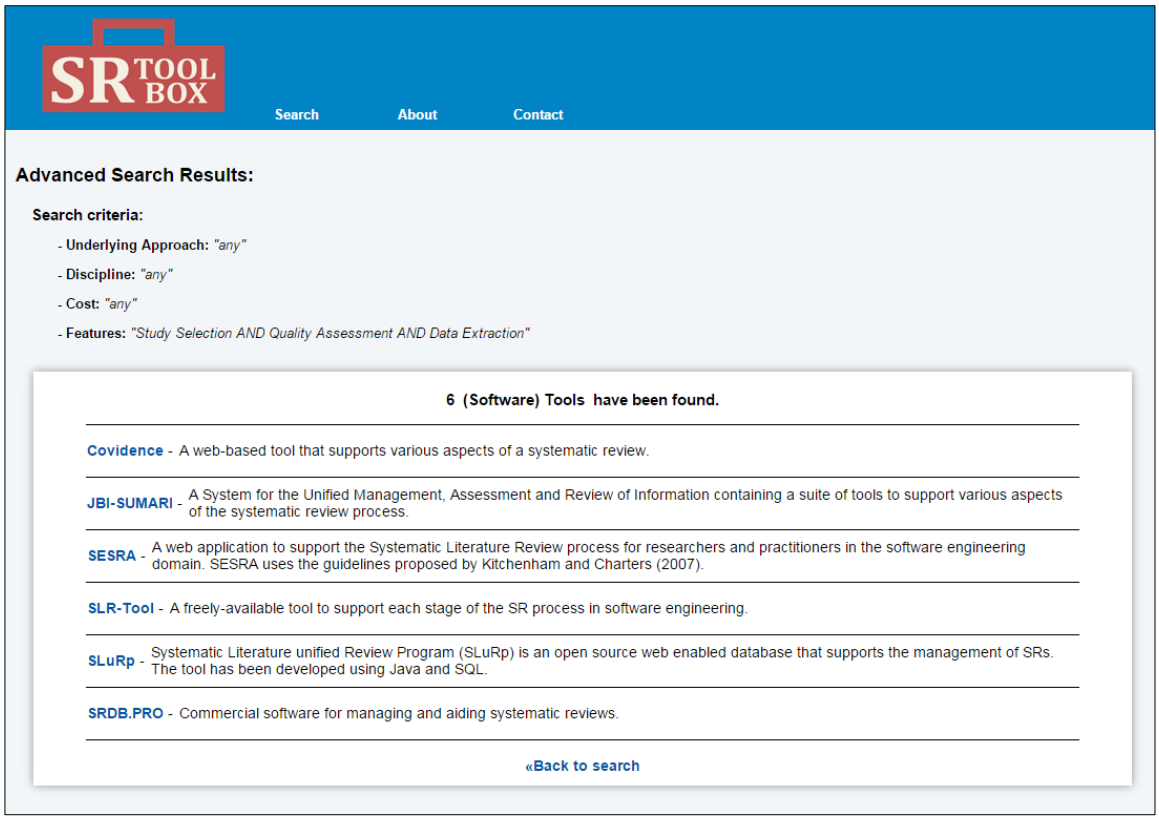


Figure 4-9. Screenshot of an example Advanced Search for tools (performed in July 2015)

were encouraged to propose new tools to add by contacting the site author by email or interaction with the *SR Toolbox* twitter account. However, the ability to submit new tools for addition is now a key feature of the resource.

Under the Advanced Search heading (see Figure 4-7) is a link to 'Add a new Tool.' Clicking this link directs the user to a submission form for completion. Within the form, presented in Figure 4-10, users can provide details about the tool they wish to submit. Details to add by the user include the name of the tool, the domain (or discipline) in which the tool focuses its support for systematic reviews (i.e. healthcare, social sciences, software engineering etc.), a short description of the tool (along with any relevant links which provide further information), the underlying approach (or technology) associated with the tool, the cost (or price) of the tool and, finally, features or aspects of the systematic review process that the tool supports. The user may also, optionally, provide their contact details, along with any comments, feedback and suggestions. When directed to the 'Add a New Tool' page, the form for adding software tools to the database, is selected by default. However, users can also submit 'Other Tools' (i.e. checklists, guidelines reporting standards etc.)

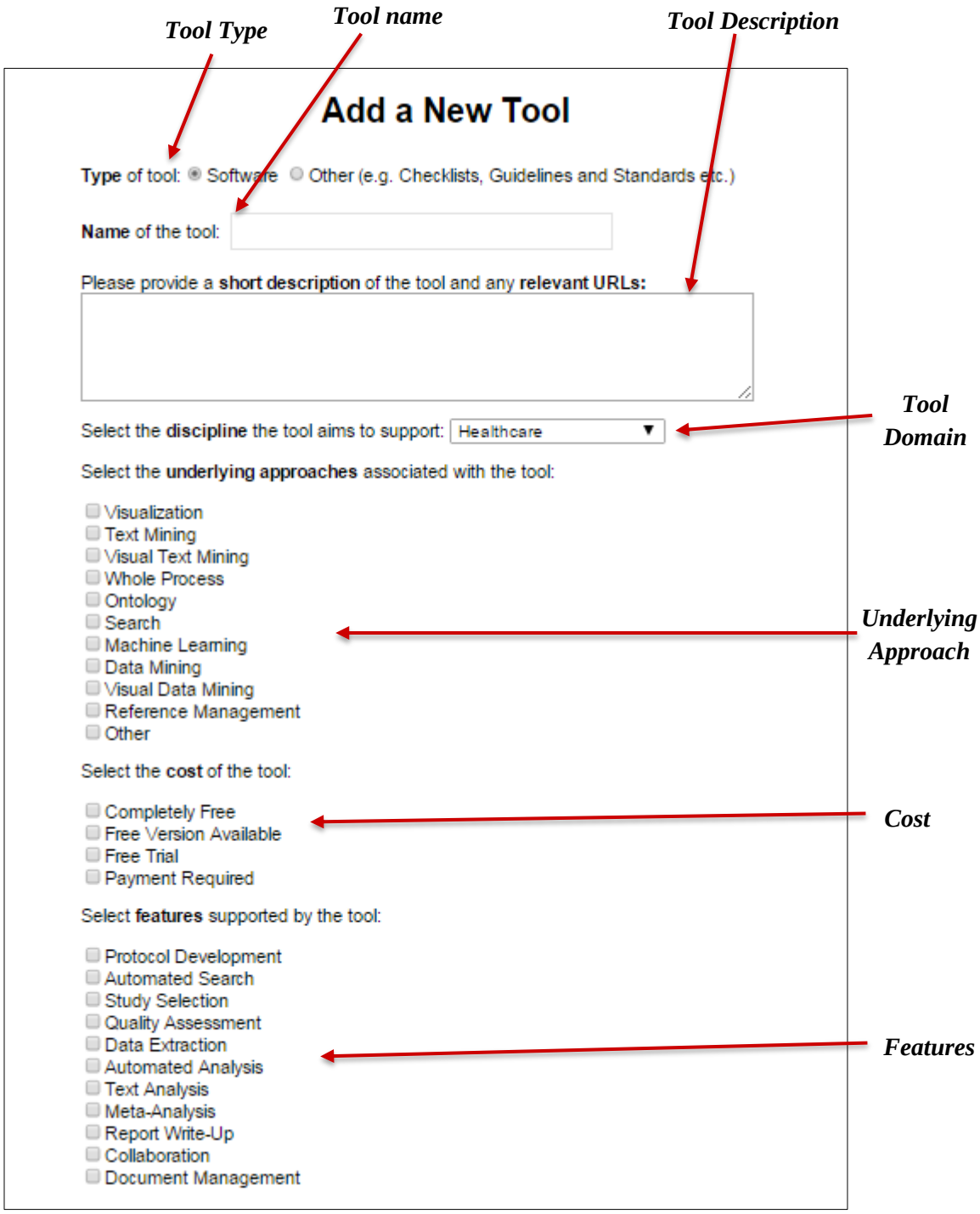


Figure 4-10. Annotated screenshot of the Add a New (Software) Tool submission form

as well. Clicking the 'Other Tools' radio button at the top of this page switches to an appropriate form for these tools to be added. As shown in Figure 4-11, users can add the name of the tool, domain, a short description (with links) and, as before, any optional contact details. On submission of a new tool, users are presented with a confirmation message informing them that the tool information has been received (see Figure 4-12). It is important to note that this information is not

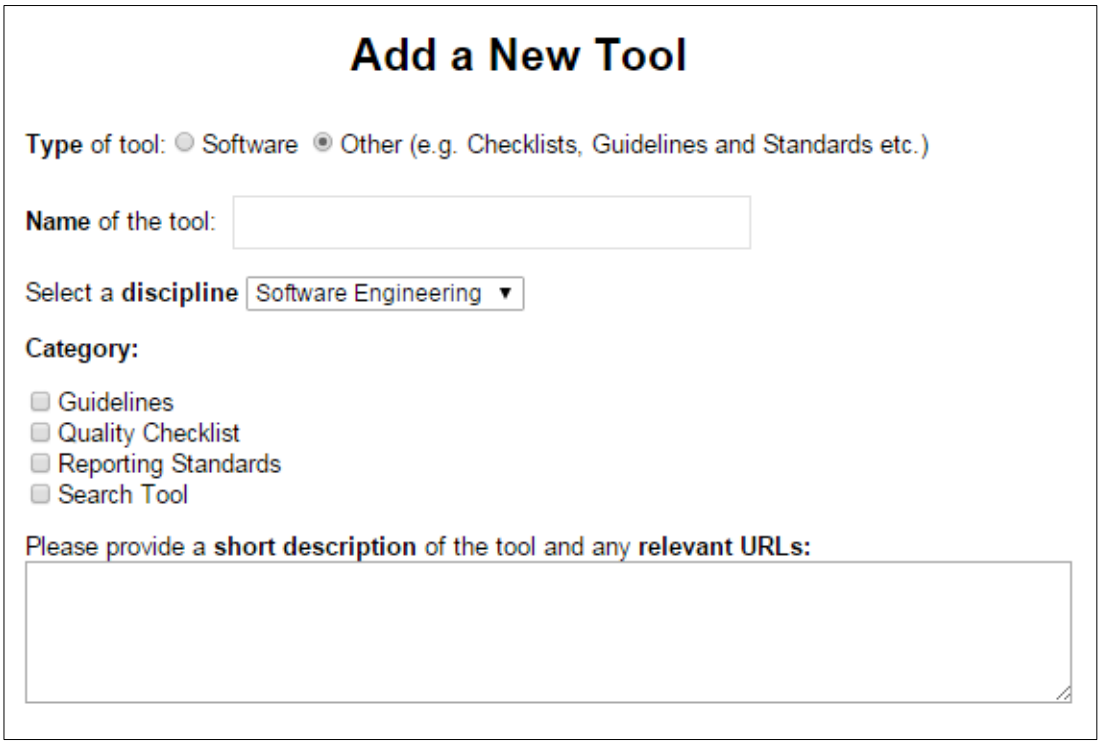


Figure 4-11. Screenshot of the Add a New (Other) Tool submission form

added to the database immediately. Instead, the data is emailed to the site author for review and, if suitable, added to the database. Once a new tool is added, the *SR Toolbox* twitter account ('@SRToolbox') is updated (see Figure 4-13). Users are encouraged to 'follow' this account for notifications on new tools and any relevant information. An embedded twitter feed can be found on the site's homepage (see Figure 4-3) allowing users to view recent tool additions without needing to 'follow' the account.

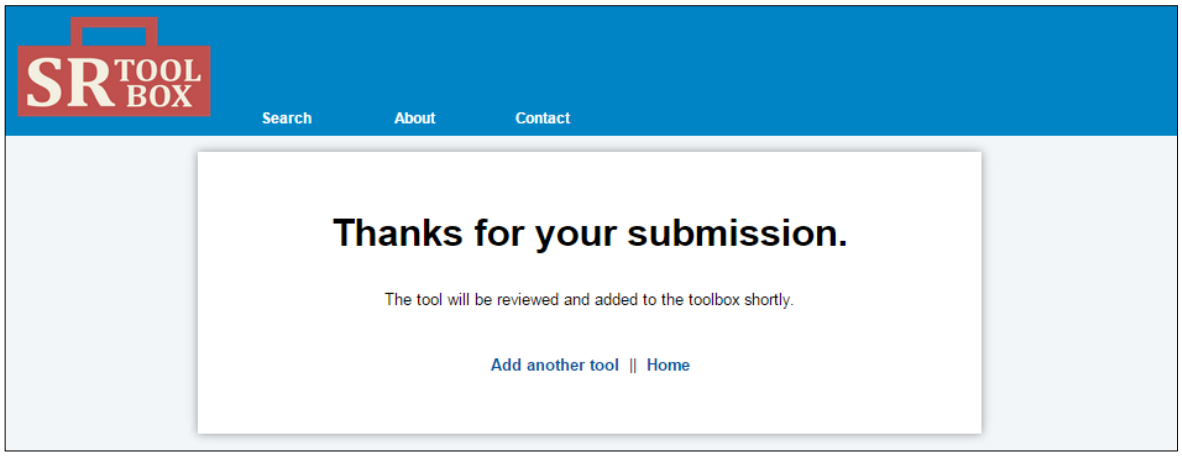


Figure 4-12. Screenshot of the confirmation message provided for adding a new tool

4.3 Conclusions, Impact and Future Development

This chapter has introduced SR Toolbox; a resource for reviewers to identify appropriate tools to support their systematic reviews based on their particular needs. This resource was developed in response to a lack of easily accessible information on currently available tools to support systematic reviews across multiple domains.

As of July 2015, the database holds a total of 112 tools. This includes 83 software tools and 29 other tools (i.e. checklists, guidelines and reporting standards). To-date, the most common underlying approach associated with tools stored in the database is reference management (27) followed by, text mining (17) and visualisation (14). Data (i.e. numerical data) mining tools are the least common (2). This information can be found in Table 4-2. As shown in Table 4-3, the most common systematic review steps supported by tools are document management (37), data analysis (23) and study selection (20). Least common are tools that provide support for protocol development (8), quality assessment (11) and meta-analysis (12). Regarding other paper-based tools (i.e. 'Other Tools'), the most common are quality checklists (22). The least common are guidelines (4).

Since becoming live in May 2014, the resource has been received positively by the research community and is actively used by many research staff and students across multiple domains.

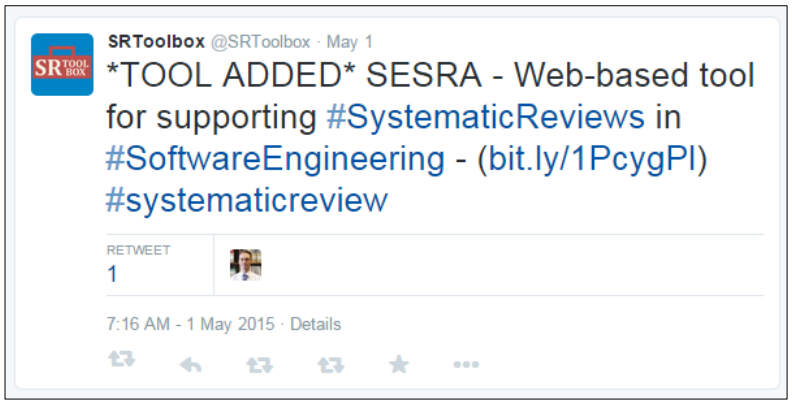


Figure 4-13. Screenshot of the notification made by the *Systematic Review Toolbox* twitter account (@SRToolbox) that a new tool has been added to the database

Within healthcare, *SR Toolbox* has been cited in the 2014 CochraneTech symposium editorial[3] (Elliott *et al.*, 2014). CochraneTech is an annual event targeted for those interested in the application and integration of existing and emerging technologies in the dissemination of systematic reviews (particularly, Cochrane reviews) and evidence synthesis in healthcare. The event is part of the larger annual Cochrane Colloquium. The theme for CochraneTech 2014 was the "Future of Review Production". The editorial discusses the current limitations of tool support for systematic reviews and in particular, focuses on the inefficient way in which technology is currently adopted by reviewers. Elliott *et al.* (2014). states:

"Review authors commonly conduct the majority of their work on a patchwork of general software products poorly adapted to their needs."

The authors mention in the editorial that "*a diverse set of technologies can be used to produce a Cochrane Review"* and then provide a link to *SR Toolbox*, as an example of where such tools can be found. Furthermore, the webpage on 'Other Software Resources' (see Figure 4-1), maintained by Cochrane, now includes a statement (at the top of the page) informing visitors that the page is no longer updated. Instead, they recommend (and link to) *SR Toolbox* for finding up-to-date information on systematic review tools (see Figure 4-14).

Elsewhere, *SR Toolbox* was presented to the empirical software engineering research community at the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE

[3] http://www.cochranelibrary.com/editorial/10.1002/14651858.ED000091

Underlying Approach	Total
Reference management	27
Text mining	17
Visualization	14
Other	13
Whole process	11
Search	11
Machine learning	4
Ontology	4
Visual text mining	3
Data mining	2
Visual data mining	2

Table 4-2. Number of tools stored in *Systematic Review Toolbox* classified by underlying approach (as of July 2015)

2015). The tool is also referenced in an upcoming book on evidence-based software engineering, due for release at the end of 2015 (Kitchenham *et al.*, 2015). Lastly, *SR Toolbox* was presented in April 2015 at the School of Health and Related Research (ScHARR), University of Sheffield. The resource was well received by members of the audience[4].

As mentioned in Section 4.2.3, *SR Toolbox* maintains an active twitter account using the handle @SRToolbox[5]. This account is used, primarily, to announce new tool additions (and user submissions) to the database, along with any new features developed for the resource.

Feature	Total
Document management	37
Automated analysis	23
Study selection	20
Text analysis	20
Data extraction	18
Report write-up	17
Automated search	16
Collaboration	16
Meta-analysis	12
Quality assessment	11
Protocol development	8

Table 4-3. Number of tools stored in *Systematic Review Toolbox* classified by feature (as of July 2015)

[4] http://scharrheds.blogspot.co.uk/2015/05/systematic-reviews-toolbox.html
[5] https://twitter.com/srtoolbox

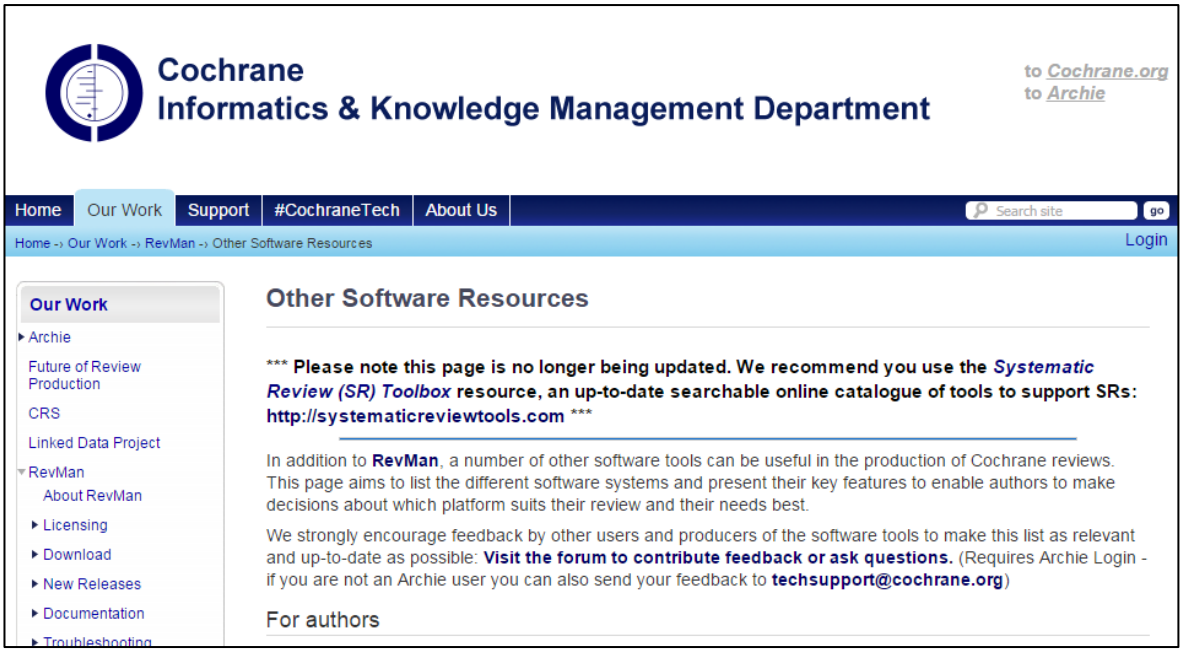


Figure 4-14. Screenshot of the message listed on Cochrane's page for providing information about tools to support systematic reviews (as of July 2015)

@SRToolbox has gathered a following of, to-date (July 2015), 120 'followers' who have subscribed to be notified when a new update is posted. These include senior researchers, academics and research students from a variety of domains. Many followers have expressed positive feedback about the resource using social media and have shared discoveries of new tools with fellow researchers within the community (see Figure 4-15).

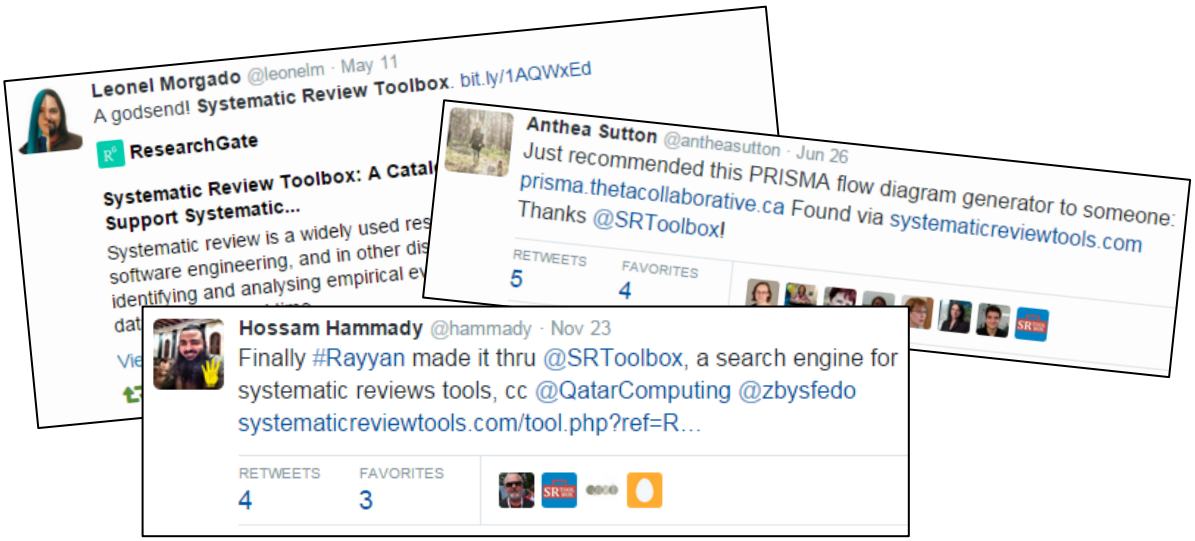


Figure 4-15. Interaction with the *Systematic Review Toolbox* over social media

Based on user traffic statistical data (generated by Google Analytics), as of July 2015, *SR Toolbox* averages between 200 - 400 unique visits a month. As future development for *SR Toolbox*, it is anticipated that the database will continue to be populated with new tools, particularly following submissions from members of the systematic review (and tool developer) community. Existing functionality will also be refined along with the development of new features. One risk is that as new tools emerge and existing systems evolve, maintaining an updated list of tools will become increasingly difficult. Therefore, to ensure the longevity of *SR Toolbox*, engagement and support from the systematic review community is essential to its maintenance.

Chapter Five

Cross-Domain Survey: Background and Study Design

This chapter presents the background and design of an interview-based survey. An introduction to the study, which aimed to explore the scope and practice of tool support for systematic reviews in domains outside of software engineering, is provided. This is followed by a discussion on the appropriateness of the survey methodology and using interviews for data collection. An overview of the structure and content of the survey, including the procedures used for the interviews and selection of participants, is provided.

5.1 Introduction

The literature review in Chapter Two identified a number of tools developed and used to support systematic reviews in software engineering (Marshall & Brereton, 2013). Most of the tools found provided support for individual stages or particular aspects of the systematic review process. A small selection, however, aimed to support the overall process (i.e. the majority of stages in a systematic review). These tools had received limited evaluation and provided motivation for a full independent evaluation.

Reported in Chapter Three, a feature analysis, which compared and independently evaluated four tools aiming to support the majority of stages of a systematic review in software engineering, was performed. The study served to investigate the feasibility of an evaluation framework for tools which support the whole systematic review process. The framework is comprised of a set of features, weightings (i.e. levels of importance) and scoring instruments. As discussed in Section 3.3.2, the features were generated based on the following criteria:

- Experiences of performing systematic reviews in software engineering reported in the literature and relevant research regarding tool support within the domain.
- The findings of the literature review reported in Chapter Two.
- Generic factors from the literature about software/tool evaluation (see Section 3.2).
- Discussion between members of the evaluation team who performed the feature analysis.

As well as providing valuable new insights into tools that support the whole systematic review process, the feature analysis also illustrated the feasibility of the framework. This led to some refinements to the features, which are discussed in Section 3.5.2 and presented in Table 5-1. Following the study, two possibilities for future work were identified. The first was to:

1. Circulate the features (and their importance levels) within the evidence-based software engineering community for feedback and suggestions.

id	Feature set	id	Feature	Importance weighting
F1	Economic	F1-F01	No financial payment	Highly Desirable
		F1-F02	Maintenance	Highly Desirable
F2	Ease of introduction and setup	F2-F01	Simple installation and setup.	Highly Desirable
		F2-F02	The tool is self-contained.	Highly Desirable
F3	Systematic reviews activity support	F3-F01	Protocol development	Desirable
		F3-F02	Protocol validation	Desirable
		F3-F03	Supports automated searches	Highly Desirable
		F3-F04	Study selection and validation	Highly Desirable
		F3-F05	Quality assessment and validation	Highly Desirable
		F3-F06	Data extraction and validation	Highly Desirable
		F3-F07	Data synthesis	Highly Desirable
		F3-F08	Text analysis	Nice-to-have
		F3-F09	Meta-analysis	Nice-to-have
		F3-F10	Report development	Nice-to-have
		F3-F11	Report validation	Nice-to-have
F4	Process management	F4-F01	Support for multiple users	Mandatory
		F4-F02	Document management	Mandatory
		F4-F03	Security	Desirable
		F4-F04	Management of roles	Highly Desirable
		F4-F05	Re-use of data from past projects	*(see Section 3.5.2)*

Table 5-1. Features and importance levels from version 1.1 of the evaluation framework

In this case, the latest version of the evaluation framework (version 1.1) would be distributed to members of the evidence-based software engineering community. Their feedback, suggestions and general expertise would then be used to help generate an updated version (i.e. version 1.2). Whilst this direction may, at first, appear to be most appropriate, it was decided against. Brown and Wallnau examined the problems of evaluating the likely impact of a new software technology (Brown & Wallnau, 1996). In this work, the authors state that:

"It is often useful to extend the model beyond the immediate technologies of interest in order to obtain a deeper understanding of the technology being evaluated"

In the context of this project, the goal is still to develop, refine and validate an evaluation framework for an overall tool to support systematic reviews within software engineering. As

discussed in Chapter One, however, the systematic review methodology is applied in many other domains, as well as software engineering. Tools are being developed and used to support systematic reviews in such areas as healthcare and social science (see Chapter One and Chapter Four). Furthermore, the adoption of the systematic review methodology within software engineering is still, when compared with other domains, relatively recent. Therefore, a second route for future work was proposed:

2. Explore systematic review tools (and their use) within other domains where systematic reviews are also undertaken.

After completing the feature analysis, work undertaken (so-far) to develop the evaluation framework had focused on investigating the systematic review process (and technology to support it) within an evidence-based software engineering context. Investigating tool support in other domains where systematic reviews are more established, would, therefore, be useful, in order to obtain a deeper understanding of the technology (Brown & Wallnau, 1996). It is worth noting, however, that the feature analysis was published and disseminated at a leading empirical software engineering conference (Marshall *et al.*, 2014). This at least partially achieves the circulation of the framework (version 1.0) within the software engineering community. Furthermore, activities to further validate the evaluation framework, reported in Chapter Seven, return to a software engineering context (see Section 7.3.1.3 and 7.3.2).

5.1.1 Study aims and objectives

This chapter reports the background and design of a study which aims to explore the experiences and opinions of systematic reviewers in domains other than software engineering, with a particular focus on their use of and views about support tools. The study takes the form of a survey and uses semi-structured interviews as the data collection technique. Three aims for this study were defined:

1. To explore what tools are currently available and used to support systematic reviews in other domains; specifically, healthcare and areas of social science.

2. To identify what participants consider to be the most important characteristics (or features) of tools to support systematic reviews.
3. To compare the features and importance levels identified in the survey with those forming part of the evaluation framework for tools which support the whole systematic review process in software engineering.

This chapter is organised as follows. Section 5.2 presents the background to and appropriateness of the research methodology (survey) and data collection technique (interview) used for this study. Details of the study design, including the selection of participants, structure, content, ethical approval of the survey instrument and pilot interview are outlined in Section 5.3.

5.2 Background

The study reported in this and the next chapter took the form of a survey and used semi-structured interviews for data collection. In this section, the background and appropriateness of undertaking an interview-based survey, is described.

5.2.1 Survey methodology

Survey research aims to obtain the same kinds of data, from a particular group of people, in a standardised and systematic way (Oates, 2006). The method attempts to provide a 'snapshot' of a situation at a particular point in time (Robson, 2011; Rea & Parker, 2014). Surveys as a research method have gained considerable (and increasing) credibility from their widespread use throughout academia and industry (Rea & Parker, 2014). The methodology has been used to support research in a variety of fields including education (Schultze *et al.*, 2011), healthcare (Howell & Caplan, 2015) and software engineering (Preuveneers & Novais, 2012). The ultimate goal of survey research is to attempt to generalise about a large population by investigating only a small portion of that population (Rea & Parker, 2014).

A survey is particularly suited for supporting research where the aims and objectives are, primarily, exploratory in nature (Rea & Parker, 2014). The method can be used as both a primary or secondary data collection technique (Robson, 2011) and is suitable for gathering self-reported quantitative and qualitative data (Lethbridge *et al.*, 2005). As described by Rea and Parker, surveys tend to collect one (or often a combination) of three types of data (Rea & Parker, 2014):

- *Descriptive* – basic information or characteristics, which enable the researcher to better understand the participant and the larger population.
- *Behaviour* – behaviourally oriented information about a participant; for example, frequency or patterns of use regarding a particular topic.
- *Attitudinal* – the participant's attitudes, opinions and preferences about a particular issue, topic or circumstance.

The majority of surveys are carried out for descriptive purposes (Robson, 2011). However, as described by Robson, a survey can go beyond the descriptive. It can provide explanations of the phenomena investigated and the patterns of results obtained (Robson, 2011).

Surveys are praised for their transparency and accountability (Robson, 2011). The techniques and procedures implemented in the overall research design can be made easily accessible to others for assessment (Hakim, 2000). Transparency is important because the reliability and validity of a survey's results are largely based on how the survey was carried out (Robson, 2011). Therefore, a survey, regardless of the data collection techniques used, should adhere to specific procedures which are applied in a systematic manner.

The following eight stages of a 'typical' survey include[1]:

1. *Identifying the focus of (and requirements for) the survey* – a decision needs to be made as to why a survey is required and what it aims to investigate. Considering these factors will determine what type of survey is needed to achieve the goals of the investigation.
2. *Determine the data collection technique(s)* – the methods used to generate data for the survey must be selected. Typically, techniques include self-completion questionnaires, face-to-face interviews, telephone interviews or web-based approaches.
3. *Establishing the sampling frame* – based on the focus of the investigation, the relevant whole population, from which a sample is selected to include in the survey, must be defined.
4. *Selecting an appropriate sampling technique* – either a 'probability' or 'non-probability' sampling procedure (see Section 5.2.2), to obtain participants from the sampling frame, must be selected.
5. *Develop the survey instruments and procedures* – the survey instrument (e.g. questionnaire or interview procedure) must be designed to systematically collect data that is relevant to the focus of the investigation (identified in Stage One).

[1] These stages have been developed based on Oates (2006), Robson (2011) and Rea & Parker (2014).

6. *Piloting the survey instrument* – once developed, it is important to pilot the instrument under survey conditions, to identify any areas for refinement.
7. *Implementing the survey* – the survey can now be performed using the instrument designed and refined in Stages Five and Six. Care must be taken to ensure the privacy and confidentiality of participant responses, in accordance with the ethical approval process.
8. *Data analysis* – quantitative and/or qualitative data, collected from the survey, must be analysed using an appropriate analysis strategy.

Surveys provide an opportunity to reveal information about communities, by investigating individuals representing those communities in a relatively unbiased and rigorous manner (Rea & Parker, 2014). In Section 5.3, details of how the survey reported in this thesis aims to address each stage of this process are described.

5.2.2 Population sampling

Various sampling techniques are available to select participants for a survey. Techniques can be classified as either a form of *non-probability* or *probability* sampling.

Non-probability sampling obtains participants in a manner which does not provide individual participants an equal chance of selection (Oates, 2006). This type of sampling is commonly employed in small-scale surveys (Robson, 2011), where a truly representative sample is not feasible or necessary (Oates, 2006). Common examples of non-probability sampling include:

- **Convenience sampling**, which involves selecting participants based on factors of convenience; such as location, knowledge of the population and availability (Oates, 2006; Robson, 2011).
- **Snowball sampling**, where current participants are asked to identify any other potential participants (Robson, 2011).

Where the main objective is to provide strong generalisations to a wider population, *probability* sampling techniques are recommended (Oates, 2006; Robson, 2011). *Probability* sampling uses

appropriate techniques (e.g. systematic, cluster, multi-stage etc.) to randomly select participants from the sampling frame (Oates, 2006; Robson, 2011; Rea & Parker, 2014).

5.2.3 Data collection using interviews

Interviews are particularly suitable for collecting qualitative data and are commonly considered as the method of choice for researchers favouring a qualitative research approach (Robson, 2011; Potter & Hepburn, 2005). This is because the purpose of using interviews in empirical studies is often to collect data about phenomena, which cannot be easily obtained using quantitative measures (Hove & Anda, 2005). Interviews have been used effectively to support research in many fields; including healthcare (Zwaan *et al.*, 2010; Cegala, 2011; Powell *et al.*, 2011), areas of social science (Austin & Toth, 2011; Vanderlinde & van Braak, 2010; Parrish *et al.*, 2012) and software engineering (Babar & Zhang, 2009; Hoda *et al.*, 2010; Beecham *et al.*, 2013). Furthermore, interviews are commonly used to support studies which investigate the experience and impact of technologies within a particular community (Karlsson *et al.*, 2010; Sheih, 2012; Major *et al.*, 2014).

Performing an interview provides insight into a participant's world; their opinions, thoughts and feelings (Hove & Anda, 2005). They are a flexible and adaptable method for finding things out and can provide rich, highly illuminating data (Robson, 2011). Interviews lend themselves to being used in combination with other data collection methods. Although they can, however, be used as the primary (or only) approach within a study (Robson, 2011). Interviews can be conducted and structured in a variety of ways. Commonly, interviews are undertaken one-to-one, but they can also be performed in a group setting (i.e. focus group), over the telephone or using Voice over Internet Protocol (VOIP) facilities (e.g. FaceTime, Skype etc.). All interviews, however, regardless of the way in which they are performed, normally adhere to the following steps:

1. Ensure the research objectives, sample and sampling technique have been defined (see Stages One to Four of a 'typical' survey in Section 5.2.1).
2. Develop an initial version of the interview questions.

3. Pilot the questions under survey conditions.
4. Make any necessary refinements to the questions, based on lessons learned from the Pilot.
5. Carry out the main data collection (i.e. perform the interviews).
6. Code and analyse data in accordance with an appropriate analysis strategy.
7. Report the findings.

Furthermore, interviews are a highly suitable method when a researcher wants to explore the experiences and feelings of participants, which cannot be easily observed or described using pre-defined questionnaires (Oates, 2006). They are an effective method for dealing with topics in-depth and in detail (Oates, 2006). Other benefits of using interviews to support this research, as opposed to other relevant methods (such as questionnaires), are as follows:

- Interviews, particularly when organised as face-to-face, make it relatively easy to check whether participants fall within the population of interest. This is particularly difficult to control with questionnaires and is also slightly more difficult with telephone interviews as well (Oates, 2006; Robson, 2011).
- Interviews can help to ensure a higher quality of recorded response. With questionnaires, it is very difficult to assess how invested (in terms of engagement shown and attention given) participants are with the questions asked (Robson, 2011; Rea & Parker, 2014). Interviews are better equipped to make this assessment and can also, in some cases, allow for classification of participants on this basis (Robson, 2011).
- Interviews can help reduce the risk of bias relating to the researcher's preconceptions and allows for elaboration probes to encourage the participant to keep talking about a particular subject (Patton, 1990, Robson. 2011).

5.2.3.1 Types of interview

Interviews can vary greatly in their degree of structure from being almost fully structured, to allowing the interviewer much more freedom and flexibility (Robson, 2011). There are three main ways in which an interview can be organised:

1. **Fully** *(or "highly")* **structured interviews** – these types of interviews are undertaken where the main focus is to find relationships between constructs (Runeson & Host, 2009). A highly (or fully) structured interview tends to use a list of fixed predefined 'closed' questions using standardised wording, where participant responses are selected from a small list of alternatives (Robson, 2011; Rea & Parker, 2014). All questions are planned in advance and asked in the same order (Runeson & Host, 2009). In a structured interview, all participant responses should be able to be quantified (Hove & Anda, 2005).
2. **Semi-structured interviews** – using a less structured (i.e. semi-structured) approach is best when the focus is to investigate the experiences of participants both qualitatively and quantitatively (Runeson & Host, 2009). A semi-structured interview will, generally, have a list of questions (and topics) to help drive the interviews, but is less rigid in its overall approach and delivery. These types of interview allow the participant greater flexibility in their response. A semi-structured approach provides more opportunity for discussion and exploration of new topics, which arise throughout the data collection process (Robson, 2011). They offer the interviewer the potential to improvise by modifying ones enquiry to follow up an interesting response (Oates, 2006; Runeson & Host, 2009; Robson, 2011). Finally, a semi-structured approach to interviews is considered most appropriate for researchers who are very closely involved with the overall project (Robson, 2011).
3. **Unstructured interviews** - an interview can also be undertaken with no structure at all. A completely unstructured interview is only appropriate where the focus is to qualitatively investigate the experiences of participants (Runeson & Host, 2009). In an unstructured interview, questions are formulated as general concerns, interests and themes, which are

informally discussed by the interviewer and participant (Hove & Anda, 2005; Runeson & Host, 2009).

5.2.3.2 Question types

The type of interview (see Section 5.2.3.1) will determine the types of questions which can be asked. There are two main ways questions can be formulated:

1. **Closed questions** provide a pre-determined set of responses for participants to choose from when responding (Robson, 2011). They are often used to assess the degree of agreement by a participant or to obtain a rating or score (Robson, 2011). An advantage of closed questions is that the set of responses is standardised, therefore, facilitating quantitative comparison between participants (Rea & Parker, 2014). One disadvantage, however, is that participants might be unsure of the best answer or, alternatively, they might disagree with all possible options given (Rea & Parker, 2014).
2. **Open questions** provide no restrictions (other than the subject area) on the content or manner of the reply (Robson, 2011). Open questions are flexible, enable cooperation and rapport (between the interviewer and participant) and allow the interviewer to go into more depth about particular topics where appropriate (Robson, 2011). It is advised having at least some open-ended questions in any interview, as they are able to collect information that cannot be obtained by more specific (i.e. closed) questions (Lethbridge *et al.*, 2005). However, responses to open questions can be more difficult to analyse (Robson, 2011). In addition, the interview can become harder to control by the interviewer, as participants can sometimes digress and go off topic during their response (Robson, 2011).

Patton (2015) outlines six different types of open/closed questions:

1. *Behaviour/experience* – elicit information about experiences, behaviours and actions about what a participant does or has done.

2. *Opinion/value* – questions aimed at understanding what participants think about a particular topic or issue, based on their personal opinion, judgment and/or values.
3. *Feeling* – aims to elicit and understand emotional responses of participants about their experience and thoughts.
4. *Knowledge* – questions which identify factual information from the participant.
5. *Sensory* – aims to collect experiences of the participant's senses.
6. *Background/demographic* – questions which capture the characteristics of the participant.

5.2.3.3 Pilot interview

It is recommended to carry out at least one practice (i.e. pilot) interview before commencing data collection (Oates, 2006). A pilot interview involves a small-scale implementation of the interview instruments and procedures, which is conducted in order to assess the following critical factors (Hove & Anda, 2005; Oates, 2006; Rea & Parker, 2014):

- *Question clarity* – during the pilot, the interviewer may identify certain ambiguities in the questioning which confuse the participant. Where necessary, questions should be refined to ensure they are clearly understood by participants.
- *Question comprehensiveness* – a pilot interview can help assess whether the responses to questions generate a sufficient amount of data required to achieve the goals of the study.
- *Question acceptability* – assessing acceptability helps identify problems relating to excessive questioning (i.e. the length of particular questions), invasive questions and any ethically or morally sensitive queries.
- *Interview format/structure* – piloting the interview can help identify the need for any structural changes to questioning. It might be better, for example, to have certain questions asked earlier (or later) during an interview, to help improve the general flow.
- *Time* – a pilot interview helps provide an estimate of the time required to complete a full interview (from start to finish). This is an important detail to report when attracting

potential participants. Furthermore, an indication of time will help the researcher in estimating the time required to transcribe and analyse the interview data.

- *Interviewer training* – performing a pilot interview helps to train and prepare the interviewer. This is particularly important for a researcher who has little to no experience undertaking this type of research.

5.3 Study Design

In this section, details of an interview-based survey undertaken to explore the scope and practice of tool support for systematic reviews in other domains (outside of software engineering), are provided. The section is structured based on how the design addresses the eight stages of a 'typical' survey, as presented in Section 5.2.1.

5.3.1 Stage One – Identifying the focus of (and requirements for) the survey

In studies where the research aims are qualitative in nature, it is appropriate to rely on qualitative measures (Hove & Anda, 2005). Qualitative research focuses on investigating and understanding social and cultural phenomena in context (Myers & Avison, 1997) and is appropriate where the purpose is to explore a topic and obtain an overview of a complex area (Robson, 2011).

As shown in Section 5.1.1, the aims of this study were:

1. To explore what tools are currently available and used to support systematic reviews in other domains; specifically, healthcare and areas of social science.
2. To identify what participants consider to be the most important characteristics (or features) of tools to support systematic reviews.
3. To compare the features and importance levels identified in the survey with those forming part of the evaluation framework for tools which support the whole systematic review process in software engineering.

Therefore, since the focus of this study was to explore the experiences and opinions of systematic reviewers, its aims were considered as being, primarily, qualitative in nature. Therefore, to achieve the goals of the study, a survey was considered appropriate to be undertaken. This is because surveys are particularly suited to support qualitative, exploratory research (see Section 5.2.1).

5.3.2 Stage Two – Determining the data collection technique

Based on the characteristics of the study, face-to-face, semi-structured interviews were selected as the sole data collection technique. Semi-structured interviews are particularly suited for collecting both qualitative and quantitative data and combine elements from both highly-structured and unstructured approaches (see Section 5.2.3.1). This flexibility was needed in order to fully achieve aims of the study; particularly with regards to Aim Three (see Section 5.1.1) where responses needed to be quantified. Furthermore, a semi-structured approach helps accommodate a level of structure and standardisation to the interview, whilst still maintaining an exploratory overall feel. Questionnaires; another common data collection technique, were considered for this survey, but not used. This is because, unlike self-administered questionnaires, interviews can allow for considerable freedom in the sequencing of questions and in the amount of time and attention given to particular topics (Robson, 2011). Other points favouring the suitability of interviews, as opposed to other data collection techniques, are considered in Section 5.2.3.

5.3.3 Stage Three – Establishing the sampling frame

The aim of this study was to explore the experiences and opinions of systematic reviewers in domains other than software engineering, with a particular focus on their use of and views about support tools (see Section 5.1.1). The target population for this survey consisted of researchers with knowledge and experience of the systematic review methodology, in areas of healthcare and social science. This sampling frame was selected because:

1. As discussed in Section 5.1, systematic review is an established and widely accepted research methodology in both of these areas.

2. The systematic review process used in software engineering is heavily influenced by evidence-based practices used in these domains (Budgen *et al.*, 2008). As a result, many of the stages of a systematic review (and their challenges) are similar. Therefore, this study may also serve to further investigate the similarities and differences of systematic reviews, when used in different domains (although this is not considered a primary objective).
3. Tools to support systematic reviews are also being investigated, developed and used to support researchers, within healthcare and areas of social science (see Chapter One and Chapter Four).

5.3.4 Stage Four – Selecting an appropriate sampling technique

In this survey, non-probability sampling techniques were used to select participants. This is because the aims of this study prioritise the exploration of a topic, over wider generalisation of findings (see Section 5.2.2). When this is the case, a non-probability sampling approach is deemed adequate (Oates, 2006; Rea & Parker, 2014).

A combination of convenience and snowballing (non-probability) sampling techniques were used to recruit participants from the sampling frame. These techniques (see Section 5.2.2) are particularly suitable where the aim of the study centres around getting a feeling for the issues involved about a particular topic area (Robson, 2011).

For this study, an email invitation (see Appendix A4), which described the research project, the aim of the study and the commitment required (i.e. the estimated time based on the pilot interview) was sent to potential participants. Snowball sampling was also used in email correspondence (and in interviews) to try and gather additional participants.

5.3.5 Stage Five – Developing the survey instruments and procedures

As mentioned in Section 5.3.2, semi-structured interviews were selected as the sole data collection technique. Using this approach allowed for a mixture of ‘open’ and ‘closed’ questions (see Section 5.2.3.2) to be used to elicit a variety of responses from participants. The questions driving the

interviews for this survey were grouped into four main categories. Group 1 focused on the background and domain of the participant (Section 5.3.5.1). Group 2 investigated a participant's experience undertaking systematic reviews (Section 5.3.5.2). Group 3 questions related to experience with tools (Section 5.3.5.3). Group 4 was a feature rating exercise (see Section 5.3.5.4).

As well as the questions, consent forms and an interview preparation sheet given to participants prior to interview, were also developed. Details of these components are provided in Section 5.3.7, which provides information on the implementation of the study (Stage Seven).

5.3.5.1 Group 1 Questions – Background and domain context

At the beginning of the interview, it is important to start with a series of introductory, informal questions, which are relatively simple for the participant to answer and get them talking (Runeson & Host, 2009). This helps create a relaxed atmosphere for both the interviewer and participant (Hove & Anda, 2005). Group 1 included a series of open questions related to the research domain and context of the participant. In particular, questions aimed to discover background/demographic characteristics from participants [Q 1.1], the use and role of systematic reviews within the participant's domain [Q 1.2] and the current infrastructure (i.e. tools, checklists, guidelines etc.) available to support their conduct [Q 1.3]. The following questions were used:

- **[Q 1.1]** Could you tell me about the domain in which you are currently situated and some of the work that you do?
- **[Q 1.2]** How do systematic reviews play a role within your domain?
- **[Q 1.3]** What infrastructure is provided in your domain to support researchers undertaking a systematic review *(e.g. go-to guidelines, checklists, tools etc.)*?

5.3.5.2 Group 2 Questions – Personal experience performing systematic reviews

Group 2 questions focused on the personal experiences of participants, performing systematic reviews. In particular, a selection of open and closed questions were asked to learn the extent of their experience using the method [Q 2.1], their thoughts on the usefulness of systematic reviews

[Q 2.2] and its main challenges [Q 2.3] and which aspects of the process (i.e. the stages, phases, activities etc.) do they feel are most in need of tool support [Q 2.4]:

- **[Q 2.1]** How many systematic reviews have you contributed to?
- **[Q 2.2]** Do you find systematic reviews useful to perform?
- **[Q 2.3]** What, in your opinion, are the main challenges associated with undertaking a systematic review?
- **[Q 2.4]** What stages or aspects of the systematic review process do you feel are most in need of tool support?

5.3.5.3 Group 3 Questions – Experience with tools

Group 3 questions addressed the participant's use and experience of tools to support their systematic reviews. In particular, open questions were asked to identify any tools used by participants [Q 3.1], the strengths [Q 3.2] and weaknesses [Q 3.3] of the tools identified and overall thoughts [Q 3.4]. Questions asked were:

- **[Q 3.1]** – What tools (i.e. software) have you used to support the conduct of your systematic reviews?
    - Where appropriate, follow up questions (or prompts) were used in order to obtain sufficient information. For example: *What was the tool called? How did you learn about the tool?* Furthermore, some tools may have already been identified from participants in their responses to Question 1.3.
- **[Q 3.2]** –Based on your opinion and experience with the tool, what were its main strengths (i.e. what did you like about the tool)?
- **[Q 3.3]** – Based on your opinion and experience with the tool, what did you consider its main weaknesses (i.e. what did you dislike about the tool)?
- **[Q 3.4]** – Overall, did you find using tools useful (i.e. did you feel sufficiently supported)?
- **[Q 3.5]** – Would you use tools again to support future systematic reviews?

5.3.5.4 Group 4 Questions – Features of a systematic review tool

The final group of questions (Group 4) related to the feature set and associated importance weightings, developed as part of the evaluation framework (version 1.1) for systematic review tools (see Table 5-1). Closed questions were used, with a list of pre-determined responses for participants to choose from. Participants were asked to rate each feature as either **mandatory**, **highly desirable**, **desirable**, **nice-to-have** or **not necessary**. The ratings (apart from 'not necessary') were the same weightings defined in the evaluation framework. When presenting each feature to a participant, an example of how that feature might be implemented, within the context of an overall systematic review tool, was provided. Examples given were based on either currently existing (or proposed) features of tools, which were identified and examined in the literature (see Chapter Two) and feature analysis (see Chapter Three).

Participants were asked to rate features, which relate to tool support for each of the three main phases of a systematic review and the steps (or activities) within these phases. Concerning the importance of support for stages in the planning phase of a systematic review (see Section 1.1.3), the following questions were asked:

- **[Q 4.1.1]** – How important is a feature which provides support for ***developing the review protocol*****?**
    - This question investigates the importance of features that provide support for developing a systematic review protocol. Examples given of how a tool might provide support include:
        1. The use of collaborative templates to develop the protocol, and
        2. Mechanisms for version control, which help keep track of any changes made to the protocol (and who made them), during its development.
- **[Q 4.1.2]** – How important is a feature which supports ***protocol validation*****?**
    - This question addresses the importance of tool support for validating the review protocol. An example, provided to participants, of how a feature might support this

stage, considers the use of automated evaluation checklists. These items could be distributed for either internal (i.e. members of the review team) or external (i.e. external experts or evaluators) assessment.

Concerning the conduct phase of a systematic review and the stages within this phase (see Section 1.1.3), the following questions were asked to participants:

- **[Q 4.1.3**] – How important are features that provide support for conducting an ***automated search process*****?**
    - This question concerns tool support for the search process in a systematic review. Examples provided of how this stage might be supported, include:
        1. Being able to perform an automated search, of various electronic resources, from within the tool.
        2. Having the tool handle any search string format conversion, depending on a given digital library/electronic resource.
        3. The tool is able to handle any duplicate papers identified by the search.
- **[Q 4.1.4]** – How important is a feature which provides support for ***study selection*** (i.e. screening) ***and validation*****?**
    - This question investigates the importance of tool support for study selection and validation. As examples, provided to participants, a tool is considered to:
        1. Be able to support a multi-stage selection process (i.e. title/abstract, then full paper).
        2. Allow multiple users (i.e. members of the review team) to apply the inclusion/exclusion criteria (defined in the protocol) independently.
        3. Provide a facility to resolve any disagreements in selections.
- **[Q 4.1.5]** – How important is a feature which provides support for ***quality assessment*** (i.e. critical appraisal) ***and validation*****?**
    - This question addresses support for quality assessment and validation. As examples of support, a tool is considered to:

1. Enable the use of a suitable quality assessment criteria (which is defined in the review protocol).
2. Support multiple users to perform the scoring, independently.
3. Provide a facility to resolve disagreements (similar to the example of support provided for study selection and validation).

- **[Q 4.1.6]** – How important is a feature which supports ***data extraction*****?**
    - This question concerns the importance of support for extracting data from included studies (or papers). An example given to participants, considers support for the extraction and storage of both qualitative data (using classification and mapping techniques), as well as quantitative data (i.e. managing the specific numerical information reported from a study).
- **[Q 4.1.7] –** How important is a feature which supports ***data synthesis*****?**
    - This question investigates the importance of support for the data synthesis (or analysis) stage. As an example, provided to participants, a tool is considered to support simple qualitative and quantitative analysis of extracted data, which may include table and chart generation.
- **[Q 4.1.8]** – How important is a feature that supports ***text analysis*****?**
    - This question concerns the importance of support for text analysis. At this stage in the framework's development, text analysis was classed as an individual feature, since it was considered to have the potential for providing support to various aspects of a systematic review.
- **[Q 4.1.9] –** How important is a feature that supports ***meta-analysis*****?**
    - This question focuses on the importance of support for meta-analysis. Meta-analysis is a specific from of quantitative statistical analysis. Therefore, tool support for meta-analysis was classified as an individual feature.

The following questions investigate the importance of features that support stages in the report phase of a systematic review (see Section 1.1.3):

- **[Q 4.1.10] –** How important is a feature which supports ***writing the report*****?**
    - This question investigates the importance of support for writing-up the systematic review. An example, provided to participants, considers the use of a suitable and dynamic template. A similar example was provided regarding support for developing the review protocol in [Q 4.1.1].
- **[Q 4.1.11] –** How important is a feature which supports ***report validation*****?**
    - This question concerns the importance of support for validating the report. As an example, a tool might support this stage using automated evaluation checklists, which are distributed for internal/external assessment. A similar example was given for validating the review protocol in [Q 4.1.2].

The importance of features, which provide support for the overall management of a systematic review, were investigated using the following questions:

- **[Q 4.2.1]** – How important is support for ***multiple users*** to be able to work on a single systematic review?
    - This question investigates the importance of tool support for collaboration, when undertaking a systematic review.
- **[Q 4.2.2]** – How important are ***document management*** facilities?
    - This question concerns the importance of support for document management. For example, the facilities to manage large collections of papers, studies and the relationships between them.
- **[Q 4.2.3] –** How important are ***security*** features?
    - This question addresses the importance of security features within a systematic review tool. An example of support considers the use of a login or similar authentication system.
- **[Q 4.2.4] –** How important is a feature which supports ***role management***?
    - This question focuses on the importance of support for managing the roles of members of the review team. It is considered, for example, that:

1. A review team is able to state which users will perform certain activities (e.g. study selection, quality assessment, data extraction etc.).
2. Based on roles, the tool is able to allocate papers accordingly and handle appropriate authorisation.

- **[Q 4.2.5]** – How important is support for the ***re-use of data*** from past systematic reviews?
    - This question concerns the importance of support for re-using data from past systematic reviews. This might be useful for:
        1. Undertaking a new systematic review in a similar area where a relevant review already exists.
        2. When updating an existing systematic review.

Two questions were asked about the importance of features relating to the level of difficulty inherent in setting up and using a tool for the first time.

- **[Q 4.3.1]** – How important is a ***simple installation and setup*** procedure?
    - This question addresses the importance of a simple installation and setup procedure, particularly, when setting up and using the tool for the first time. Examples of support may include:
        1. A comprehensive installation guide.
        2. Video tutorials.
        3. An interactive tutorial, which uses example review data to explain the tool's features.
- **[Q 4.3.2]** – How important is it that the tool is as ***self-contained*** as possible?
    - This question concerns the importance of having a systematic review tool, which is as 'self-contained' as possible. 'Self-contained' refers to the system being able to work, primarily, as a stand-alone application, with minimal requirements for other external technologies in order to function.

Two questions were asked to investigate the importance of economic features (or factors) relating to the initial cost of the tool and subsequent support for its maintenance and any upgrades.

- **[Q4.4.1]** – How important is the ***financial cost*** of the tool?
- **[Q4.4.2]** – How important is a well and freely ***maintained*** tool?

Once all features had been rated, participants were invited to add any further features (either existing or novel), which they felt were missing from the set.

The set of questions reported in this section was the final version used for data collection. A pilot interview was performed, which led to some minor refinements in the delivery and sequencing of questions. Details of the pilot interview, and the modifications made, are described in Section 5.3.6. All questions were discussed and agreed upon by the lead researcher and student (CM) and his PhD supervisors.

5.3.5.5 The ethical approval process

To ensure that the survey was performed in an ethical manner, an application was submitted to the Keele University Ethical Review Panel (KUERP). KUERP undertakes assessments of whether proposed applications of research methods are ethically acceptable. A number of steps were taken to ensure how the study was ethically sound. Full ethical approval was granted on the 22nd May 2014. The relevant approval confirmation letter can be viewed in Appendix A8.

5.3.6 Stage Six - Pilot interview and modifications to the interview process

The survey instruments and procedures for this study were piloted with a PhD student who had undertaken two systematic reviews. The pilot interview confirmed the expectation that interviews would take approximately 45 minutes and also led to some changes in the delivery and sequencing of questions. Notably, the order in which features were presented (and rated) in Group 4, was changed. Questions concerning features relating to support for the systematic review process, were swapped with those addressing the importance of economic factors. Initially, economic features

were the first set of features presented to participants, when beginning Group 4 questions (see Section 5.3.5.4). This was followed by questions addressing the importance of ease of introduction and setup, systematic review activity support and, finally, process management. However, following the pilot interview, it was deemed more suitable to ask the questions which address the importance of systematic review activity features, before those addressing features concerning economic factors. There were two main reasons for this change:

1. In Group 2 (see Section 5.3.5.2), participants were asked questions that focused on their personal experience performing systematic reviews, followed by questions addressing their experience with any relevant tools in Group 3 (see Section 5.3.5.3). With these topics fresh in both the interviewer and participant's mind, it felt more natural to begin Group 4, with questions investigating the importance of features that support systematic review activities.
2. Certain questions in interviews can deal with sensitive issues such as religion, ethnicity and finance (Rea & Parker, 2014). Generally, the nature of the interview in this survey is free from any sensitive or unethical topics. As discussed in Section 5.3.5.6, the survey was successfully approved by the Keele University Ethical Review Panel, without any necessary revisions. It is recommended, however, that any potentially 'risky' questions should be asked relatively late in the interview (Oates, 2006; Robson, 2011; Rea & Parker, 2014; Patton, 2015). Doing so allows the participant to build trust with the interviewer, which can help improve the quality of responses given to more sensitive questions (Oates, 2006; Rea & Parker, 2014). It was decided, therefore, to have questions concerning economic factors about systematic review tools, asked last.

Furthermore, when presenting each feature to participants in Group 4, no examples of how it could be implemented were provided. Initially, the participant was expected to ask for an example if they required one. The concern was that the participant would focus their decision on the importance of the example implementation and not on the overall feature itself. In the pilot interview, however, the participant found it difficult to contextualise the feature and an example was needed for almost each feature presented. Based on the experience gained from the pilot interview, it was decided,

therefore, that an example would always be given when presenting each feature. Participants were instead informed that the examples provided were just one of many different ways in which the feature could be supported by a tool and that this should be taken into account when rating its importance. A final modification included the addition of 'not necessary'. This was provided as a new response option for participants to consider when rating the importance of features. Initially, when rating features, participants could only select one of the levels of importance (see Table 3-6) as a viable response. During the interview, however, there were instances where the participant felt the initial options available did not adequately reflect their true feelings about a particular feature. In some cases, for example, the participant felt a feature was unnecessary or irrelevant. The addition of 'not necessary' as a possible option when rating features, aimed to address this concern. In some cases, it is recommended that a second pilot be undertaken to assess the instruments and procedures in light of any refinements or modifications made (Oates, 2006; Robson, 2011; Rea & Parker, 2014). However, since the modifications made to this interview's format were only considered minor, this was not deemed a necessary action. (Robson, 2011).

5.3.7 Stage Seven – Implementing the survey

The interviews were carried out between June 2014 and September 2014. It is considered useful to send interview participants a list of topics (or questions) in advance, which gives them time to think about their views and prepare answers (Oates, 2006). Therefore, each participant was sent an 'Interview Preparation Sheet', prior to interview. This document, shown in Appendix A5, outlined the main themes to be covered during the interview, the expected duration (based on the time taken in the pilot interview) and measures which would be taken to ensure privacy and confidentiality. This document was re-visited at the start of each interview, with an opportunity for participants to raise any concerns and ask questions. Participants were also given two consent forms to complete (see Appendix A6 and A7) and permission was requested for the interviews to be (audio) recorded. In practice, there were no objections to interviews being recorded. All interviews were carried out face-to-face by a single interviewer and recorded using a digital audio recorder. The researcher took notes throughout each interview. On average, each interview lasted approximately 45 minutes.

The shortest interview lasted for 32 minutes and the longest interview lasted for 68 minutes. The same questions, topics and format were used for all interviews.

Data analysis took place concurrently with data collection. This approach is recommended by Miles *et al*., who advise interweaving data collection with analysis from the very start (Miles *et al.*, 2014). Full transcriptions of each interview were produced. Miles *et al*. notes that transcriptions can be produced to varying levels of detail. For this survey, transcripts aimed to reflect a straightforward summary of the main ideas, which were presented by a fluently spoken participant (Miles *et al.*, 2014). They did not include any mispronunciations, pauses or word emphases which might have occurred during the interview. In total, the interviews generated approximately 10 hours of audio recordings, each taking between five and six hours to fully transcribe.

5.3.8 Stage Eight – Data analysis approach

Four main categories were identified for analysis (see Figure 5-1). The first category (Category One) focused on analysing data about each participant. This includes data about a participant's domain (i.e. healthcare or social science), the number of systematic reviews they have undertaken and the type of systematic review they have performed (i.e. quantitative or qualitative). These results are presented in the next Chapter, Section 6.1.2. Category Two concerned the analysis of data about participant's views on the usefulness of systematic reviews (within their domain) and particular challenges associated with their conduct. These results are presented in Section 6.1.3 and discussed in Section 6.2.1. The third category (Category Three) aimed to classify tools (and associated types) identified by participants and analyse their main strengths and weaknesses. These results are presented in Section 6.1.4 and discussed in Section 6.2.2). The final category (Category Four) examined the participant's feature ratings and focused on analysing the strengths and weaknesses for each feature. These results are presented in Section 6.1.5 and discussed in Section 6.2.3. A discussion of the comparison between participants feature ratings and those included in version 1.1 of the evaluation framework is presented in Section 6.2.4.

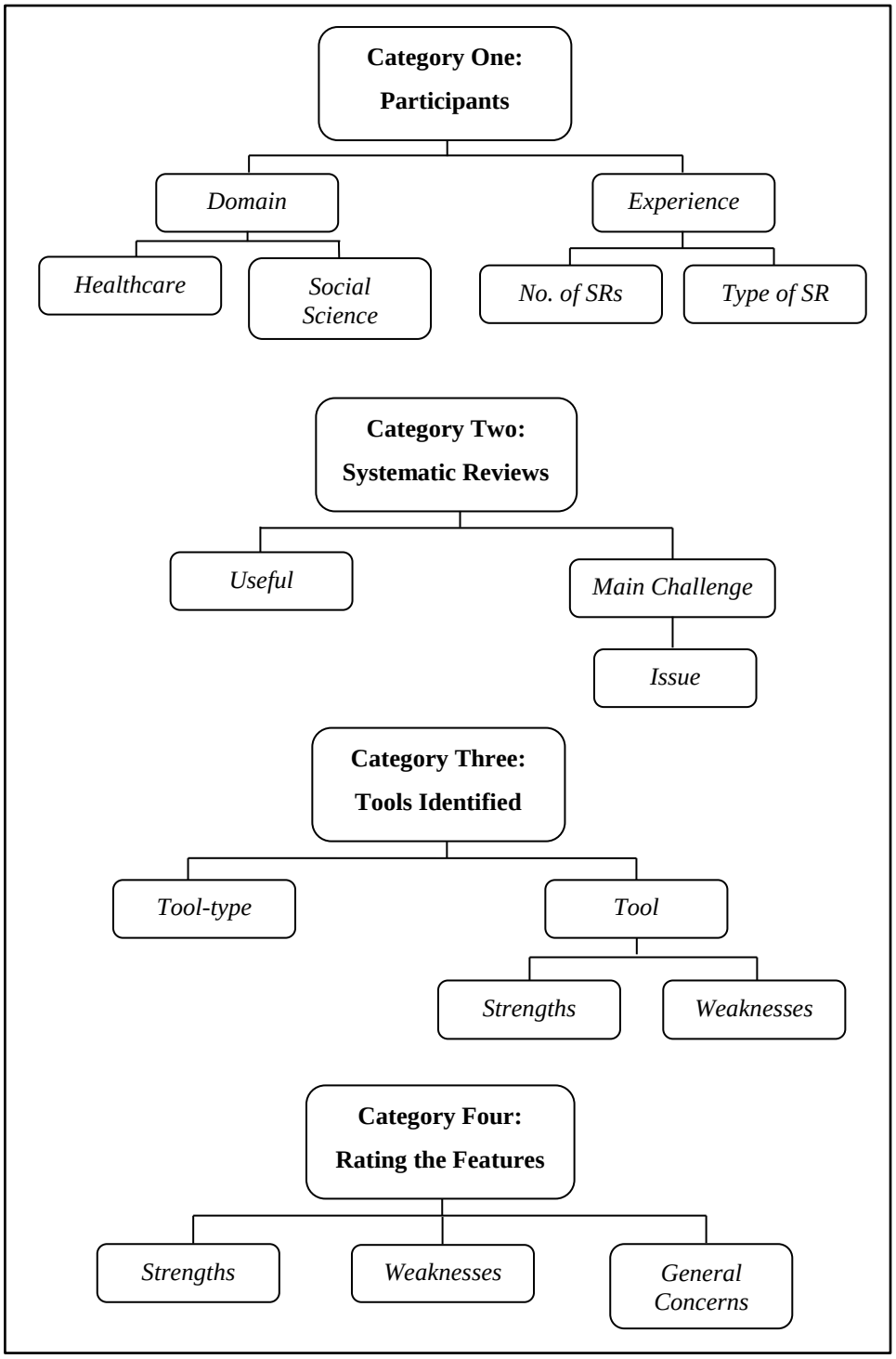


Figure 5-1. Categories identified for analysis

As indicated in the previous section (Section 5.3.7), analysis was an inductive process. This allowed for categories and codes to emerge progressively during the data collection (Miles *et al*., 2014). For example, in Category Three (see Figure 5-1), tools identified by participants were classified by type (see Table 6-4). Some tool types, such as reference managers and statistical

packages, were anticipated before beginning the interviews. However, throughout data collection additional types emerged. For example, ‘Custom-build’ (see Section 6.1.4) was defined in order to classify bespoke tools developed by a particular review team.

5.4 Summary

In this chapter, an introduction to an interview-based survey, undertaken to explore the experiences and opinions of systematic reviewers in domains outside of software engineering (specifically, in healthcare and areas of social science), has been provided. The background and design of the study have been reported. In particular, the appropriateness of the survey methodology and using interviews for data collection has been discussed, and the structure and content of the survey instrument has been described. An overview of the selection of participants, pilot interview and interview procedures, has been given. Details of the approach taken to analyse data have been provided. In the following chapter (Chapter Six), the results of the study, along with implications for the evaluation framework, are presented and discussed.

Chapter Six

Cross-Domain Survey: Results, Discussion and Conclusions

This chapter presents the results of the survey which explored the scope and practice of tool support for systematic reviewers in domains outside of software engineering. 13 researchers with experience of performing systematic reviews in healthcare and social science were interviewed. Qualitative and quantitative data was collected through semi-structured interviews and analysis followed an inductive approach. 21 software tools categorised into one of seven types were identified. Reference managers were the most commonly mentioned tools, followed by special-purpose systematic review tools. Features considered particularly important by participants were support for multiple users, support for data extraction and support for tool maintenance. Less important was the cost of the tool and support for preparing and validating the report. The features and importance levels identified by participants were compared with version 1.1 of the evaluation framework for an overall tool to support systematic reviews in software engineering. The results of the study and implications for the evaluation framework are discussed. Limitations of the survey and lessons learned from using semi-structured interviews are also provided.

6.1 Results

This section presents the results of the survey, based on the analysis of data collected from interviews with 13 participants. In Section 6.1.1, details of the participant response rate are provided. This is followed by the results which address the aims of the study outlined in the previous chapter (Section 5.1.1). To recap, the aims of the study were to:

1. Explore what tools are currently available and used to support systematic reviews in other domains; specifically, healthcare and areas of social science.
2. Identify what participants consider to be the most important characteristics (or features) of tools to support systematic reviews.
3. Compare the features and importance levels identified in the survey with those forming part of evaluation framework for tools which support the whole systematic review process in software engineering.

6.1.1 Participant response rate

Using the sampling approach described in Section 5.3.4, 18 responses were received from 49 emails sent. Seven of them expressed initial interest but were not able to commit to an interview due to workload or other personal reasons. Two additional participants, who met the requirements of the sampling frame (see Section 5.3.3), were found through snowballing. Therefore the findings reported in this chapter are based on the analysis of data collected from interviews with 13 participants. This study achieved a response rate of 22%.

6.1.2 Group 1 Questions – Background and domain context

13 participants from one of six departments from six institutions across the UK were interviewed. The departments and institutions were:

- Faculty of Health (Keele University)

- School of Psychology (Staffordshire University)
- School of Environment, Education and Development (University of Manchester)
- EPPI-Centre (University of London)
- Faculty of Education (University of Cambridge)
- School of Health & Related Research (University of Sheffield).

This section provides a short summary of each participant. In particular, their role, field of interest and experience with systematic reviews, is described based on participant's responses to Group 1 Questions (see Section 5.3.5.1). This information is also summarised in Table 6-1.

- **Participant 1 (P-01):** Research Associate in healthcare (primary care). Has experience with at least 10 systematic reviews, dealing with quantitative and qualitative data. The participant also has experience with meta-analysis.

- **Participant 2 (P-02):** Research Associate working in healthcare (primary care). Has experience undertaking three systematic reviews which all deal, primarily, with quantitative data.

- **Participant 3 (P-03)**: PhD student and Physiotherapist working in healthcare (primary care). The participant has completed two systematic reviews which deal, primarily, with qualitative data.

- **Participant 4 (P-04):** Senior Lecturer in health psychology. Delivers a taught course on systematic reviews. Has completed many systematic reviews dealing with, primarily, qualitative data, with at least four published.

- **Participant 5 (P-05):** Research Information Manager working in healthcare. The participant has been involved with at least 12 published systematic reviews, including several Cochrane Reviews.

id	Role	Domain	No. of SRs	Type of SR (Qualitative or Quantitative)
P-01	Research Associate	Healthcare	6 – 10	Both
P-02	Research Associate	Healthcare	1 – 5	Quantitative
P-03	PhD Student	Healthcare	1 – 5	Qualitative
P-04	Senior Lecturer	Healthcare	1 – 5	Qualitative
P-05	Information Officer	Healthcare	11 – 15	Quantitative
P-06	Lecturer	Healthcare	1 – 5	Quantitative
P-07	Lecturer	Social Science	1 – 5	Quantitative
P-08	Information Officer	Social Science	15+	Both
P-09	Professor	Social Science	15+	Both
P-10	Systematic Reviewer	Social Science	6 – 10	Both
P-11	Research Associate	Social Science	1 – 5	Both
P-12	Professor	Social Science	15+	Qualitative
P-13	Information Specialist	Healthcare	15+	Both

Table 6-1. Participant information

- **Participant 6 (P-06):** Lecturer of Nursing, qualified Nurse and PhD student. The participant has completed two large-scale systematic reviews which deal, primarily, with quantitative data.

- **Participant 7 (P-07):** Lecturer in Psychology of Education. Has completed three large-scale systematic reviews dealing with, primarily, quantitative data. The participant also has experience with meta-analysis.

- **Participant 8 (P-08):** Information Officer working in healthcare, social care and international development. The participant has been involved with a large number of systematic reviews, dealing with both qualitative and quantitative data.

- **Participant 9 (P-09)** Professor of Social Research and Policy. The participant has considerable experience performing large-scale systematic reviews across a variety of disciplines, dealing with both quantitative and qualitative data.

- **Participant 10 (P-10)** Systematic Reviewer based in healthcare (public health). |Has completed seven large-scale systematic reviews, primarily dealing with qualitative data. The participant also has experience with meta-analysis.

- **Participant 11 (P-11)** Research Associate based in Education Technology. The participant has undertaken two systematic reviews dealing with both qualitative and quantitative data.

- **Participant 12 (P-12)** Professor of Education and Child Psychology. The participant has been involved with approximately 20 systematic reviews at various levels, dealing with, primarily, qualitative data.

- **Participant 13 (P-13)** Senior Information Specialist based in healthcare and information sciences. Experience with over 30 large-scale systematic reviews, dealing with both quantitative and qualitative data.

Participants can be broadly classified into either a healthcare or social science domain. However, it is worth noting that participants covered many different branches of these two domains.

6.1.3 Group 2 Questions – Personal experiences of performing systematic reviews

In this section, participant's opinions on the overall usefulness, main challenges and specific issues associated with systematic reviews, are presented. This data was obtained based on responses to Group 2 Questions (see Section 5.3.5.2) The main positive characteristics about systematic reviews, identified by participants, are summarised in Table 6-2. The main challenges and issues are shown in Table 6-3.

Positive characteristics about systematic reviews
Platform for future research
Useful for students and novices - (aids learning a new field).
Good for publications and citations
High rigour

Table 6-2. Main positive characteristics of systematic reviews identified by the participants

Main Challenges	Specific Issues
Search process	*Search string translation for individual databases* *Inconsistency with terminology* *Time consuming* *Developing the search strategy*
Generally time consuming	-
No standardisation	-
High difficulty	-
Management	*Managing large-scale SRs* *Transparency* *Handling duplicates* *Collaboration* *Negotiation with policy makers* *Managing the relationships between studies and papers* *Version control*
Analysis	*Qualitative analysis* *Meta-analysis*
Study selection / screening	*Resolving disagreements* *Managing the criteria* *Criteria consistency across multiple coders*
Quality assessment / critical appraisal	*Resolving disagreements* *Managing the criteria* *Criteria consistency across multiple coders* *Assessing the quality of the study and not the paper*
Protocol Development	*Developing the research questions*
Developing the report	*Formatting references*
Validation	*Knowing when to check for consistency*

Table 6-3. Main challenges (and specific issues) of systematic reviews identified by the participants

Generally, participants were very positive about the usefulness of systematic reviews and their impact in their respective domains. In particular, many participants praised a systematic review's ability to ***"act as a platform for further research"*** and help ***"bridge the gap"*** between research and practice. Furthermore, systematic reviews were said to be particularly useful for PhD students (***"as someone doing a PhD, I think it's a great way of encompassing a lot of the literature"***) and for researchers entering a new field (***"as someone new to the area, it was a great way for me to become familiar with a lot of the research"***). Some participants mentioned that, in their fields, systematic reviews were ***"held in good regard publication wise and citation wise"*** and that ***"people always go to systematic reviews for their 'next thing'.*** Participants also praised the ***"rigour"*** of a systematic review; particularly in helping to ***"structure and make sense"*** of large amounts of evidence. Interestingly, one participant talks about how systematic reviews can improve the perception of some qualitative papers, which are often overlooked in favour of highly-

quantitative (i.e. statistical) studies (***"a qualitative paper might just get published and not really looked at. But, by doing a*** [systematic review] ***you can make a stronger argument*** [for the qualitative study]***"***).

There were, however, a number of common challenges/issues identified by participants. One of the biggest complaints was the time consuming and difficult nature of a systematic review ***("they are really bloody hard and time consuming")***. Having to ***"manage thousands and thousands of records"*** in a way that is ***"robust and transparent"*** was considered particularly hard. Some participants also had concerns about the lack of standardisation of a systematic review (***"every review I've read has had a different methodology … there's no standardisation"***). Managing the search process was also a frequently stressed issue by participants. Developing the search strategy, for example, was considered particularly challenging (and time consuming) due to ***"inconsistent terminology"*** and having to ***"adapt the search strings for different databases"*** because of formatting restrictions imposed by electronic resources. One participant states that the goal of the search strategy is ***"striking a balance between being comprehensive and making it manageable."*** Other participants shared this view and mentioned how because of the ***"sheer volume of literature"***, comprehension was often compromised in favour of what was ***"realistically achievable"***. Developing the review protocol was also considered a difficult task. Some participants even considered developing the protocol as the ***"most difficult and complex"*** part of a systematic review; particularly, coming up with the ***"right sort of research question"***. A number of participants discussed issues associated with quality assessment and study selection. Particular issues with these stages included ***"managing the criteria across multiple coders"*** and ***"resolving disagreements".*** These concerns were considered particularly challenging within teams comprised of researchers with varying levels of experience (e.g. PhD students and their supervisors). Some participants also stressed issues with preparing the final report; in particular, ***"formatting the reports and references at the end is quite problematic"***.

6.1.4 Group 3 Questions – Experience with tools

In this section, the tools referenced by participants are presented and have been classified by type. Results were analysed based on data collected from responses to Group 3 Questions (see Section 5.3.5.3). A summary of these results is presented in Table 6-4.

There were 21 tools identified by participants, which have been classified into seven categories:

- **Reference management tools**. Five reference managers were identified; namely, *RefWorks*, *EndNote* / *EndNote-Web*, *Mendeley*, *Reference Manager* and *ProCite*.

- **Special-purpose tools**, which relate to applications that target particular stages of a systematic review or the process as a whole. Two tools were identified; namely, *Review Manager (RevMan)* and '*EPPI-Reviewer.'*

- **Basic productivity tools** (or general-purpose tools), which relate to applications such as word processors and spreadsheets. Two tools were identified; namely, *Microsoft Word* and *Excel*.

- **Advanced analysis software**, which concern high-end data analysis or statistical packages. Five tools were identified; *STATA*, *SPSS*, *NVivo*, *ATLAS.ti* and *Mplus*.

- **Custom-built tools**, which relate to custom, bespoke tools developed specifically for a participant's review. Two tools were identified; an unnamed *web-based coding tool* with support for multiple users and an unnamed *excel add-in* to assist analysis

- **Meta-analysis**, which includes tools developed to specifically support this aspect of a systematic review. Two tools were identified; namely, *MetaEasy* and *MetaLight.*

- **Other**. Three tools were unclassified; namely, *PubReMiner* (search tool for the PubMed literature database), *FreeMind* (a freely available mind mapping tool and the *RIS Conversion Tool*.

Tool Type	Tools	Participants (P)	Total
Reference Management Tools	RefWorks	P-01; P-03; P-04; P-05; P-06	5
	EndNote / EndNote Web	P-04; P-05; P-08; P-09; P-13	5
	Mendeley	P-03; P-07; P-12; P-13	4
	Reference Manager (RefMan)	P-02; P-08; P-13	3
	ProCite	P-09	1
Special Purpose Tools	Review Manager (RevMan)	P-01; P-02; P-03; P-05; P-07; P-09; P-13	7
	EPPI-Reviewer	P-08; P-09; P-10; P-11	4
Basic Productivity Tools	Microsoft Word	P-02; P-04; P-09; P-13	4
	Microsoft Excel	P-02; P-07; P-12	3
Advanced Analysis Software	STATA	P-01; P-02; P-09	3
	NVivo	P-07; P-12	2
	SPSS	P-06; P-09	2
	Mplus	P-07	1
	ATLAS.ti	P-12	1
Other	FreeMind	P-04; P-13	2
	RIS conversion tool	P-08	1
	PubReMiner	P-13	1
Custom-built tool	Web-based coding tool	P-07	1
	Excel add-in	P-02	1
Meta-analysis tools	MetaEasy	P-07	1
	MetaLight	P-07	1

Table 6-4. Tools identified by participants

The majority of tools identified by participants were reference managers. In particular, *RefWorks* and *EndNote* were mentioned most often. In the following sub-sections, the strengths and weaknesses of both systems (as summarised by participants), are presented.

6.1.4.1 RefWorks

RefWorks was praised by participants for its ability to ***"aid your systematic search process"*** and being able to ***"check for duplication"*** of papers. To some extent, *RefWorks* could also support study selection, with one participant explaining how they ***"classified studies using folders"*** to manage included and excluded papers. *RefWorks*, however, was criticised for the lack of a bulk

Strengths	Weaknesses
Managing the search	No bulk-export
Aiding study selection	Usability issues
Duplicate removal	Citation formatting

Table 6-5. Main strengths and weaknesses of RefWorks

export feature (***"you cannot export all your searches in one go. You have to do them in bits and pieces.")***. Some participants also complained about usability ***(I don't think it's easy to use at all. There is a lot compacted onto one screen")*** and formatting references ***("I've never found anyone who hasn't had trouble with it.").*** Table 6-5 summarises the main strengths and weaknesses identified by participants for *RefWorks*.

6.1.4.2 EndNote

EndNote was praised for having a web-based interface for remote access ***("I use EndNote Web so I can access it anywhere, which is good.")***. Similar to *RefWorks*, some participants used *EndNote* to support study selection even though a feature to support this stage is not explicitly supported ***("I don't think it's built to do that, it's just the way I use it.")***. Participants also liked having ***"discrete databases for each review."*** This is not the case in *RefWorks*, which uses a ***"folder driven system."*** Participants at times, however, felt restricted by the tool; with some feeling they were unable to take their data to the ***"next stage of the review"*** due to weak export capabilities. Some raised concerns about poor support for team-based systematic reviews ***("It is not ideal when you've got a big team doing things.")*** and whether the system could effectively handle large numbers of papers/studies ***("people are concerned that it doesn't have the capacity to deal with huge numbers of references").*** Table 6-6 summarises the main strengths and weaknesses identified by participants for *EndNote*.

The second most common types of tool identified by participants were special-purpose systems, designed to support particular stages of a systematic review (or the whole process). In particular, the two tools identified were, *EPPI-Reviewer* and *RevMan*. In the context of the overall project,

Strengths	Weaknesses
Web-based/remote access	Usability issues
Aiding study selection	Concerns for collaboration
Support for multiple projects	Capacity for large-scale reviews

Table 6-6. Main strengths and weaknesses of EndNote

these tools were particularly interesting, as they conform to the style of tool forming the main focus of the evaluation framework. They are the closest, in terms of how support is offered, to the candidate tools evaluated in the feature analysis reported in Chapter Three. The strengths and weaknesses of *EPPI-Reviewer* and *RevMan*, identified by participants, are presented:

6.1.4.3 EPPI-Reviewer

The current version of *EPPI-Reviewer*, *EPPI-Reviewer 4*, is a comprehensive single or multi-user web-based system for managing systematic reviews across healthcare and social science domains. During the interviews, participants were very positive about the variety of ways in which the tool can support the systematic review process. For example, *EPPI-Reviewer* includes a feature aiming to improve the efficiency of a systematic review, which uses text mining ***"to prioritise the most relevant studies."*** This feature ***"pulls the most relevant ones*** [studies] ***to the beginning"*** and allows the review team ***"to start the full data extraction of the studies before finishing the screening."*** *EPPI-Reviewer* also uses visualisation techniques to support thematic analysis. This feature, which allows users to ***"depict the relationships between concepts,"*** was also considered useful. Participants, however, felt *EPPI-Reviewer* had a steep learning curve ***("There is a learning curve on it. It's not something you can just pick up and use instantly")*** and that it ***"takes a while***

Strengths	**Weaknesses**
Text mining	Learning curve/difficulty
Qualitative analysis	

Table 6-7. Key strengths and weaknesses of EPPI-Reviewer

to learn all of the different things." In addition, some participants felt the ***"training could be improved."*** Table 6-7 summarises the key strengths and weaknesses for *EPPI-Reviewer*.

6.1.4.4 RevMan

RevMan primarily supports the preparation and maintenance of Cochrane Reviews; although, it can be used to support other reviews. *RevMan* was praised by participants for its good support for statistical analysis techniques; in particular, meta-analysis ***("meta-analysis is quite easy").*** Support for protocol development was also considered useful ***("It helps with the protocol stage as well. It helps guide you.").*** Some users, however, felt, at times, restricted by the tool since some of its features were not accessible unless it was a Cochrane Review ***("if your review is not Cochrane commissioned then you can't use that feature of RevMan.").*** Other users also felt ***"confused"*** by the tool and felt it was all a ***"bit over the top."*** Key strengths and weaknesses of *RevMan* are summarised in Table 6-8

Strengths	Weaknesses
Meta-analysis	Locked features
Protocol Development	Usability issues

Table 6-8. Key strengths and weaknesses of RevMan

6.1.5 Group 4 Questions – Features of a systematic review tool

In this section, the results of the feature rating exercise described in Section 5.3.5.4, is presented. In addition, a summary of the key points raised by participants, for each feature, is given. The feature ratings are presented in Table 6-9 (the bold, underlined number is the modal response rating for the feature).

6.1.5.1 Development of the review protocol (F3-F01)

This feature is concerned with support for the development of a review protocol, including, version management, by a review team. Ratings for this feature are shown in row 5 of Table 6-9 and visualised in Figure 6-1

Feature Set	Row No.	id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F1 - Economic	1)	F1-F01	No financial payment	0	**5**	3	1	4	Highly Desirable
	2)	F1-F02	Maintenance	6	**7**	0	0	0	Highly Desirable
F2 - Ease of Introduction	3)	F2-F01	Simple installation and setup procedure	**6**	5	1	1	0	Highly Desirable
	4)	F2-F02	Self-contained	0	**6**	**6**	0	1	Highly Desirable
F3 – Systematic Review Activity Support	5)	F3-F01	Development of review protocol	2	**4**	2	3	2	Desirable
	6)	F3-F02	Protocol validation	1	1	**5**	1	**5**	Desirable
	7)	F3-F03	Supports automated searches	3	**4**	3	3	0	Highly Desirable
	8)	F3-F04	Study selection and validation	5	**6**	2	0	0	Highly Desirable
	9)	F3-F05	Quality assessment and validation	5	**7**	1	0	0	Highly Desirable
	10)	F3-F06	Data extraction	**7**	5	1	0	0	Highly Desirable
	11)	F3-F07	Data synthesis	5	**7**	1	0	0	Highly Desirable
	12)	F3-F08	Text analysis	0	3	2	**5**	3	Nice-to-have
	13)	F3-F09	Meta-analysis	4	**5**	2	2	0	Nice-to-have
	14)	F3-F10	Report write-up	0	2	**6**	4	0	Nice-to-have
	15)	F3-F11	Report validation	0	3	3	3	**4**	Nice-to-have
F4 – Process management	16)	F4-F01	Multiple users	**9**	2	2	0	0	Mandatory
	17)	F4-F02	Document management	**6**	4	2	1	0	Mandatory
	18)	F4-F03	Security	**6**	2	1	3	1	Desirable
	19)	F4-F04	Role management	3	3	2	**4**	1	Highly Desirable
	20)	F4-F05	Re-use of data from past projects	3	**7**	3	0	0	N/A

Table 6-9. Summary of participant ratings for each feature

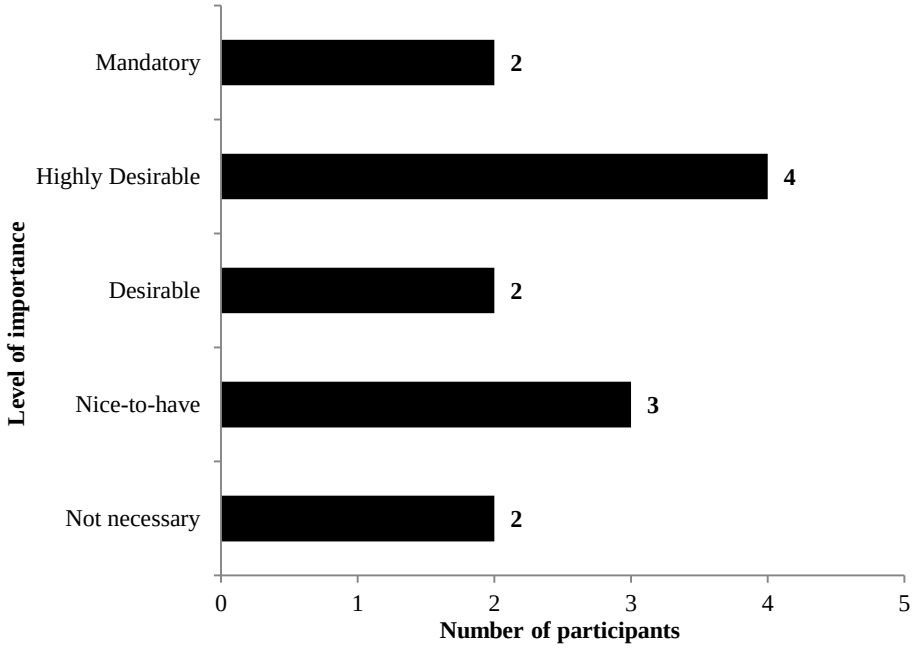


Figure 6-1. Participant ratings for developing the review protocol

Participants stated that support for the review protocol would be ***"highly useful for collaboration"***; particularly, within a ***"large-scale review team"***. One participant also thought this would be ***"particularly useful for developing a trial protocol"***. Some participants, however, were unsure of its usefulness, stating that there were ***"already resources*** (e.g. Cochrane Handbook) ***which support this"*** and that using ***"Word and track changes"*** is sufficient.

6.1.5.2 Protocol validation (F3-F02)

This feature concerns support for validation of the protocol. This may be supported, for example, with an automated checklist that is distributed to team members and/or external evaluators. Ratings for this feature are shown in row 6 of Table 6-9 and visualised in Figure 6-2.

Participants felt a feature would be useful for ***"making sure you don't miss anything"*** and that by having a ***"workable check-list"***, it makes things easier. Some participants, however, felt that whilst protocol validation was ***"incredibly important"***, introducing automation might be ***"over-complicating the process"***, with many feeling it was simply ***"unnecessary."***

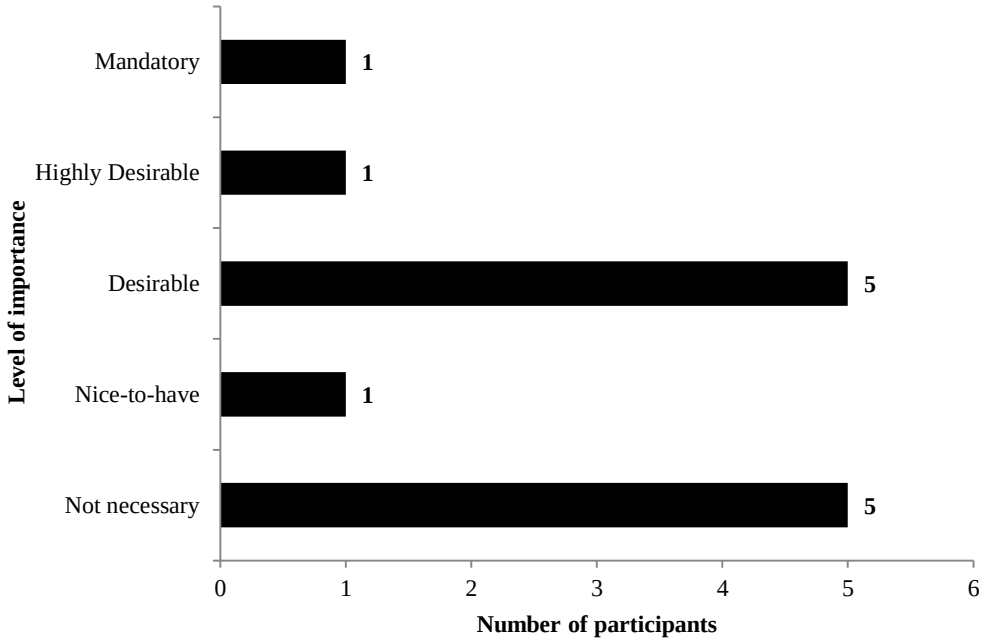


Figure 6-2. Participant ratings for protocol validation

6.1.5.3 Supports automated searches (F3-F03)

This feature concerns support for the search. For example, the user is able to perform an automated search from within the tool, which identifies duplicate papers and handles them accordingly. Ratings for this feature are shown in row 7 of Table 6-9 and visualised in Figure 6-3.

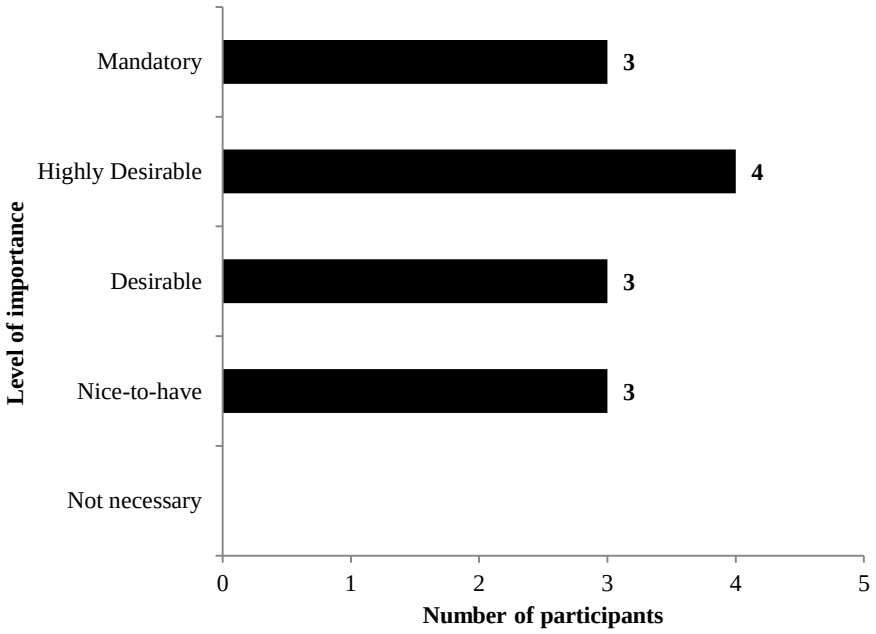


Figure 6-3. Participant ratings for supporting automated searches

Many participants felt this would be a ***"very useful"*** feature and ***"save a lot of time"***. In particular, participants felt that automated support could be helpful for ***"developing the search strategy"*** particularly when ***"piloting your search terms."*** A number of participants, however, raised concerns about the ***"reliability"*** and that using such a feature would not be *"**searching the databases properly."***

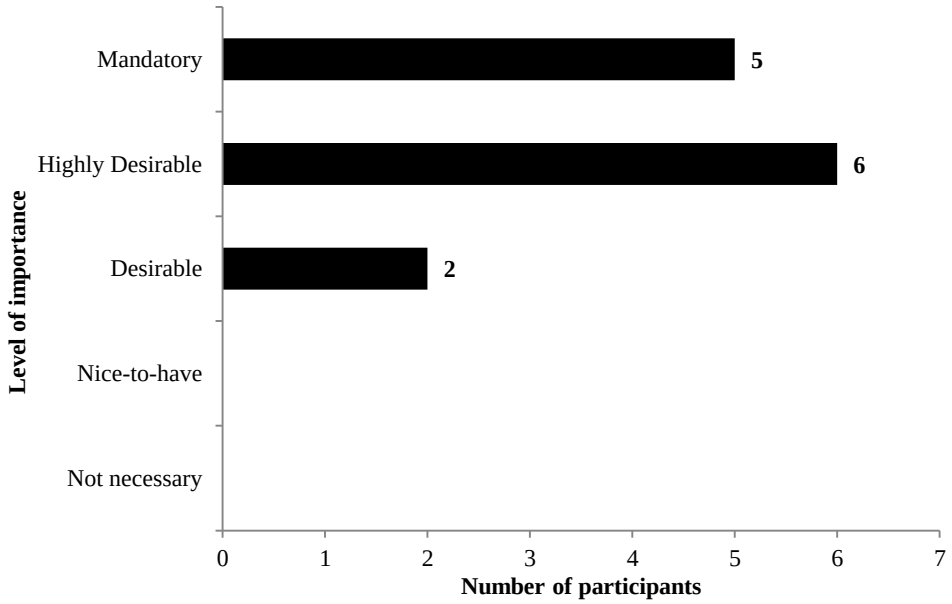


Figure 6-4. Participant ratings for study selection

6.1.5.4 Study selection and validation (F3-F04)

This feature concerns support for study selection and validation. It is considered, for example, that a tool provides support for a multi-stage selection, for multiple users to apply the inclusion/exclusion criteria independently and a facility to resolve disagreements. Ratings for this feature are shown in row 8 of Table 6-9 and visualised in Figure 6-4.

Participants felt it had the potential to ***"reduce a lot of workload"*** and could ***"speed up the overall process."*** In particular, the ability for multiple users to ***"be doing it simultaneously"*** was considered ***"very useful."*** A facility for resolving disagreements was also praised. Some participants, however, felt that a lot of what the feature was targeting support for could be solved

with a ***"quick conversation"*** between members of the review team, and that ***"we shouldn't lose the value of that."***

6.1.5.5 Quality assessment and validation (F3-F05)

This feature concerns support for quality assessment and validation. It is considered, for example, that a tool enables the use of a suitable quality assessment criteria, allows multiple users to perform the scoring and provides a facility to resolve disagreements. Ratings for this feature are shown in row 9 of Table 6-9 and visualised in Figure 6-5.

The majority of participants felt this would be another useful feature since ***"all these things otherwise require meetings and organisation."*** In particular, a facility to compare user assessments and ***"identify where your disagreements are, would be really good."*** Some participants raised concerns about the feature's ***"flexibility"*** and that, as a user, you'd need to be able to ***"tailor the quality criteria"*** based on the type of studies included in a review.

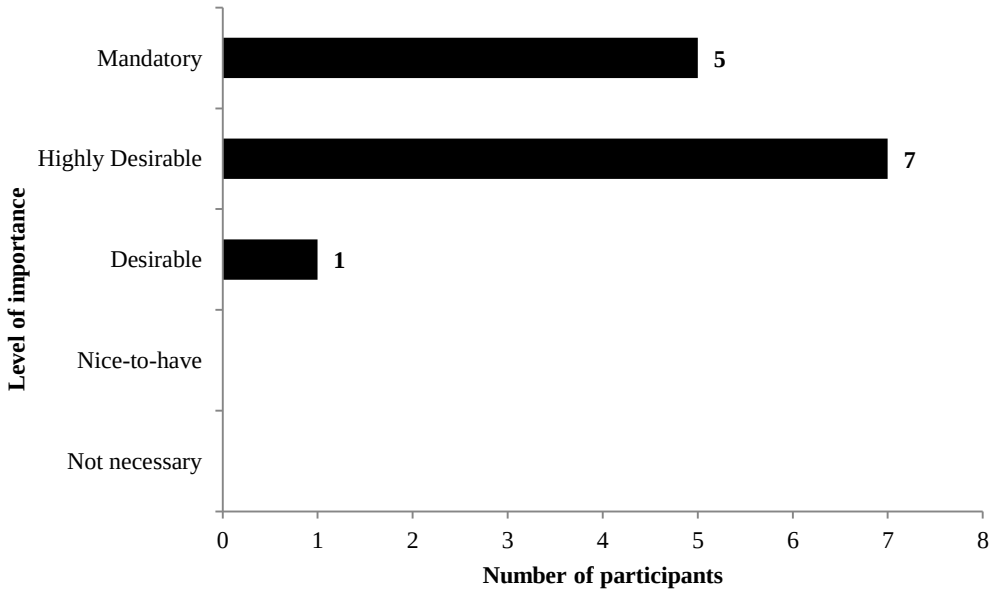


Figure 6-5. Participant ratings for quality assessment

6.1.5.6 Data extraction (F3-F06)

This feature concerns support for data extraction, including, the extraction of qualitative data using classification and mapping techniques, as well as quantitative data (i.e. managing the specific

numerical data reported from a study). Ratings for this study are shown in row 10 of Table 6-9 and visualised in Figure 6-6.

Many participants felt that ***"something to store all that information would be useful."*** In the context of an end-to-end (i.e. overall) tool, the ability to have extracted data ready to go ***"straight into the analysis"*** was also praised. Some participants, however, had a ***"hard time seeing how*** [the feature] ***would work properly in practice"***, particularly when handling qualitative data.

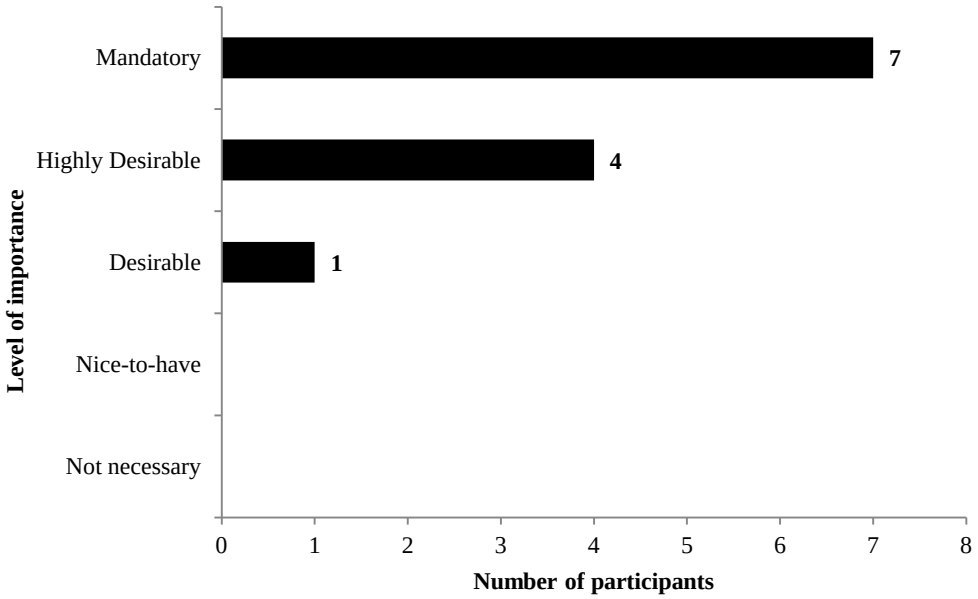


Figure 6-6. Participant ratings for data extraction

6.1.5.7 Data synthesis (F3-F07)

This feature concerns support for data synthesis; in particular, automated analysis of extracted data. Ratings for this feature are shown in row 11 of Table 6-9 and visualised in Figure 6-7.

Many participants felt that support for this aspect of a systematic review would be ***"very helpful"*** and would ***"save a lot of work"*** One participant felt that ***"less experienced reviewers would find*** [this feature] ***particularly useful."*** A number of participants stressed that, although support for analysis would be useful, ***"data preparation"*** (i.e. preparing data into a suitable format for analysis in more advanced applications) would also be helpful, with one participant claiming it should be ***"mandatory for being able to get structured data out into different formats."***

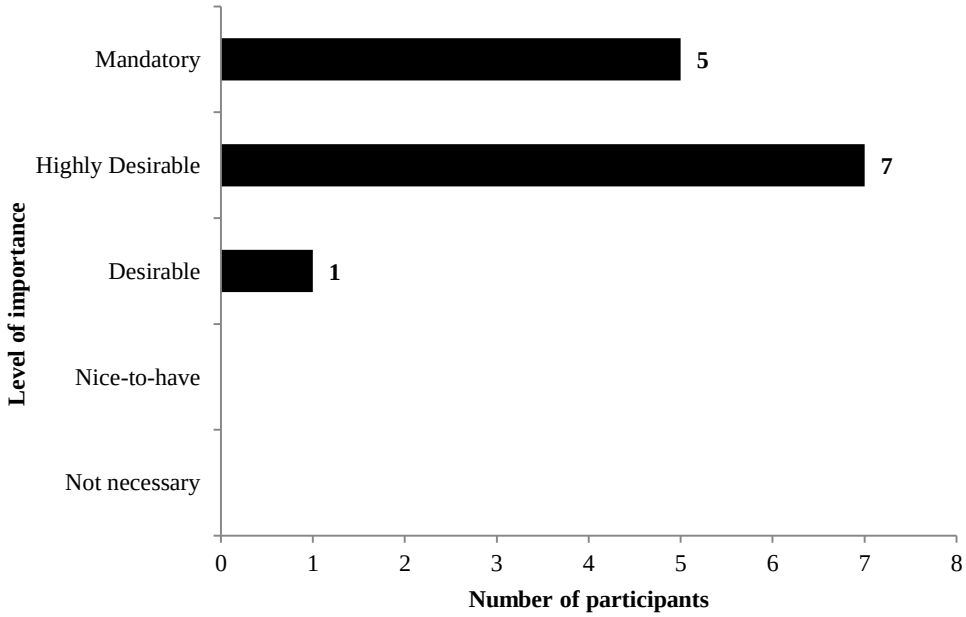


Figure 6-7. Participant ratings for data synthesis

6.1.5.8 Text analysis (F3-F08)

This feature concerns support for text analysis. Ratings for this feature are shown in row 12 of Table 6-9 and visualised in Figure 6-8.

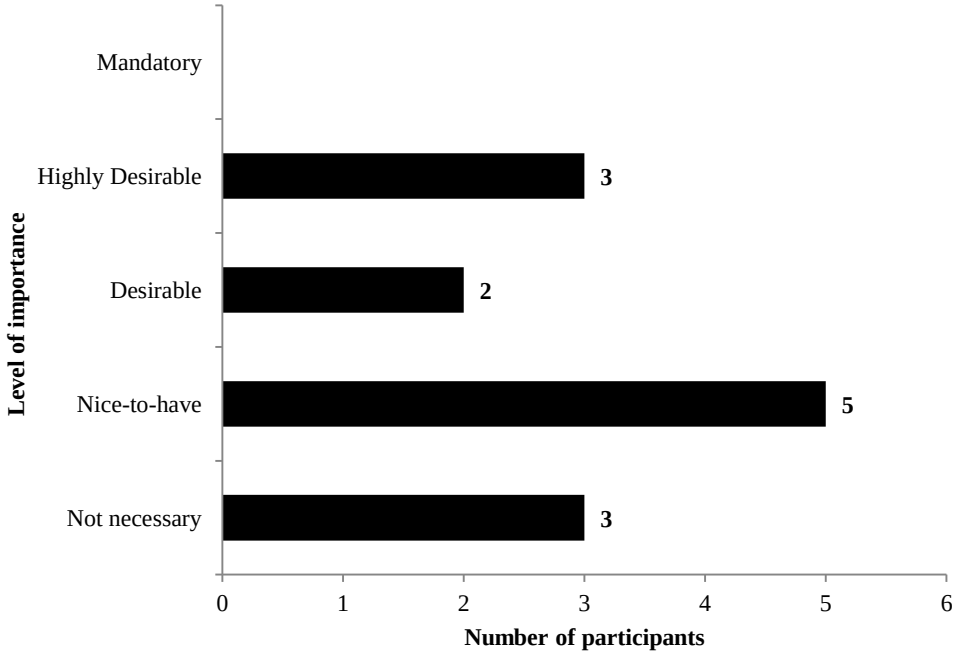


Figure 6-8. Participant ratings for text analysis

Some participants felt text analysis would be a useful aid to certain stages of the systematic review process (e.g. study selection), and had the potential to ***"cut down on time for very big reviews."*** One participant felt that text analysis would become ***"increasingly more important as the complexity of the literature increases."*** For now, however, many participants ***"struggled to see the value"*** in such a feature.

6.1.5.9 Meta-analysis (F3-F09)

This feature concerns support for meta-analysis. Ratings for this feature are shown in row 13 of Table 6-9 and visualised in Figure 6-9.

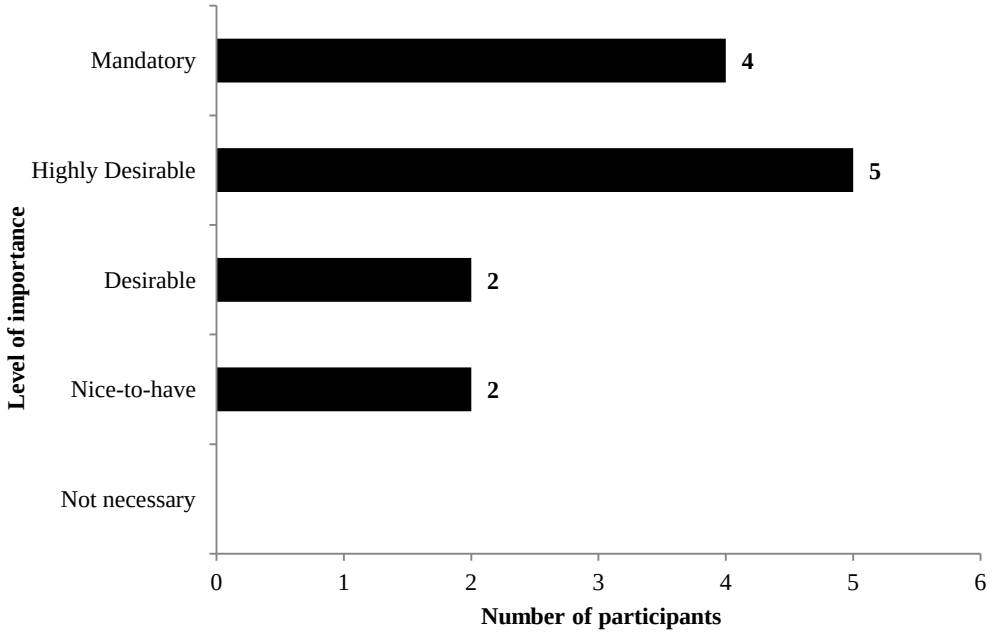


Figure 6-9. Participant ratings for meta-analysis

Some participants felt support for meta-analysis was ***"very important"*** particularly for novices as, ***"for a lot of people undertaking a systematic review for the first time, meta-analysis is their biggest fear."*** Another participant noted that not having to ***"mess around importing and exporting data to, and from, different applications would be nice."*** Some participants, however, challenged the importance of support for meta-analysis as ***"not all reviews need it"*** so, for some users, it would be an ***"irrelevant feature"***.

6.1.5.10 Report write-up (F3-F10)

This feature concerns support for writing the report. It is considered, for example, that the tool enables the use of a suitable template to assist the write-up. Ratings for this feature are shown in row 14 in Table 6-9 and visualised in Figure 6-10.

Participants felt support for the write-up would give reviewers a ***"starting point*** and that a ***"good template"*** would provide the ***"bones with which to put on the flesh."*** Many participants, however, felt such a feature would suffer because there are ***"so many different journals, which have so many different ways that they want you to present your work,"*** that having a feature, which could ***"map to all of them,"*** would be ***"difficult."***

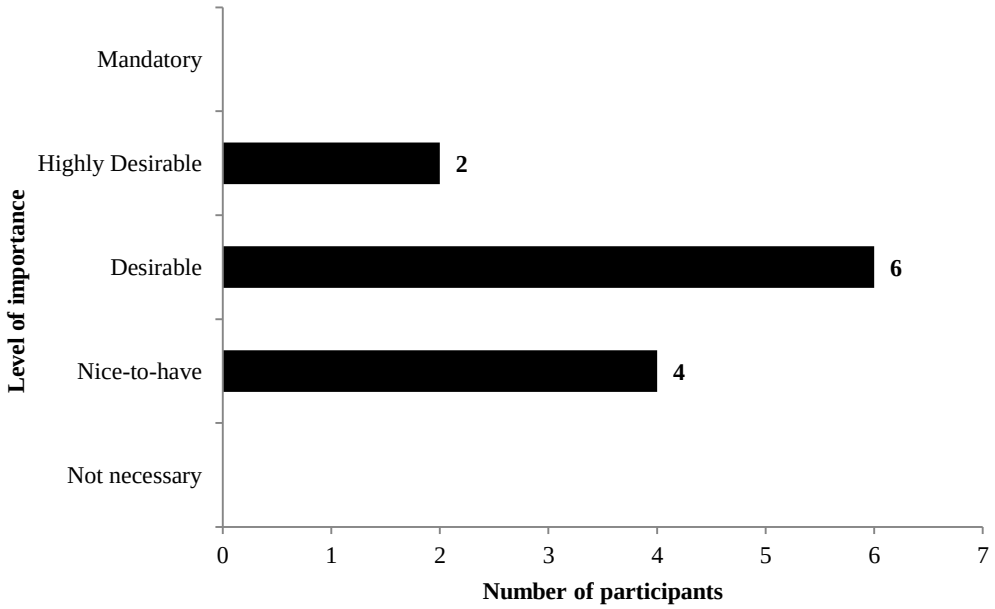


Figure 6-10. Participant ratings for writing the report

6.1.5.11 Report validation (F3-F11)

This feature concerns support for validation of the report. Similar to support for protocol validation, this might be supported using an automated checklist that is distributed to team members and/or external evaluators. Ratings for this feature are shown in row 15 of Table 6-9 and visualised in Figure 6-11.

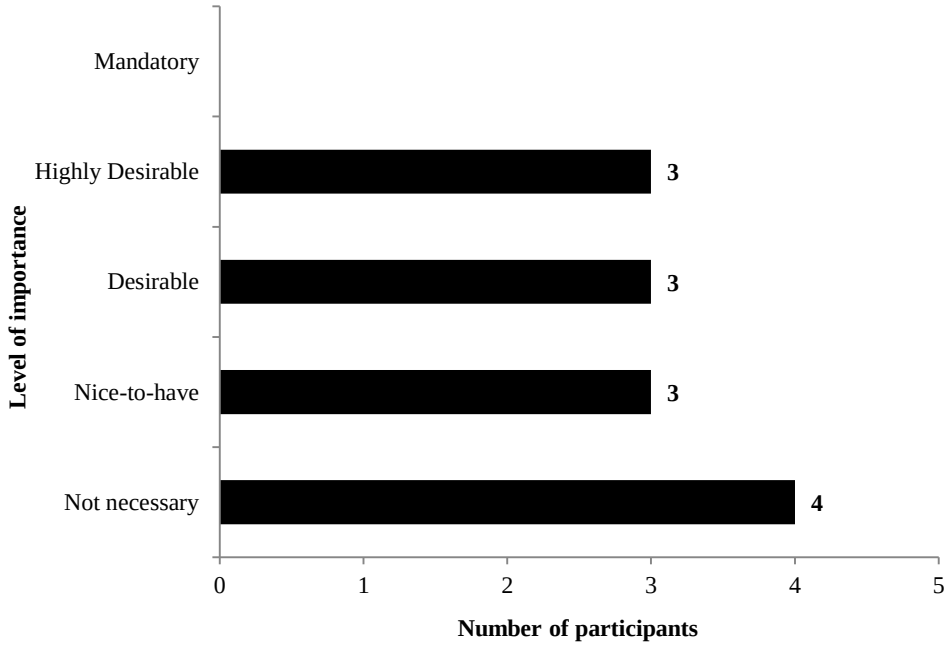


Figure 6-11. Participant ratings for report validation

One participant felt this might be a useful feature if, for example, ***"the validation itself is done by the team members, but the framework for the validation is generated by the tool, possibly through previous sets of criteria."*** Some participants felt an internal ***"peer review process"*** supported by a tool could also be useful. Many participants, however, felt that there were already ***"plenty of resources"*** that already supported this aspect of a systematic review.

6.1.5.12 Multiple users (F4-F01)

This feature concerns support for multiple users to be able to work on a single review. Ratings for this feature are shown in row 16 of Table 6-9 and visualised in Figure 6-12.

Many participants felt support for multiple users was really important. In particular, allowing users to collaborate within ***"large-scale teams"*** was considered very useful. Participants felt that it was ***"quite rare that you would just have one person doing a systematic review on their own"***, with one participant even stating that *"**you could not do a*** [systematic review] ***on your own"***. Therefore, in order for other features such as study selection, data extraction and quality assessment to be fully supported by a tool, support for collaboration would need to be in place.

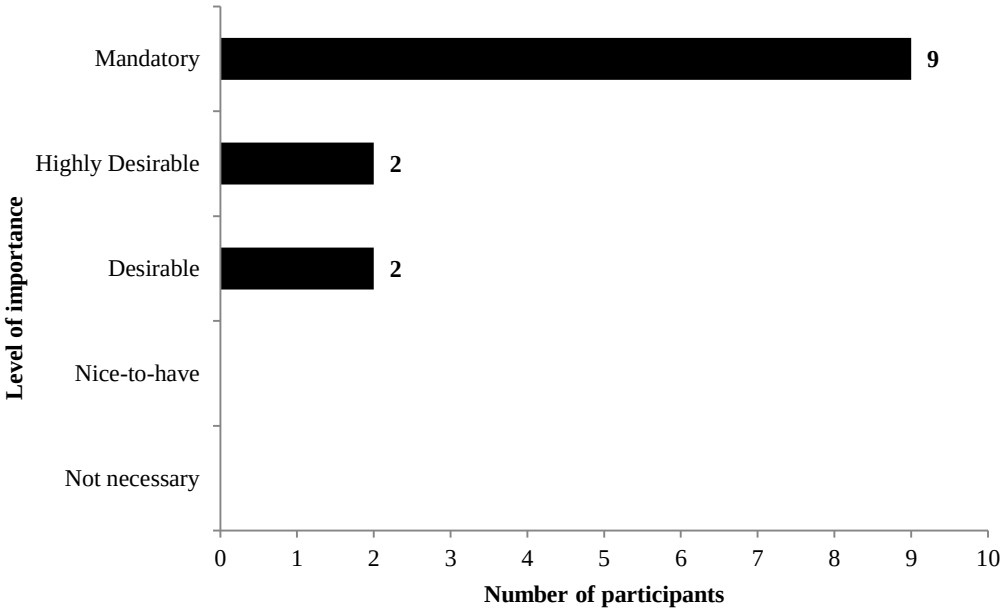


Figure 6-12. Participant ratings for multiple users

6.1.5.13 Document management (F4-F02)

This feature concerns support for document management, which involves managing large collections of papers, studies and the relationships between them. Ratings for this feature are shown in row 17 of Table 6-9 and visualised in Figure 6-13.

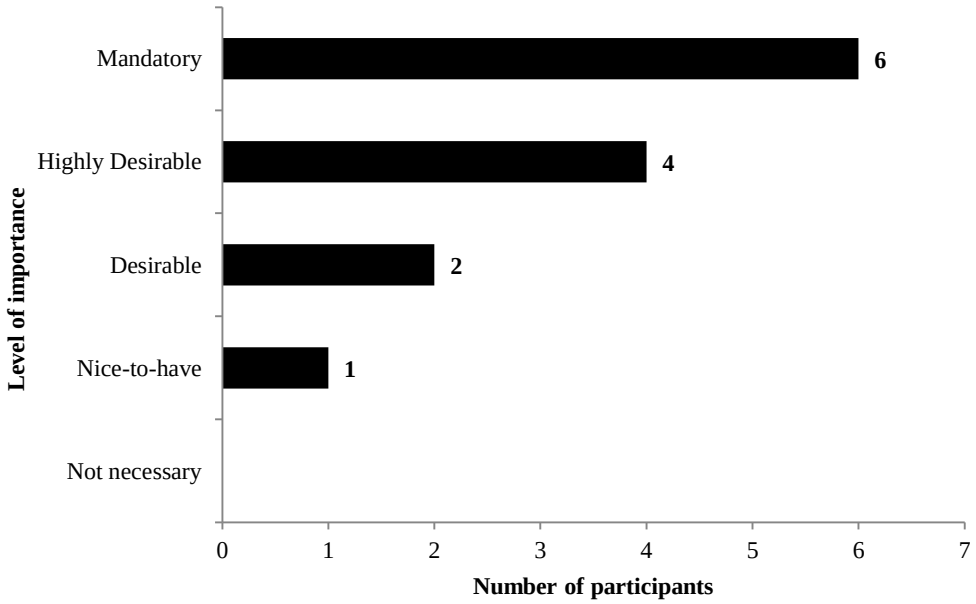


Figure 6-13. Participant ratings for document management

Many participants felt support for document management would be a useful feature. In particular, having the relationships between the papers and studies ***"closely integrated"*** would be ***"really helpful"***. Furthermore, such a feature might help transition the tool from a ***"reference manager to a study-based system"***. A key issue raised by one participant was copyright. With multiple users collaborating and sharing documents, problems concerning permissions/access of certain papers may occur.

6.1.5.14 Security (F4-F03)

This feature concerns security, including a log-in or similar system. Ratings for this feature are shown in row 18 of Table 6-9 and visualised in Figure 6-14.

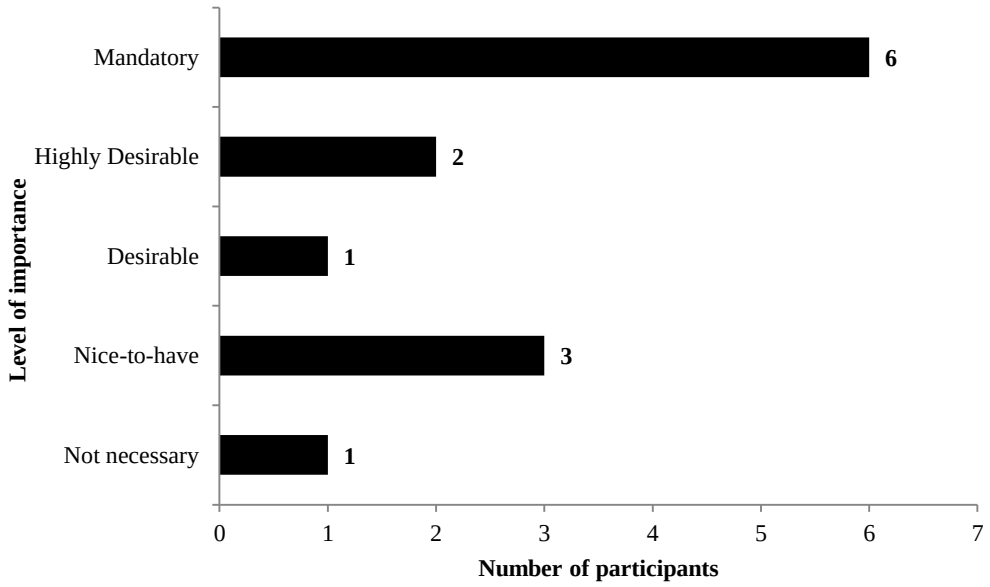


Figure 6-14. Participant ratings for security

Many participants felt a feature, which supports security, should be included within a tool. One participant argued, however, that since systematic reviews deal with ***"published studies"*** that have ***"already been anonymised"***, security wouldn't be necessary. Another participant, however, felt security was important because ***"you might include unpublished stuff that the authors have let you use"***. Similarly, another participant noted that ***"some reviews use industry supplied data, which is not in the public domain"***. Other participants commented that there ***"might be conflict of***

interest between shared researchers" and, therefore, security features would be important to help address this.

6.1.5.15 Role management (F4-F04)

This feature concerns support for role management. It is considered, for example, that a review team is able to state which users will perform certain activities (e.g. study selection, quality assessment, data extraction etc.) and allocate papers accordingly. Ratings for this feature are shown in row 19 of Table 6-9 and visualised in Figure 6-15.

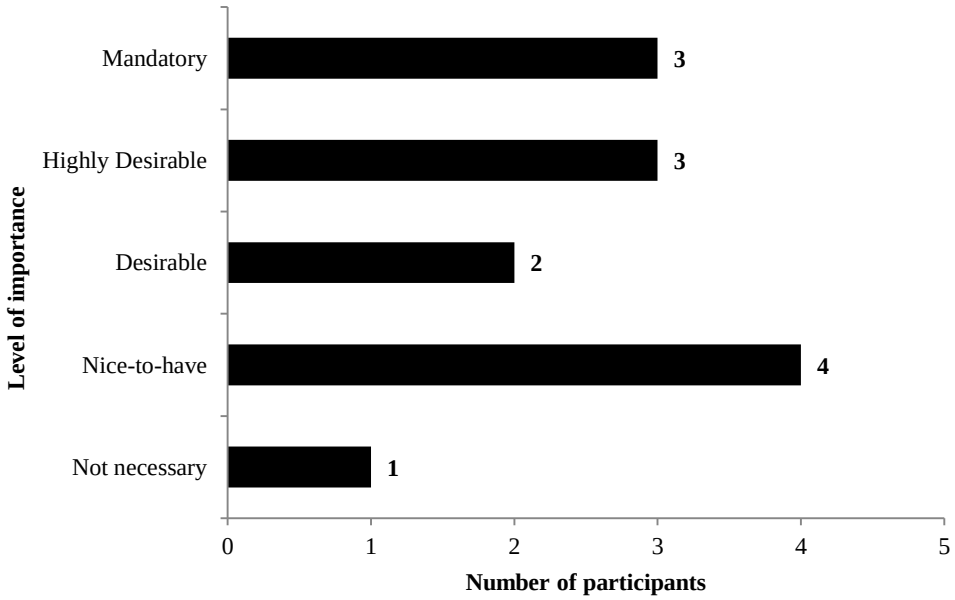


Figure 6-15. Participant ratings for role management

Support for role management where, for example, you could ***"see all the people in the team and what their roles were"*** was generally considered a useful feature. One participant raised concerns about allowing others to see your role and contribution within the project. Another participant, however, points out that ***"it's not necessarily that you don't trust people to do a good job, it would just cut down the chances of a mistake"***. Other participants noted that in smaller teams, this sort of feature might not be so necessary.

6.1.5.16 Re-use of data from past projects (F4-F05)

This feature concerns support for the re-use of data from past systematic reviews in new systematic reviews, or, when updating an existing systematic review. Ratings for this feature are shown in row 20 of Table 6-9 and visualised in Figure 6-16.

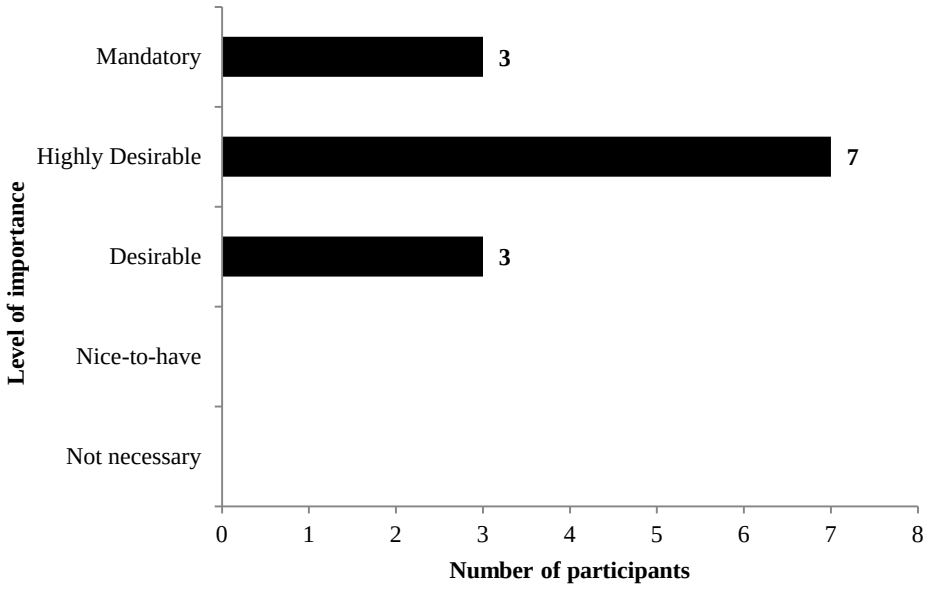


Figure 6-16. Participant ratings for re-use of data from past projects

Many participants felt support for re-using data from past systematic reviews would be useful; particularly, when updating systematic reviews (which ***"is happening more and more now."***) The potential for time-saving was also praised. In particular, speeding up quality assessment by including a previously assessed study (from a past systematic review) might mean that ***"you wouldn't have to quality assess it again"***. Similarly, participants note that it could also help during the search. For example, ***"you run the search and it automatically excludes any paper that was found in a previous systematic review"***.

6.1.5.17 Simple installation and setup (F2-F01)

This feature concerns support for a simple installation and setup procedure, including an installation guide and/or tutorial. Ratings for this feature are shown in row 3 of Table 6-9 and visualised in Figure 6-17.

Some participants felt that without a simple installation process, users would become ***"frustrated with it and they won't want to use it"***. One participant pointed out that you ***"you don't pick your collaborators based on their IT skills"*** and, therefore, a simple installation is important. Other participants, however, felt that ***"if the tool is good enough"***, then even if the installation is difficult, ***"some people are prepared to give it a go and work it out"***.

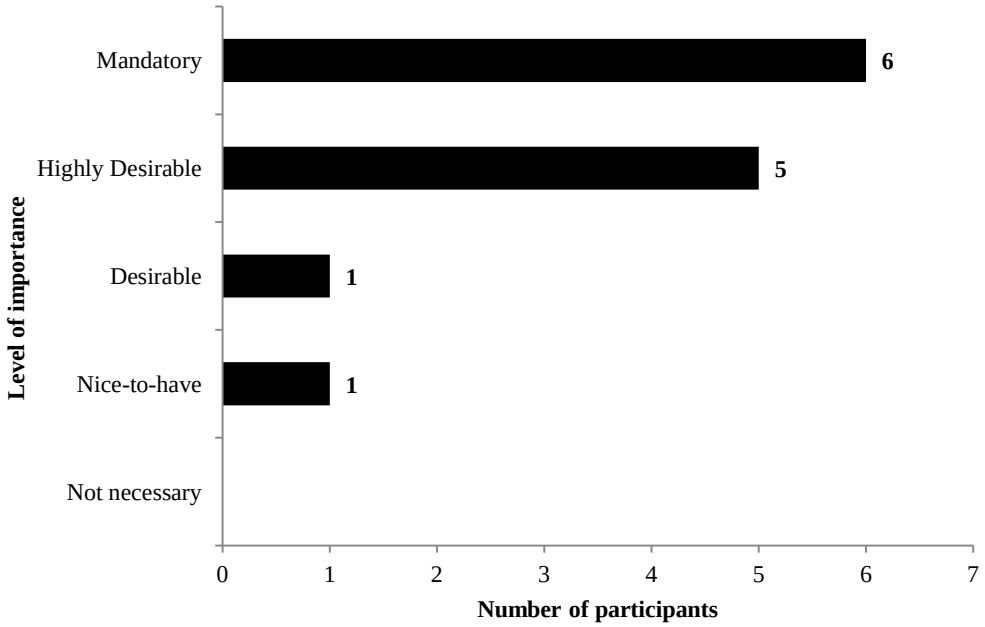


Figure 6-17. Participant ratings for a simple installation and setup

6.1.5.18 Self-contained (F2-F02)

This feature concerns having a tool that is as 'self-contained' as possible i.e. able to function as a stand-alone application with minimal requirements for other external technologies. Ratings for this feature are shown in row 4 of Table 6-9 and visualised in Figure 6-18.

Many participants felt it was preferable for ***"everything to be all-in-one"*** and that, if this was the case, then as a user ***"you are more likely engage with the tool"***. Other participants, however, felt it wasn't an issue and that they'd ***"probably be quite happy installing other packages"***. One participant points out that if the tool ***"does stuff that nothing else can do"*** then you'd put up with having to install other applications.

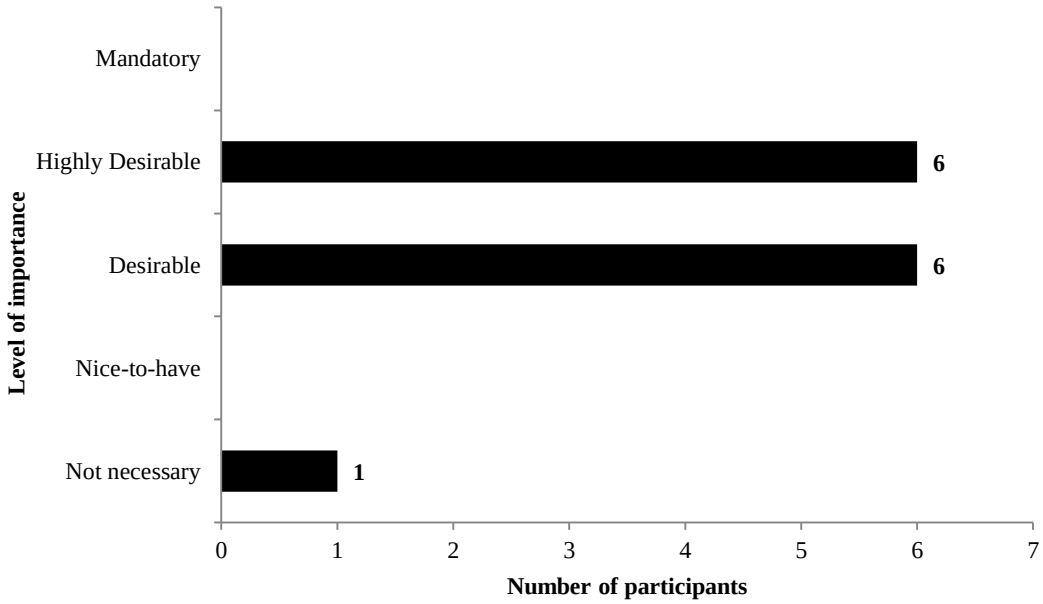


Figure 6-18. Participant ratings for 'self-contained'

6.1.5.19 No financial payment (F1-F01)

This feature relates to the financial cost of the tool. It is considered, for example, that there should be no payment required. Ratings for this feature are shown in row 1 of Table 6-9 and visualised in Figure 6-19.

Some participants thought having the tool ***"free for personal use"*** with ***"different licenses for different*** [types] ***of user"*** would be a good idea. The majority of participants, however, felt having to pay for a tool was not an issue. One participant stated they would be ***"less inclined to use something if it was completely free"*** as they are placing trust in the tool to hold their data (***"which is very valuable in terms of time and effort"***). One participant commented about a lack of confidence in free; specifically, web-based tools, noting that they could ***"disappear tomorrow"***. Another participant suggests ***"trials are the most important things that are free"***.

6.1.5.20 Maintenance (F1-F02)

This feature concerns maintenance. It is considered that a tool should be well maintained by its developers, including regular updates and a single point of contact for users to obtain support if needed. Ratings for this feature are shown in row 2 of Table 6-9 and visualised in Figure 6-20.

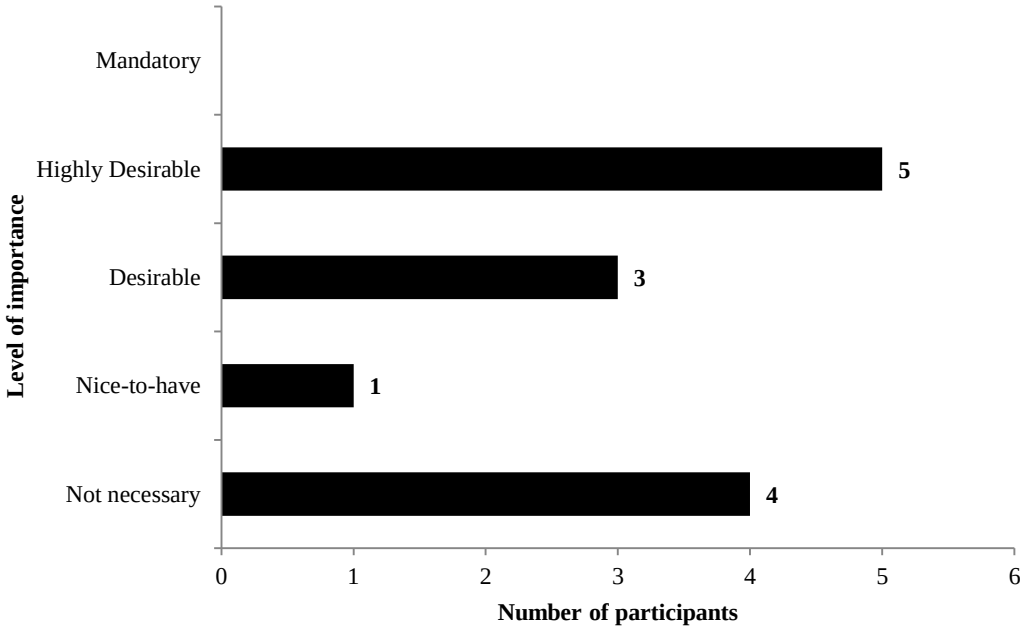


Figure 6-19. Participant ratings for no financial payment

Many participants felt maintenance, post development, was very important as there are ***"bound to be teething problems with something this massive"***. Also, as the ***"systematic review method changes"*** over time, the tool needs to ***"evolve"*** with those changes and bring new features and updates. Another participant mentions that, by having a well maintained tool, ***"it gives people confidence in the tool."*** Similarly, another participant states that users ***"wouldn't invest data in a tool which didn't have any institutional support"***.

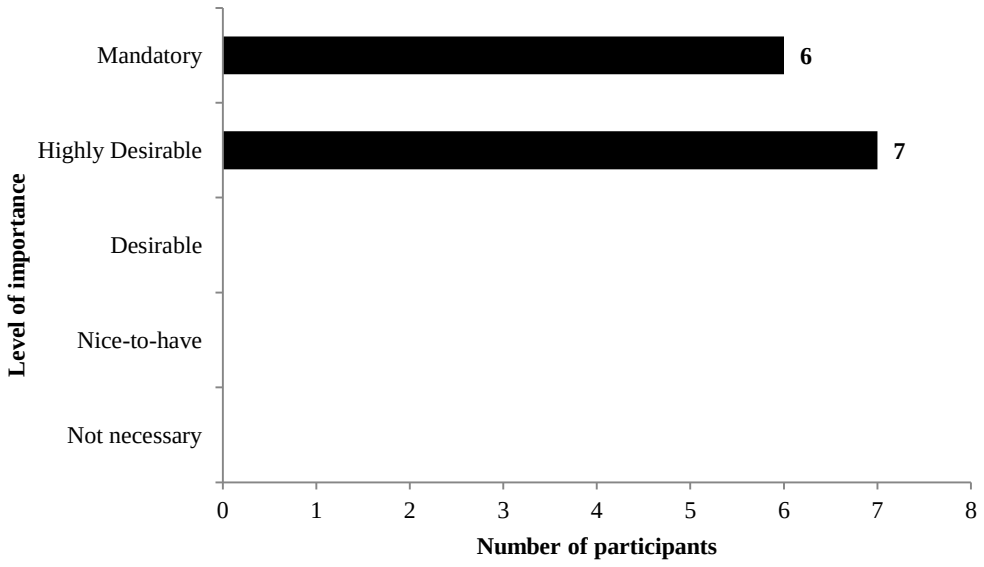


Figure 6-20. Participant ratings for maintenance

6.2 Discussion

This section presents a discussion of the results of the survey highlighting some of its key findings. Limitations of the study are also discussed with some lessons learned from conducting semi-structured interviews. Finally, Implications to the evaluation framework, as a consequence of the survey, are given.

6.2.1 Participants views on the usefulness and challenges of systematic reviews

At the beginning of the interview, participants were asked about the overall usefulness and main challenges associated with systematic reviews. These questions were primarily used as a device to relax the participant and get them talking (Runeson & Host, 2009; Hove & Anda, 2005). However, participant responses also provided useful insight into the systematic review process, in healthcare and social science.

The main positive characteristics about systematic reviews, identified by participants, are summarised in Table 6-2. Participants praised the rigour of the methodology and that it was particularly useful for students and novices entering a new field. Furthermore, systematic reviews were identified as a useful activity for learning about a particular topic and to provide recommendations for future research. Participants also felt systematic reviews were popular amongst researchers for obtaining publications and citations. These views on the usefulness of systematic reviews are shared by researchers in software engineering (Babar & Zhang, 2009; Kitchenham *et al.*, 2009; Santos & Da Silva, 2013).

The main challenges of systematic reviews, identified by participants, are presented in Table 6-3. Participants complained that reviews were time consuming, difficult and error prone. Particular issues with the search process, study selection and quality assessment stages of a systematic review, were highlighted. Some participants also mentioned meta-analysis as a challenging aspect of a systematic review. Generally, many of these problems are shared by researchers in software engineering (Carver *et al.*, 2013). Issues with the search process, for example, which are considered

particularly prevalent in software engineering (Brereton *et al.*,2007; Carver *et al.*, 2013), were frequently expressed by participants as well. Some issues, however, such as difficulties surrounding meta-analysis, are not shared. In software engineering, meta-analysis is seldom undertaken and is, therefore, not considered a priority issue within the domain.

6.2.2 Tools identified by participants

A variety of tools were identified by participants. As shown in Table 6-4, the most common type of tool identified by participants were reference managers. The systematic storage and management of citations is a critical part of any systematic review (in any domain) and it was, therefore, unsurprising that these types of tool were mentioned most frequently.

Interestingly two custom-built tools were reported. These tools (i.e. a web-based coding tool that supports collaborative study selection and a customised excel add-in that supports analysis), were developed by their respective review teams, as they felt that available tools did not provide sufficient support for the complexity of their reviews. It may be, however, that suitable tools were available but were not known to the teams. A web-based catalogue, the *Systematic Review Toolbox*, which aims to help reviewers identify tools to support systematic reviews, has since been developed. Details of this resource are reported in Chapter Four.

6.2.3 Participant feature ratings

The set of features, ranked by level of importance, are shown again in Table 6-10. Features considered by participants to be particularly important (i.e. features that received many ratings of Mandatory or Highly Desirable) include support for multiple users, data extraction and maintenance. Clearly, collaboration is a key aspect of systematic reviews and is recommended for many stages in the process (e.g. study selection, quality assessment and data extraction) to ensure maximum reliability and validity.

Some features generated a wide range of opinions and, thus, resulted in little consensus amongst participants. In particular; support for role management, developing the review protocol and

id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F4-F01	Multiple users	**9**	2	2	0	0	Mandatory
F3-F06	Data extraction	**7**	5	1	0	0	Highly Desirable
F1-F02	Maintenance	6	**7**	0	0	0	Highly Desirable
F2-F01	Simple installation and setup procedure	**6**	5	1	1	0	Highly Desirable
F4-F02	Document management	**6**	4	2	1	0	Mandatory
F4-F03	Security	**6**	2	1	3	1	Desirable
F3-F05	Quality assessment and validation	5	**7**	1	0	0	Highly Desirable
F3-F07	Data synthesis	5	**7**	1	0	0	Highly Desirable
F3-F04	Study selection and validation	5	**6**	2	0	0	Highly Desirable
F3-F09	Meta-analysis	4	**5**	2	2	0	Nice-to-have
F4-F05	Re-use of data from past projects	3	**7**	3	0	0	N/A
F3-F03	Supports automated searches	3	**4**	3	3	0	Highly Desirable
F4-F04	Role management	3	3	2	**4**	1	Highly Desirable
F3-F01	Development of review protocol	2	**4**	2	3	2	Desirable
F3-F02	Protocol validation	1	1	**5**	1	**5**	Desirable
F2-F02	Self-contained	0	**6**	**6**	0	1	Highly Desirable
F1-F01	No financial payment	0	**5**	3	1	4	Highly Desirable
F3-F11	Report validation	0	3	3	3	**4**	Nice-to-have
F3-F08	Text analysis	0	3	2	**5**	3	Nice-to-have
F3-F10	Report write-up	0	2	**6**	4	0	Nice-to-have

Table 6-10. Summary of participant ratings for each feature ranked by importance

validation fall into this category. It was checked whether the lack of consensus could be explained by participants' different experience levels or areas of work. However, no patterns relating to these factors were found. Another possible explanation could stem from the fact that although some participants thought that tool support for a particular stage would be useful, they gave it a low rating because they were not able to imagine how such support could be provided (e.g. ***"I have a hard time seeing how that would work properly."*** and ***"it would be highly difficult to automate all that."***). The issue of financial payment for a tool (or, rather, lack of) also received varying opinions

amongst participants. It had been assumed that having a tool free of financial cost would be a positive characteristic. Results show, however, that many participants suggest some payment for a tool provides a degree of confidence in the reliability and longevity of the tool (see Section 6.1.5.19). One participant notes that ***"trials are the most important things that are free"***.

Features not considered particularly important include support for writing the report, text analysis and report validation. Therefore, results seem to suggest that tool support for the reporting phase of a systematic review is not a high priority for reviewers.

6.2.4 Implications for the evaluation framework

This section discusses the implications for the evaluation framework of the results of this study. Section 6.2.4.1 compares the features and importance levels identified by participants in the survey compared with those in version 1.1 of the evaluation framework (described in Section 5.1). Refinements to the framework are presented in Section 6.2.4.2.

6.2.4.1 Comparing the features

Generally, there was a good level of agreement between the ratings of the survey participants and the ratings of version 1.1 of the evaluation framework. Comparing the modal value from the 13 participants with the ratings in the framework, there were no differences for 11 features (see Table 6-11) and only slight differences (i.e. one level of importance higher or lower) for five features (see Table 6-12). Results suggest, therefore, that many of the frequently raised difficulties faced by reviewers (e.g. time consuming, labour intensive etc.) are shared by reviewers in most domains. Clearly there is considerable commonality between systematic reviews in software engineering and other disciplines, so it is not surprising that there is some agreement about the importance of tool features. There are, however, notable differences relating to three features; namely, support for meta-analysis, role management and security (see Table 6-13).

id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F4-F01	Multiple users	**9**	2	2	0	0	Mandatory
F1-F02	Maintenance	6	**7**	0	0	0	Highly Desirable
F4-F02	Document management	**6**	4	2	1	0	Mandatory
F3-F05	Quality assessment and validation	5	**7**	1	0	0	Highly Desirable
F3-F07	Data synthesis	5	**7**	1	0	0	Highly Desirable
F3-F04	Study selection and validation	5	**6**	2	0	0	Highly Desirable
F3-F03	Supports automated searches	3	**4**	3	3	0	Highly Desirable
F3-F02	Protocol validation	1	1	**5**	1	**5**	Desirable
F2-F02	Self-contained	0	**6**	**6**	0	1	Highly Desirable
F1-F01	No financial payment	0	**5**	3	1	4	Highly Desirable
F3-F08	Text analysis	0	3	2	**5**	3	Nice-to-have

Table 6-11. Summary of participant ratings where there were no differences

As shown in Table 6-12, the modal value of the responses of participants for a feature which supports meta-analysis indicates a 'Highly Desirable' level of importance. From a software engineering perspective in our evaluation framework, this feature was considered only 'Nice-to-Have.' This can be explained by the fact that few meta-analyses are undertaken within the software engineering domain because primary studies seldom report comparable results. In healthcare, however, where reviewers often extract and analyse data from randomized controlled trials, synthesis tools and, in particular, meta-analysis tools are more important. Support for this feature can therefore be considered context-dependent. A context-dependent feature is determined, where its relative importance is influenced by the particular systematic review-related issues with, or characteristics of, a specific domain. In this case, it is suggested that the prevalent type of primary study used in a particular domain, influences the importance of tool support for meta-analysis.

There were also differences concerning support for security and role management (see Table 6-13). From a software engineering perspective, support for role management was rated as 'Highly Desirable.' The modal response from survey participants, however, rated this feature as 'Nice-to-

Have.' This was somewhat surprising since support for multiple users (i.e. collaboration) was rated highly by both software engineering researchers and participants in this study. It was expected, therefore, that being able to manage those users within the context of a review would be important to users in other domains as well.

id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F3-F06	Data extraction	**7**	5	1	0	0	Highly Desirable
F2-F01	Simple installation and setup procedure	**6**	5	1	1	0	Highly Desirable
F3-F01	Development of review protocol	2	**4**	2	3	2	Desirable
F3-F11	Report validation	0	3	3	3	**4**	Nice-to-have
F3-F10	Report write-up	0	2	**6**	4	0	Nice-to-have

Table 6-12. Summary of participant ratings where there were slight differences

For security, software engineering researchers rated this as a 'Desirable' feature. The modal response from participants, however, considered security features as 'Mandatory.' This higher level of importance might be explained by the, sometimes, sensitive nature of data (see Section 6.1.5.14) that is included in a systematic review (i.e. patient or industry related data). This, again, may be an example of a context-dependent feature. Furthermore, it should be noted that in both cases (i.e. security and role management), responses were spread fairly evenly over most of the categories. This is another indication of a context-dependent feature. Other features, showing a similar pattern are support for the development of the review protocol, report validation and text analysis.

id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F4-F03	Security	**6**	2	1	3	1	Desirable
F3-F09	Meta-analysis	4	**5**	2	2	0	Nice-to-have
F4-F04	Role management	3	3	2	**4**	1	Highly Desirable

Table 6-13. Summary of participant ratings where there were notable differences

6.2.4.2 Refinements to the framework

The results of the survey suggest two modifications to version 1.1 of the evaluation framework. The changes relate to:

1. **Evaluating support for text analysis (F3-F08) has changed.**

The low rating (see Section 6.1.5.8) and lack of consensus amongst participants (see Section 6.2.4.1) for text analysis have motivated a change in how the feature is presented and assessed by the framework. It was considered that version 1.1 of the evaluation framework did not adequately reflect how text analysis can support multiple stages of a systematic review. To address this, text analysis (F3-F08) has been removed as a separate feature and will instead form part of the suggested assessment criteria used to evaluate a tool's search, study selection, data extraction and data analysis features.

2. **A level of importance has been determined for supporting the re-use of past systematic review data (F4-F05).**

A feature which supports reusing past systematic review data was introduced following changes made to version 1.0 of the evaluation framework (see Section 3.5.2). A level of importance, however, was not yet determined, due to limited evidence about reusing data from past systematic reviews in software engineering. The majority of participants in the survey rated this feature as highly desirable. Therefore, this level of importance has been selected.

These refinements have led an updated version of the framework (version 1.2). The updated set of features and weightings are presented in Table 6-14. Changes made to version 1.1 of the evaluation framework are further examined in Section 7.3.1.2.

6.2.5 Limitations of the survey

Semi-structured interviews rely heavily on the communication skills of the interviewer (Clough & Nutbrown, 2012). It is possible, therefore, that the quality of the data collected is limited by the

interviewer's lack of experience. This problem was at least partially addressed by performing a pilot interview (see Section 5.3.6).

id	Feature Set	id	Feature	Importance Weighting
F1	Economic	F1-F01	No financial payment	Highly Desirable
		F1-F02	Maintenance	Highly Desirable
F2	Ease of introduction and setup	F2-F01	Simple installation and setup.	Highly Desirable
		F2-F02	The tool is self-contained.	Highly Desirable
F3	Systematic reviews activity support	F3-F01	Protocol development	Desirable
		F3-F02	Supports automated searches	Highly Desirable
		F3-F03	Study selection and validation	Highly Desirable
		F3-F04	Quality assessment and validation	Highly Desirable
		F3-F05	Data extraction and validation	Highly Desirable
		F3-F06	Data synthesis	Highly Desirable
		F3-F07	Meta-analysis	Nice-to-have
		F3-F08	Report write-up	Nice-to-have
F4	Process management	F4-F01	Support for multiple users	Mandatory
		F4-F02	Document management	Mandatory
		F4-F03	Security	Desirable
		F4-F04	Management of roles	Highly Desirable
		F4-F05	Re-use of data from past projects	Highly Desirable

Table 6-14 Set of features from version 1.2 of the evaluation framework

Furthermore, research suggests that people respond differently depending on how they perceive the interviewer ('the interviewer effect') (Denscombe, 2010). Factors such as gender, age and the ethnic origins of the interviewer have a bearing on the amount of information people are willing to contribute (Denscombe, 2014). In addition, participant's responses can be influenced by what they think the situation requires (Gomm, 2004). To try to address this, every effort was made to put participants at ease and to explain the purpose and the topics to be covered by the interviews.

Another risk associated with adopting a semi-structured interview format, is that topics may be inadvertently missed (Patton, 2005). Comparability between interview data may be reduced because the sequencing and wording of questions may be slightly different for each interview. To

help address this, a script was produced which highlighted the key themes to explore and questions to ask.

6.2.6 Semi-structured interviews – Lessons learned

Hove and Anda report on the experiences of conducting semi-structured interviews in empirical software engineering research (Hove & Anda, 2005). They call on others to share their experiences of using the method "in order to increase the probability of collecting measures of high quality." In this section, the interviewer's experience using semi-structured interviews, with some lessons learned, is presented.

It was considered important that the interviewer was knowledgeable about the subject under investigation. The interviewer (a second year PhD student at the time) had performed a mapping study (Marshall & Brereton, 2013) and feature analysis (Marshall *et al.*, 2014) related to the topic. This knowledge allowed the interviewer to understand the information provided by participants, to ask relevant follow-up questions, to clarify ambiguities and to maintain control of the interview (Hove & Anda, 2005).

The quality of the data collected using interviews relies on the skills of the interviewer (Clough & Nutbrown, 2012). Performing a pilot (or trial) interview beforehand was important since "these skills are developed mainly through practice" (Hove & Anda, 2005). Although interviews were recorded using a digital audio recorder, the interviewer also took written notes. However, note taking during the interview felt, at times, distracting. The interviewer often needed to pause between questions to catch up with the written notes, which interrupted the flow of conversation. During analysis, it became clear that the audio recordings, once converted to written transcripts, were the more valuable data source. The written notes were, however, still useful as a backup.

Hove and Anda commented on and recommended the use of visual artifacts (Hove & Anda, 2005). There were two circumstances where a visual artifact was used during an interview. In both cases, the interviewer produced a simple illustration to help explain an example implementation of a

feature; specifically, a feature to support the search process. This was appreciated by the participants and helped them to visualise how the feature would work.

6.3 Summary and Conclusions

The study reported in this chapter has explored the experiences and opinions of systematic reviewers in healthcare and social science domains, with a particular focus on their use of and views about automated tools to support systematic reviews; using, an interview-based survey.

In relation to the first goal of the study, participants identified 21 software tools (see Table 6-4, which were each categorised into one of seven groups; namely, reference management tools, special-purpose tools, basic productivity tools, advanced analysis software, custom-built, meta-analysis and other (i.e. unclassified) tools. Reference management tools were the most commonly mentioned forms of automated support. Special-purpose tools (i.e. *RevMan* and *EPPI-Reviewer*) were the second most common. Since many problems relating to systematic reviews (and mapping studies) faced in other disciplines are similar to those faced by software engineering researchers, it may be that *EPPI-Reviewer* (and *RevMan*) could be used within software engineering too.

Addressing the second goal of the study, the top three most important features classified by participants were support for multiple users, data extraction and maintenance. The three least important features for a tool were support for writing the report, text analysis and report validation.

To address the third goal, the importance levels of features identified by participants were compared with ratings from a software engineering perspective (i.e. the ratings used in the feature analysis in Chapter Three). Generally, there was a good level of consensus, with only a small number of notable differences; specifically, ratings for meta-analysis, role management and security. However, it is noted that anyone, wanting to use the evaluation framework to assess support tools for their own use, should take care to determine the importance of context dependent features for their own circumstances, rather than using the suggested weightings. This point is discussed further in the following chapter (Chapter Seven) and Chapter Eight.

Based on the results of and experience gained from the survey, additional refinements were made to the set of features, forming part of the evaluation framework (see Section 6.2.4.2). Version 1.2 of the evaluation framework will undergo further validation, reported in Chapter Seven.

Chapter Seven

Discussion

In this chapter, the findings from all of the research undertaken and reported in this thesis are brought together and discussed in relation to the original research questions. The development of an evaluation framework for tools to support systematic reviews in software engineering is discussed. The most recent version of the framework is presented, validated and used to evaluate a new tool aiming to support the whole systematic review process in software engineering.

7.1 Introduction

The overall aim of this thesis was to investigate the usefulness and development of tools that provide support for the systematic review process in software engineering. As part of this investigation, an evaluation framework for tools to support systematic reviews in this domain was developed, refined and validated.

Two research questions were established to direct the focus of this project:

> **RQ1 - Can tools provide useful support when undertaking a systematic review in software engineering?**
>
> **RQ2 – What are the most important features of tools to support systematic reviews in software engineering?**

In this chapter, the findings of this research are brought together and discussed in relation to the research questions. Focusing on addressing RQ1, the findings from work undertaken to investigate the usefulness of tools are discussed in Section 7.2. A summarised response to RQ1 is provided in Section 7.2.4. In response to RQ2, the features of an overall systematic review support tool are established and presented as part of an evaluation framework in Section 7.3. The development of the evaluation framework is discussed and the latest version presented as part of a final comparative assessment of two overall support tools. A summarised response to RQ2 is provided in Section 7.3.3. Recommendations to assist both tool developers and users, along with suggestions for future work, are outlined in Chapter Eight.

7.2 Tool Support for Systematic Reviews

In this section, a discussion of the work undertaken to understand the current state of tools that support systematic reviews, is presented. Research activities, undertaken to investigate tool support, involved the following:

- **Literature review** to identify and classify systematic review tools in software engineering (see Section 7.2.1).
- **Feature analysis** to compare and evaluate a selection of overall support tools for software engineering reviewers (see Section 7.2.2).
- **Cross-domain survey** to explore tool support for systematic reviews in other domains (see Section 7.2.3).

The findings of all these activities are brought together and discussed in a summarised response to RQ1 in Section 7.2.4.

7.2.1 Literature review

A literature review, reported in Chapter Two, was undertaken to identify and classify tools that support systematic reviews in software engineering. A variety of tools supporting various aspects of the systematic review process were found and discussed. The results showed a small but encouraging growth of tools to support systematic reviews. In particular, visualisation and text mining tools, which aimed to support study selection, data extraction and data synthesis, were the largest cluster. Due to the novelty of the field, however, limited primary data on the effectiveness of tools was able to be obtained. In fact, only two substantial evaluation studies of tools were found. These studies were very positive about their respective tools and highlighted their effectiveness for supporting systematic reviews. Most tools, however, had received limited evaluation and, generally, only speculation over their potential was reported. As a consequence, only expected benefits about tools (and associated costs) could be extracted and analysed in the

literature review. The benefits of tools seemed to reflect some of the main challenges of performing systematic reviews, as reported by researchers in the literature. Many tools focused on reducing the overall time and effort involved with undertaking a systematic review. These are frequently stressed concerns by reviewers and it was positive to find tools being developed to help address them. However, as explored in Chapter One (see Section 1.1.5), there are many other important issues faced by researchers in software engineering when undertaking a systematic review. Support for the planning phase, search, quality assessment and report phase of a systematic review were largely absent from the majority of tools identified in the literature. There were, however, a small selection of tools found, which aimed to support all (or at least the majority of) stages of a systematic review in software engineering. Based on the results of the literature review, there remained scope to perform an independent evaluation of these tools. The literature review made the following contributions to the project:

- Available tools to support systematic reviews in software engineering were identified and classified, based on various characteristics.
- Insight into the current potential and usefulness of tools was provided.
- The degree to which tools had been evaluated was established.

7.2.2 Feature analysis

Following the literature review, four tools aiming to support the whole systematic review process in software engineering, were independently evaluated. The evaluation took the form of a feature analysis (which is part of the DESMET methodology for evaluating methods and tools), which can be classified as a type of multi-criteria decision analysis. In order to perform the study, an initial framework, comprised of a set of required features, weightings and scoring instruments, was developed to assess the tools. Tools were mainly assessed based on how well they provided support for each phase of a systematic review, and the steps within these phases. Other aspects of tools, such as economic factors and ease of setup, were also evaluated, but were considered less

important. The results of the feature analysis highlighted some strengths and weaknesses of each tool and also identified the strongest (and weakest) candidate.

A number of limitations were found to be common across all (or most) of the candidate tools. Support for protocol development, by most tools, was generally quite limited. Only one tool, *SLR-Tool*, was considered to assist this stage effectively. In addition, support for the search process (a frequently stressed issue within the community), was also found to be largely absent. *SLRTOOL* was the only candidate that provided an internal search facility for querying digital libraries. However, as noted in Chapter Three, its implementation was rather limited. It is considered in this thesis that poor support for this aspect of a systematic review may be a consequence of the inherent difficulties associated with automated searching in software engineering (Brereton *et al.*, 2007; Dieste *et al.*, 2009; Bailey *et al.*, 2007; Dyba *et al.*, 2007). As mentioned in Section 1.1.5, interoperability between electronic resources is, currently, very limited. Each digital library usually has some variation regarding the format, syntax and vocabulary required for search strings to be used. This makes searching consistently across multiple resources challenging. It may be that these 'higher-level' issues need to be addressed first before effective tool support can be realised. Support for collaboration, within a team-based systematic review, was also found to be limited. Only one tool, *SLuRp*, provided reasonably effective facilities for collaboration, amongst multiple users. Given the collaborative nature of a systematic review, this limitation amongst tools was, again, surprising. This is because many stages of the process, such as study selection, data extraction and quality assessment, are strongly recommended to be undertaken by multiple researchers. This suggests that a deeper understanding of the systematic review process and; in particular, what are considered the collaborative activities, may be needed in order to develop effective support. Undertaking this study made the following contributions to the project:

- Further insight was provided into the usefulness of tools to support systematic reviews in software engineering. In particular, tools aiming to support the whole systematic review process were compared and evaluated, independently, for the first time.

- The feasibility of an evaluation framework for tools that provide support for the whole systematic review process in software engineering was investigated.
- The framework was refined based on the results of, and experience gained from undertaking the feature analysis.

7.2.3 Tools in other domains

The literature review and feature analysis both provided insight into the current usefulness and development of tool support for systematic reviews. However, these studies focused only on tool support for systematic reviews undertaken within software engineering. As discussed throughout this thesis, systematic reviews are undertaken in many different domains. The methodology is very established, for example, in areas of healthcare and social science. Exploring tool support for systematic reviews in other domains was, therefore, considered necessary, in order to obtain a deeper understanding about the technology (Brown & Wallnau, 1996). A survey was designed (see Chapter Five) and implemented to interview researchers in healthcare and social science about tool support for systematic reviews.

A variety of tools were identified by participants. The most common type of tool identified were reference managers. In particular, *RefWorks* and *EndNote* were mentioned most often. *RefWorks* was praised for its ability to manage the search, aid study selection and remove duplicates. Weaknesses noted were limited export capabilities and issues with usability and citation formatting. Strengths of *EndNote* were its web-based remote access, support for study selection and managing multiple projects. Weaknesses identified were concerns over support for large-scale collaborative reviews and issues with usability. Whilst it was unsurprising to find reference managers as the most common form of tool support, it was interesting to learn how they were being used by researchers to support their systematic reviews. Typically, reference managers are used for the systematic management of papers, studies and citations. However, participants highlighted how they were able to use these systems to support other aspects of a systematic review, such as managing the search and study selection. These aspects of a systematic review are not directly

supported by *EndNote* or *RefWorks*. However, the ways in which participants used them to support these stages, were considered some of the most useful points about the tools.

The second most common types of tool identified by participants were special-purpose systems. These tools are similar to those evaluated in the feature analysis and are designed to support the overall systematic review process. Two tools, *EPPI-Reviewer* and *RevMan*, were identified by participants. *EPPI-Reviewer* focuses, primarily, on supporting systematic reviews undertaken in areas of social science. *RevMan*, however, aims to support systematic reviews in healthcare. The main strengths of *EPPI-Reviewer* include support for text mining and qualitative analysis. Some participants, however, mentioned that *EPPI-Reviewer* had a steep learning curve and was difficult to use. *RevMan* was praised for its effective support for meta-analysis and, also, for developing the review protocol. However, a selection of participants mentioned how some potentially useful features (of *RevMan*), were unavailable, unless they were undertaking a Cochrane Review (i.e. a specific type of systematic review commissioned in healthcare). Participants raised issues about the usability of *RevMan* as well.

Two custom-built tools were identified. One system aimed to support collaborative, independent, study selection. The other tool; an add-on for excel, aimed to support meta-analysis. As discussed in Section 6.2.2, the development of these tools may be explained based on the following factors:

- Participants felt that current tools did not provide useful support.
- Appropriate (and potentially useful) tools may well have been available, but were difficult for participants to find.

Similar to those identified in Chapter Two, there are a number of tools which target support for systematic reviews in healthcare and social science domains. Tools such as *Abstrackr*[1], *GAPScreener*[2] and *Rayann*[3] all aim to support the study selection stage of a systematic review.

[1] http://abstrackr.cebm.brown.edu/account/login
[2] http://www.hugenavigator.net/HuGENavigator/HNDescription/opensource_GAP.htm
[3] http://rayyan.qcri.org/

Furthermore, a range of tools, add-ons and extensions (e.g. *MIX 2.0*[4], *MetaEasy*[5] and *MetaXL*[6]) exist, which support meta-analysis. It is more likely, therefore, that participants were not aware of available tools that provide support for these aspects of a systematic review. This may also help explain why some participants were found to arguably stretch the functionality of reference managers, in order to assist with more complex aspects of their systematic review. The development of the *Systematic Review (SR) Toolbox* aims to address this issue *SR Toolbox* (see Chapter Four) is a web-based catalogue of tools, which aims to help researchers find appropriate tools to support their systematic reviews, based on their needs (Marshall & Brereton, 2015).

Project CAST and Transform

In 2015, the development of a new tool involving developers of several major systems, which already support systematic reviews in healthcare and social science, was announced[7]. The Cochrane Author Support Tool (*CAST*) combines features from *EPPI-Reviewer*, *RevMan* and *Covidence*[8] (an additional tool which offers support for many stages of a systematic review) to support systematic reviews in healthcare; specifically, Cochrane Reviews.

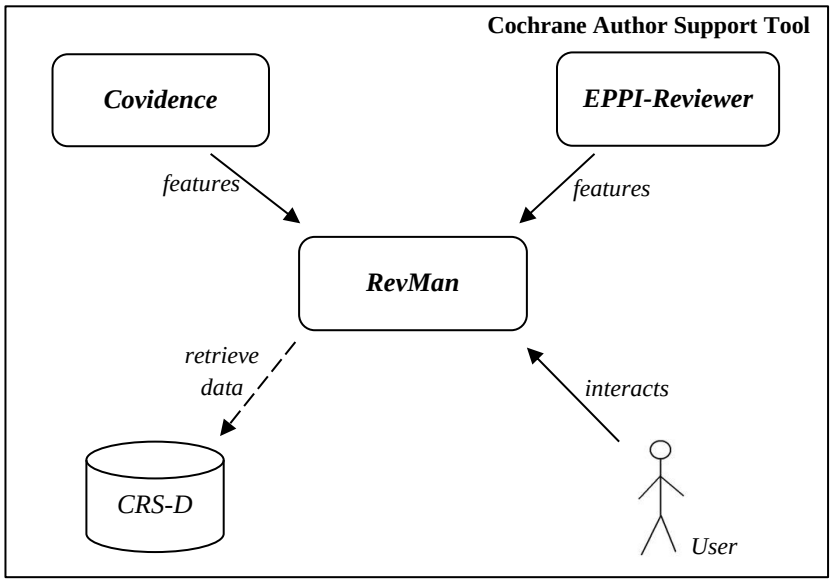


Figure 7-1. Cochrane Author Support Tool (*CAST*) architecture

[4] http://www.meta-analysis-made-easy.com/index.html
[5] http://www.statanalysis.co.uk/meta-analysis.html
[6] http://www.epigear.com/index_files/metaxl.html
[7] http://tech.cochrane.org/news/introducing-cochrane-author-support-tool
[8] http://www.covidence.org/

As shown in Figure 7-1, users interact with *CAST* using a web-based version of *RevMan* (currently in development), which inherits features from *Covidence* and *EPPI-Reviewer*. Features from *Covidence* include support for automated analysis, study selection and quality assessment. *EPPI-Reviewer* brings support for qualitative analysis (e.g. thematic synthesis) to *CAST*, as well as improved support for meta-analysis. The Cochrane Register of Study Database (CRS-D) acts as a central repository for primary study data, which can be queried and extracted into a systematic review developed within the *CAST* environment.

The development of *CAST* is part of a wider initiative in healthcare to integrate technology with evidence. Project Transform[9] aims to work with researchers and developers to improve the way people, processes and tools come together to produce evidence. Its focus is to address four key challenges inherent in evidence production:

1. Finding relevant research in a timely and reliable way.
2. Developing pathways for potential new contributors.
3. Increasing the efficiency of working collaboratively.
4. Ensuring content is relevant and up to date.

To help address these challenges, the team behind Project Transform encourages the development and use of:

- Automated search tools (to help locate relevant studies).
- Machine learning technologies (to intelligently distribute work to appropriate members of a review team).
- Online, social networking techniques (to improve and encourage collaboration).

The development of *CAST* is a key step toward achieving this level of support. Project Transform encompasses a community-driven effort to improve the effectiveness of tools to support systematic reviews within healthcare. Furthermore, it follows a period in tool development (and usefulness)

[9] http://community.cochrane.org/transform

within the domain, not unlike the current state of tools to support systematic reviews in software engineering. It may be, therefore, that a similar initiative could be established, which brings together current developers and researchers within evidence-based software engineering to help push forward tool development.

7.2.4 Summarised response to RQ1

Can tools provide useful support when undertaking a systematic review in software engineering?

The findings of this research have determined that tools to support systematic reviews in software engineering can provide useful support for systematic reviews in software engineering. General purpose tools such as spreadsheets and reference managers are already useful. The quality of support provided by special-purpose tools, however, is yet to be fully determined. This is because these tools are still in the early stages of development, usage and investigation, and are not yet widely used. To-date, the majority of work in this area has focused on developing tools to support particular stages of a systematic review. The literature review showed that whilst the potential for tools is high, work undertaken to assess their effectiveness, has been limited. The feature analysis provided additional insight into the potential of tools; particularly, those that support the overall systematic review process. However, more work is needed to determine their effectiveness. In particular, it is important that researchers in software engineering begin to employ more special-purpose tools to support their systematic reviews (see Section 8.3). Feedback from researchers, as to whether tools benefited (or hindered) the conduct and quality of their systematic reviews, would provide a valuable contribution to this area. In other domains, similar work is being undertaken to investigate tool support for systematic reviews. The survey identified ways in which tools are being used to support systematic reviews in healthcare and social science. Similar to the state of tools in software engineering, the use of novel, special-purpose tools was limited, with a focus on adapting traditional systems to support challenging aspects of systematic reviews. Tools similar to those evaluated in the feature analysis were also identified and their strengths and weaknesses explored. However, the effectiveness of these tools was still not clear.

7.3 The Evaluation Framework

In this section, an evaluation framework for an overall tool to support systematic reviews in software engineering is presented and validated. The framework is composed of a set of features, levels of importance (i.e. weightings) and scoring instruments used to assess the usefulness of tools. Details of earlier versions of the framework and any refinements made are provided in Section 7.3.1. In Section 7.3.2, version 1.2 of the evaluation framework is checked against the newest version of the guidelines for performing systematic reviews in software engineering (Kitchenham *et al.*, 2015). Finally, the newest version of the framework (version 1.3) is presented and validated as part of a new tool evaluation in Section 7.3.3.

7.3.1 Earlier versions of the evaluation framework

As part of this project, an evaluation framework for tools which support the whole systematic review process in software engineering has been created. During its development, four versions were established. These include:

- **Version 1.0** – a preliminary framework comprised of a set of features, weightings and scoring instruments, developed to perform the feature analysis. Changes made to this version of the evaluation framework are discussed in Section 7.3.1.1.
- **Version 1.1** – a refined version of the framework, based on the results of and experience gained from the feature analysis. Changes made to this version of the evaluation framework are discussed in Section 7.3.1.2.
- **Version 1.2** – a further refined version of the framework, motivated by the conduct of a cross-domain survey. Changes made to this version of the evaluation framework are discussed in Section 7.3.1.3.
- **Version 1.3** – an updated version, based on checking the framework against the new guidelines for performing systematic reviews in software engineering. This framework is presented and used in a comparative evaluation in Section 7.3.3.

Where appropriate, aspects of the framework have been modified between versions. In this section, changes made to the first three versions (i.e. version 1.0, 1.1 and 1.2) of the evaluation framework are discussed. The most recent version of the framework (version 1.3) is presented as part of a new tool evaluation in Section 7.3.3.

7.3.1.1 Changes made to version 1.0 of the evaluation framework

The first version of the evaluation framework (version 1.0) was formed to carry out a comparative assessment of tools aiming to support the overall systematic review process in software engineering. As described in Section 3.3.2 (and visualised in Figure 7-2), the initial features and levels of importance (see Table 7-2) were determined based on various factors relating to systematic reviews and their undertaking, within software engineering.

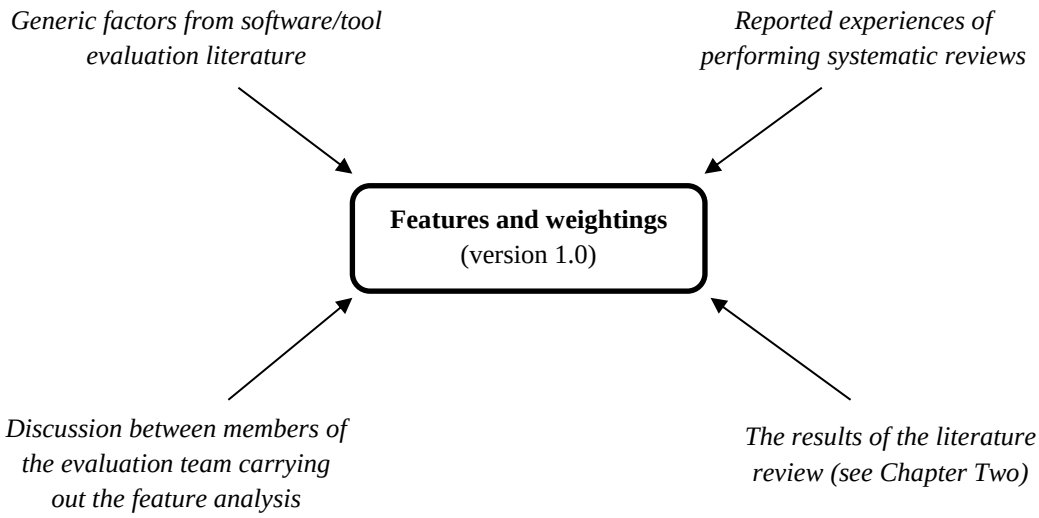


Figure 7-2 Factors influencing the development of the features and importance levels in version 1.0 of the evaluation framework

As explained in Section 3.5.2, the results of and experience gained from the feature analysis motivated two refinements. The first focused on condensing some of the features in Feature Set 2 (ease of introduction and setup). As part of version 1.0 of the evaluation framework, features were generated to assess whether a tool had reasonable system requirements, a simple installation and setup procedure, an installation guide and a tutorial (see Table 7-1). However, following the feature analysis (reported in Chapter Three), separate features for assessing a tool's system requirements,

installation guide and tutorial, were removed. This change was made because; in practice, it became clear that assessing these characteristics (i.e. system requirements, installation guide, and tutorial) formed part of a much larger assessment of the simple installation and setup feature. Therefore, consideration of a tool's system requirements, installation guide and/or tutorial, are instead included as suggested criteria for evaluating the overall installation and setup.

id	Feature Set	id	Feature	Level of importance
F1	Economic	F1-F01	No financial payment	Highly desirable
		F1-F02	Maintenance	Highly desirable
F2	Ease of introduction and setup	F2-F01	The tool has reasonable system requirements.	Mandatory
		F2-F02	Simple installation and setup.	Highly desirable
		F2-F03	There is an installation guide.	Highly desirable
		F2-F04	There is a tutorial.	Highly desirable
		F2-F05	The tool is self-contained.	Highly desirable
F3	Systematic review activity support	F3-F01	Protocol development	Desirable
		F3-F02	Protocol validation	Desirable
		F3-F03	Supports automated searches	Highly desirable
		F3-F04	Study selection and validation	Highly desirable
		F3-F05	Quality assessment and validation	Highly desirable
		F3-F06	Data extraction and validation	Highly desirable
		F3-F07	Data synthesis	Highly desirable
		F3-F08	Text analysis	Nice-to-have
		F3-F09	Meta-analysis	Nice-to-have
		F3-F10	Report write up	Nice-to-have
		F3-F11	Report validation	Nice-to-have
F4	Process Management	F4-F01	Support for multiple users	Mandatory
		F4-F02	Document management	Mandatory
		F4-F03	Security	Desirable
		F4-F04	Management of roles	Highly desirable
		F4-F05	Support for multiple projects	Mandatory

Table 7-1 Features and importance levels from version 1.0 of the evaluation framework

Another change was made to a process management (Feature Set 4) feature. The assessment of a tool's capability for undertaking multiple projects was strengthened. Initially, the ability for users to perform multiple systematic reviews using a tool was targeted for evaluation. However; on

reflection, this feature alone was not considered a substantial or meaningful addition to the framework. Therefore, the scope of support for undertaking multiple projects was expanded to reflect the value of multi-project support. The framework now aims to assess a tool's support for re-using past systematic review data. It was determined that such support may be useful when:

- A new systematic review is being undertaken in a topic where a relevant systematic review already exists.
- Updating a previously completed systematic review.

However, it is noted that the quality of support for reusing past systematic review data will rely on a tool's capacity for handling multiple projects. Therefore, effective support for multiple projects is still suggested as an important criteria to consider when evaluating this feature within a tool. As reported in Section 3.5.2, at the time that this change was made, a level of importance for reusing past systematic review data, was not yet able to be determined (see Table 7-2). This was because the factors which influenced the development of features for version 1.0 of the framework were deemed insufficient to inform a suitable weighting. As shown in Figure 7-2, developing the features in version 1.0 relied heavily on the characteristics and experiences of performing systematic reviews in software engineering. Experience of reusing past systematic review data, in this domain, is limited. However, as discussed in the next section (Section 7.3.1.2), further research undertaken to explore systematic reviews and support tools in other domains, was used to help inform a suitable level of importance.

Based on these refinements, a new version of the framework (version 1.1) was established. Changes made to version 1.1 are discussed in the next section (Section 7.3.1.2).

7.3.1.2 Changes made to version 1.1 of the evaluation framework

As reported in Chapters Five and Six, a cross-domain, interview-based survey was designed and undertaken to explore the experiences and opinions of systematic reviewers in other domains (outside of software engineering) about support tools.

As reported in Section 5.1.1, the aims of the study were:

1. To explore what tools were currently available and used to support systematic reviews in other domains; specifically, healthcare and areas of social science.
2. To identify what participants considered to be the most important characteristics (or features) of tools to support systematic reviews.
3. To compare the features and importance levels identified in the survey with those forming part of version 1.1 of the evaluation framework for tools supporting the whole systematic review process in software engineering.

To achieve the second and third aim, participants were presented with features from version 1.1 of the evaluation framework (see Table 7-2) and asked to rate them using the same levels of importance included as part of the framework's scoring process (i.e. mandatory, highly desirable, desirable, nice-to-have or not necessary). The features and importance levels identified by participants were compared with ratings from a software engineering perspective (i.e. the importance levels allocated to features in version 1.1 of the evaluation framework). The results from this part of the survey motivated two changes to the framework.

As reported in Section 6.1.5.8, results suggested that the majority of participants did not consider support for text analysis as particularly important (see Table 7-3). Furthermore, as mentioned in Section 6.2.4.1, text analysis was identified as one of the features showing little consensus amongst participants. These issues prompted reconsideration of how text analysis was presented within the framework. Text analysis techniques underpin many of the current approaches being investigated, developed and used to support systematic reviews. The literature review reported in Chapter Two, for example, identified text mining (a form of text analysis) as the most common underlying approach of (at the time) currently available tools to support systematic reviews in software engineering (see Section 2.3.3). Such tools aimed to support various stages of a systematic review; including, study selection, data extraction and data synthesis (see Section 2.3.4).

id	Feature Set	id	Feature	Level of importance
F1	Economic	F1-F01	No financial payment	Highly desirable
		F1-F02	Maintenance	Highly desirable
F2	Ease of introduction and setup	F2-F01	Simple installation and setup.	Highly desirable
		F2-F02	The tool is self-contained.	Highly desirable
F3	Systematic review activity support	F3-F01	Protocol development	Desirable
		F3-F02	Protocol validation	Desirable
		F3-F03	Supports automated searches	Highly desirable
		F3-F04	Study selection and validation	Highly desirable
		F3-F05	Quality assessment and validation	Highly desirable
		F3-F06	Data extraction and validation	Highly desirable
		F3-F07	Data synthesis	Highly desirable
		F3-F08	Text analysis	Nice-to-have
		F3-F09	Meta-analysis	Nice-to-have
		F3-F10	Report write up	Nice-to-have
		F3-F11	Report validation	Nice-to-have
F4	Process Management	F4-F01	Support for multiple users	Mandatory
		F4-F02	Document management	Mandatory
		F4-F03	Security	Desirable
		F4-F04	Management of roles	Highly desirable
		F4-F05	Support for reuse of past systematic review data	*Not yet determined*

Table 7-2 Features and importance levels from version 1.1 of the evaluation framework

In areas of healthcare and social science, work has also been undertaken to investigate the application of text analysis within systematic reviews (Thomas *et al.*, 2010; Cohen *et al.*, 2010). Similarly, research in these domains suggests text analysis can support searching for papers (O'Mara-Eves *et al.*, 2015), study selection (Shemilt *et al.*, 2013), data extraction (Siddhartha *et al.*, 2015) and data analysis (Thomas *et al.*, 2010). As shown in Table 7-4, these stages of a systematic

F3-F08 - Text analysis					
Mandatory	**Highly Desirable**	**Desirable**	**Nice-to-have**	**Not Necessary**	**Framework (V1.1) Ratings**
0	3	2	**5**	3	Nice-to-have

Table 7-3. Summary of participant ratings for text analysis (F3-F08)

review were considered by participants as high priorities for tool support. In particular, features supporting data extraction, data analysis, study selection and the search process, all received ratings of highly desirable (or mandatory) by the majority of participants. In version 1.1 of the evaluation framework, text analysis was presented to participants as a single feature (see Table 7-2). It was determined, however, that this presentation did not adequately reflect how text analysis can support multiple stages of a systematic review. This limitation of the framework may suggest why the importance of text analysis was rated low and inconsistently by participants. Therefore, the framework has been updated. As reported in Section 6.2.4.2, text analysis is no longer presented as a separate feature in the framework. Instead, text analysis is considered an important characteristic of support, which is provided by other features. In particular, text analysis is now suggested as part of the assessment criteria when evaluating a tool's search, study selection, data extraction and data analysis features.

id	Feature	Mandatory	Highly Desirable	Desirable	Nice-to-have	Not Necessary	Framework (V1.1) Ratings
F3-F06	Data extraction	**7**	5	1	0	0	Highly Desirable
F3-F05	Quality assessment and validation	5	**7**	1	0	0	Highly Desirable
F3-F07	Data synthesis	5	**7**	1	0	0	Highly Desirable
F3-F04	Study selection and validation	5	**6**	2	0	0	Highly Desirable
F3-F09	Meta-analysis	4	**5**	2	2	0	Nice-to-have
F3-F03	Supports automated searches	3	**4**	3	3	0	Highly Desirable
F3-F01	Development of review protocol	2	**4**	2	3	2	Desirable
F3-F02	Protocol validation	1	1	**5**	1	**5**	Desirable
F3-F11	Report validation	0	3	3	3	**4**	Nice-to-have
F3-F08	Text analysis	0	3	2	**5**	3	Nice-to-have
F3-F10	Report write-up	0	2	**6**	4	0	Nice-to-have

Table 7-4. Summary of participant ratings for systematic review activity support (F3) features

Furthermore, as discussed in the previous section, support for reusing past systematic review data was introduced as a new feature to version 1.1 of the evaluation framework (see Table 7-2). However, it was not yet provided with a level of importance. As mentioned in the previous section (Section 7.3.1.1), this was because limited evidence about the experience of reusing data from past systematic reviews in software engineering, was available. Therefore, the full importance of a tool's support for this action was too difficult to determine based solely on a software engineering perspective. In other disciplines, systematic reviews commonly reuse and build upon findings identified by previous reviews (Chaudhry *et al.,* 2006; Deb *et al.,* 2008; DePanfilis & Zlotnik, 2008). In particular, systematic reviews are often revisited, maintained and updated over time, either by the original authors, or by other researchers in the field (Pai *et al.,* 2008; Harris *et al.*, 2012; Jones *et al.,* 2014). In healthcare, updating a systematic review is defined as a process to identify (and analyse) new evidence to incorporate into a previously completed systematic review (Moher & Tsertsvadze, 2006). For example, a Cochrane Review (i.e. a specific type of systematic review in healthcare), is recommended to be updated by the original authors within two years of the published version, or the previous update[10]. This is because certain healthcare interventions, which are known to be effective based on the findings of an existing Cochrane Review, may become ineffective in the future (Moher & Tsertsvadze, 2006). However, maintaining systematic reviews is described as a difficult process (Moher *et al.,* 2008). In particular, updates of previous reviews have been suggested as being just as costly and time consuming as conducting the original systematic review (Sutton *et al.,* 2009). To address these challenges; in healthcare, tools which aim to support this process have started to be developed. Wallace *et al.,* for example, has developed a tool, which aims to "reduce the burden of updating systematic reviews without sacrificing their comprehensiveness" (Wallace *et al.,* 2012). Its key feature involves reusing study selection data from a previous review to help distinguish relevant (from irrelevant) studies during an updated search. Early results have shown that the tool was able to reduce the workload by 70 – 90% (Wallace *et al.,* 2012).

[10] http://community.cochrane.org/editorial-and-publishing-policy-resource/cochrane-review-updates

Whilst maintaining and updating reviews is an intrinsic part of the systematic review process in; for example, healthcare, these activities are not wholly exclusive to the domain. Rather, they are considered, primarily, a consequence of the maturity of systematic reviews within the discipline. In software engineering, systematic reviews are still relatively recent and the majority (to-date) are still being undertaken to investigate new topics. However, as the field advances, the findings of previous reviews risk becoming out of date and losing their value. Therefore, it is anticipated that maintaining reviews will become an important activity in software engineering as well.

F4-F04 – Support for reuse of past systematic review data					
Mandatory	**Highly Desirable**	**Desirable**	**Nice-to-have**	**Not Necessary**	**Framework (V1.1) Ratings**
3	7	3	0	0	*Not yet determined*

Table 7-5. Summary of participant ratings for reuse of past systematic review data (F4-F04)

The majority of participants in the survey, who were all researchers in areas of healthcare and social science, identified tool support for reusing past systematic review data, as highly desirable (see Table 7-5). In particular, its potential for supporting updates of previously completed systematic reviews, as well as aspects of new reviews into topics where a similar systematic review already exists, were common themes (see Section 6.1.5.16). Therefore, in version 1.2 of the evaluation framework, a feature which supports reuse of past systematic review data has been classified as highly desirable.

These refinements led to a new version of the evaluation framework (version 1.2). Changes made to version 1.2 are discussed in the next section (Section 7.3.1.3).

7.3.1.3 Changes made to version 1.2 of the evaluation framework

Since beginning this project, research has been undertaken to investigate the systematic review process in software engineering (Kitchenham & Brereton, 2013). The objective was to identify, evaluate and analyse research published by software engineering researchers concerning their experiences of performing systematic reviews and their proposals for improving the systematic

review process. Findings suggested a number of changes to the guidelines for systematic reviews in software engineering and, as a consequence, the guidelines have since been revised (Kitchenham *et al.*, 2015). It was considered that some of the changes made to the guidelines might have implications for the evaluation framework. In response, the evaluation framework (version 1.2) was checked against the updated guidelines to make sure it was up-to-date.

The following three updates to the guidelines were considered to have implications for the evaluation framework:

Update 1. Recommending the use of a quasi-gold standard approach to integrate manual and automated searches.

The updated guidelines provide more information for validating the overall search strategy. In the previous version, the authors recommended using a known set of relevant papers to validate the search (Kitchenham & Charters, 2007). This particular approach was used to validate the search strategy used for the mapping study in this thesis (see Section 2.2.2.1). Determining a known set of papers for this method can be achieved using the experience and knowledge of researchers within the topic area of the systematic review, and/or by using studies identified in a previous review, which addressed a similar or overlapping topic (Kitchenham *et al.*, 2015). However, where the experience and knowledge of the researchers is limited or a similar review in the area does not exist, the updated guidelines now suggest an alternative way to validate the search strategy using a quasi-gold standard. A quasi-gold standard is generated by carrying out a limited manual search of relevant journals and conference proceedings (Zhang *et al.*, 2011). The papers found can then be used as the known set to validate the automated search. As highlighted in Figure 7-3, this process potentially establishes new areas for tool support, which were not previously considered by version 1.2 of the evaluation framework. Specifically, tools may be able to provide support for developing search strings and evaluating the search performance.

Update 2. Removing the recommendation for constructing structured research questions and using them to construct search stings.

Previously, researchers were advised to use structured research questions to construct their search terms (Kitchenham & Charters, 2007). However, issues with inconsistent terminology and poor support from digital libraries made developing search terms difficult (Kitchenham *et al.,* 2015). In response, the updated guidelines now recommend developing simple search strings based on a systematic review's main topic of interest. Furthermore, the authors recommend using tools to help identify key search terms and build search strings (Kitchenham *et al.*, 2015). In version 1.2 of the evaluation framework, features which support search string development were not included.

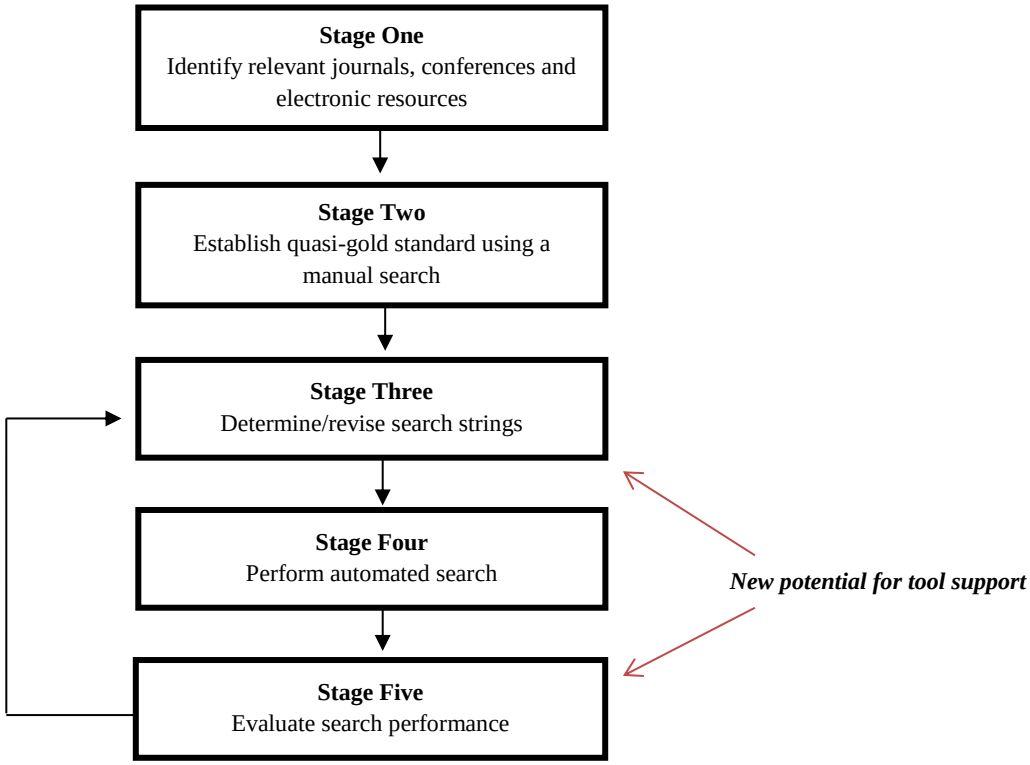


Figure 7-3. A process of validating the search using a quasi-gold standard

Update 3. Provide more guidance on using citation-based search strategies

The previous version of the guidelines focused, primarily, on providing details about how to perform an automated search (i.e. searching electronic resources using a defined set of search strings) and a manual search (i.e. manually searching through conference and journal proceedings). However, the updated guidelines now provide additional information about how to perform a citation-based search (also known as a snowballing search strategy). Snowballing involves

checking papers that cite other papers included in a systematic review (i.e. forwards snowballing) or checking papers that are cited in papers included in a systematic review (i.e. backwards snowballing). Although snowballing is referenced in the previous guidelines, it was considered to be only a secondary method which can be adopted to support an overall automated search. In the new guidelines, however, snowballing is also presented as a stand-alone search strategy. This update has been motivated by several studies, which have investigated the effectiveness of snowballing compared with a standard automated approach (Skoglund & Runeson, 2009; Jalali & Wohlin, 2012). Guidelines specifically for using snowballing in systematic reviews have also been published (Wohlin, 2014) Furthermore, as identified in the supplementary literature update, snowballing tools have started to surface (Bezerra *et al.*, 2014). As of version 1.2 of the evaluation framework, support for snowballing was not considered in the feature set (see Table 7-6).

id	Feature Set	id	Feature	Level of importance
F1	Economic	F1-F01	No financial payment	Highly desirable
		F1-F02	Maintenance	Highly desirable
F2	Ease of introduction and setup	F2-F01	Simple installation and setup.	Highly desirable
		F2-F02	The tool is self-contained.	Highly desirable
F3	Systematic review activity support	F3-F01	Protocol development	Desirable
		F3-F02	Protocol validation	Desirable
		F3-F03	Supports automated searches	Highly desirable
		F3-F04	Study selection and validation	Highly desirable
		F3-F05	Quality assessment and validation	Highly desirable
		F3-F06	Data extraction and validation	Highly desirable
		F3-F07	Data synthesis	Highly desirable
		F3-F08	Meta-analysis	Nice-to-have
		F3-F09	Report write up	Nice-to-have
		F3-F10	Report validation	Nice-to-have
F4	Process Management	F4-F01	Support for multiple users	Mandatory
		F4-F02	Document management	Mandatory
		F4-F03	Security	Desirable
		F4-F04	Management of roles	Highly desirable
		F4-F05	Support for reuse of past systematic review data	Highly desirable

Table 7-6 Features and importance levels from version 1.2 of the evaluation framework

Based on these updates to the guidelines, the scope of assessment within the framework for evaluating support for the search process has been widened. The following adjustments have been made to F3-F03 shown in Table 7-7:

- Support for developing search strings has been added as part of the suggested assessment criteria for evaluating a tool's support for the search process. This change was influenced by **Update 1** and **Update 2**.
- Support for validating/evaluating the search strategy has been added as part of the suggested assessment criteria for evaluating a tool's support for the search process. This change was influenced by **Update 2**.
- Support for backwards and forwards snowballing has been added as part of the suggested assessment criteria for evaluating a tool's support for the search process. This change was influenced by **Update 3**.
- The label for this feature (*"Supports automated searches"*) has been renamed *"Supporting the search"* to adequately reflect the expanded scope of assessment.

7.3.2 Presenting, applying and validating version 1.3 of the evaluation framework

Over the course of this research project, new tools developed to support systematic reviews in software engineering, have emerged. The literature review reported in Chapter Two aimed to identify and classify currently available tools to support systematic reviews in software engineering. As reported in Section 2.4, a supplementary search of the literature was performed to update its findings. Five additional papers reporting new tools or approaches to support systematic reviews in software engineering were found (see Table 2-9). Most of the papers present a novel visualisation or text mining approach developed to support a specific stage of a systematic review. One of the papers presents a new tool (*SESRA*) to support the whole process (Molléri & Benitti, 2015). Similar tools identified by the literature review were compared and evaluated as part of the feature analysis reported in Chapter Three.

As discussed in Section 7.3.1.1, version 1.0 of the evaluation framework was developed in order to perform the feature analysis. The framework was later refined and updated to version 1.1 based on its results. Findings from a cross-domain survey (reported in Chapters Five and Six) led to additional changes, resulting in version 1.2 of the framework (see Section 7.3.1.2). In Section 7.3.1.3, the evaluation framework was checked against the 2015 guidelines for performing systematic reviews in software engineering. This exercise motivated further changes and the framework was updated to version 1.3.

In this section, version 1.3 of the evaluation framework is presented and validated as part of a new comparative evaluation. The tool identified by the supplementary literature update (*SESRA*) is compared and evaluated against the strongest candidate (*SLuRp*) from the feature analysis.

7.3.2.1 Candidate tools

The two candidate tools selected for evaluation are:

- ***SESRA*** **-** Described as a web-based automated tool to support all phases of the systematic review process in software engineering (Molléri & Benitti, 2015). *SESRA* was identified in the supplementary literature update (see Section 2.4). Results from some preliminary evaluation work (performed by its developers) suggest the tool can improve the reliability and productivity of a team-based systematic review. However, as with the majority of tools found, *SESRA* has not yet been independently evaluated.
- ***SLuRp*** **(Systematic Literature unified Review Program)** - SLuRp was previously evaluated as part of the feature analysis reported in Chapter Three (see Section 3.3.1) and was determined as the strongest candidate. The tool, developed using Java and SQL, is described as an open source web-enabled database that supports the management of systematic reviews in software engineering (Bowes *et al.*, 2012). No major changes have been made to the tool since its previous evaluation.

SESRA will undergo a full evaluation using the evaluation framework. However, *SLuRp* will only be assessed against any features which have changed during the framework's development. This is because *SLuRp* was already evaluated using a previous version of the evaluation framework (version 1.0) as part of the feature analysis in Chapter Three. As no major changes have been made to *SLuRp* since the feature analysis, its scores for features not refined after version 1.0 of the evaluation framework remain unchanged.

7.3.2.2 Applying the framework

Each feature, its level of importance and any suggested assessment criteria, are presented in this section as a series of figures, organised by feature set:

- **Economic** features (feature set 1) are presented in Figure 7-4.
- **Ease of introduction and setup** features (feature set 2) are presented in Figure 7-5.
- For **systematic review activity support** features (feature set 3), features concerning the:
    - *planning phase* are presented in Figure 7-6,
    - *conduct phase* are presented in Figure 7-7,
    - *reporting phase* are presented in Figure 7-8.
- **Process management** features (i.e. feature set 4) are presented in Figure 7-9.

The same scoring process explained in Section 3.3.3 was used to evaluate the candidate tools here. To recap, this involved:

1. Scoring each tool against each feature to produce a raw score,
2. Using the level of importance weightings (i.e. multiplier) to covert raw scores to weighted scores for each feature.
3. Determining scores for each feature set and an overall score for each candidate tool.

The weightings (assigned to features and feature sets), judgement scale (and its interpretations) and equations used to generate scores are summarised in Figure 7-10.

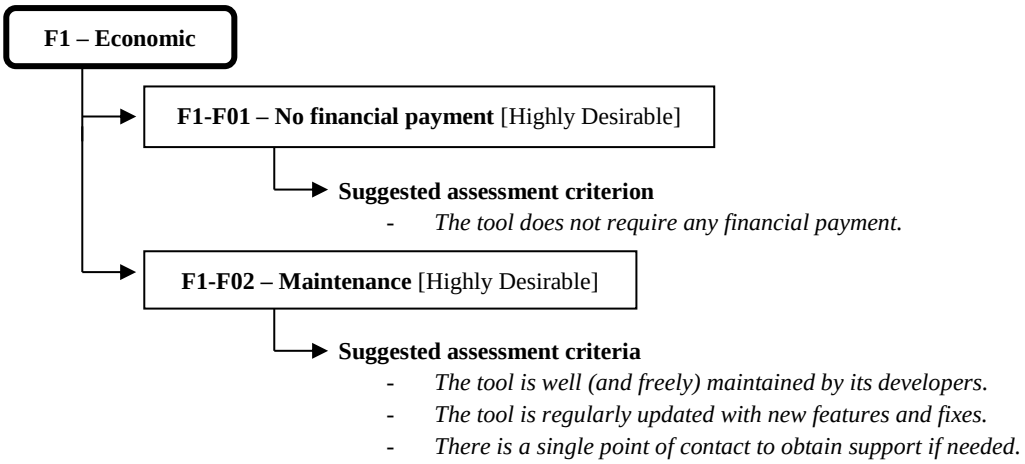


Figure 7-4. Feature set 1 (F1) Economic

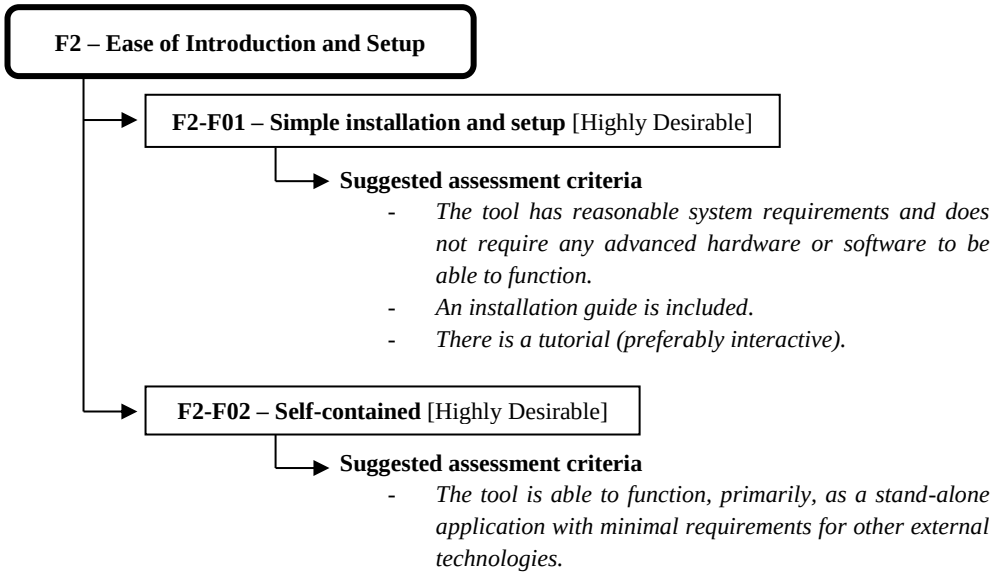


Figure 7-5. Feature set 2 (F2) Ease of introduction and setup

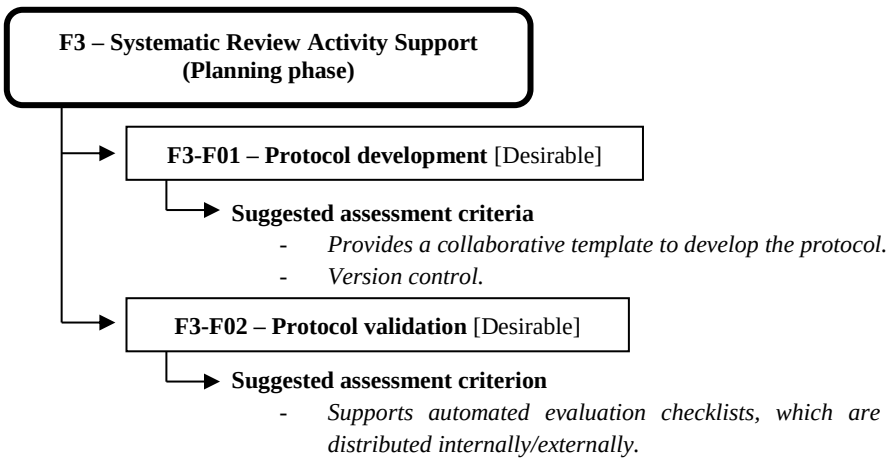


Figure 7-6. Feature set 3 (F3) Systematic review activity support (planning phase)

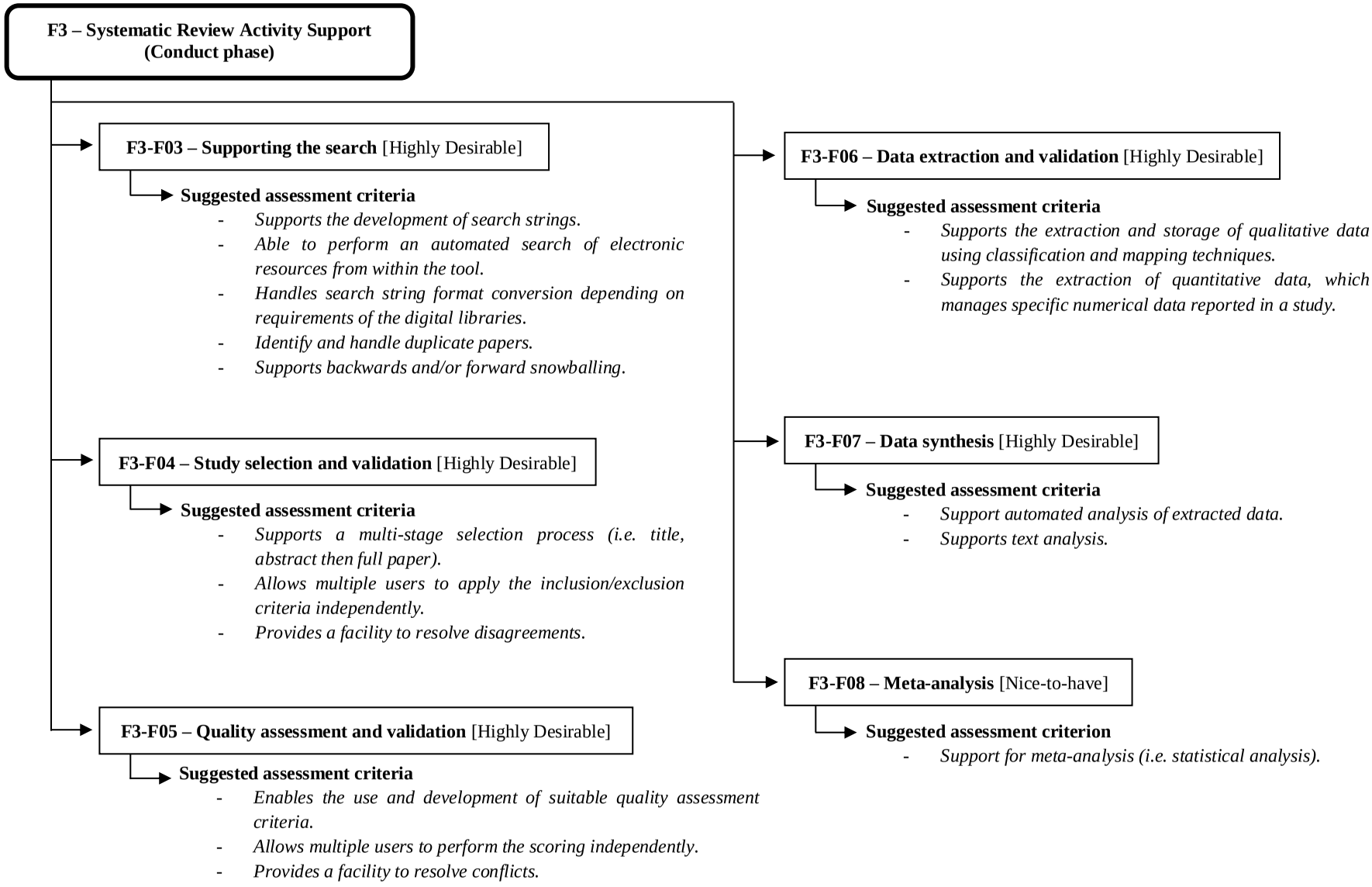


Figure 7-7. Feature set 3 (F3) Systematic review activity support (conduct phase)

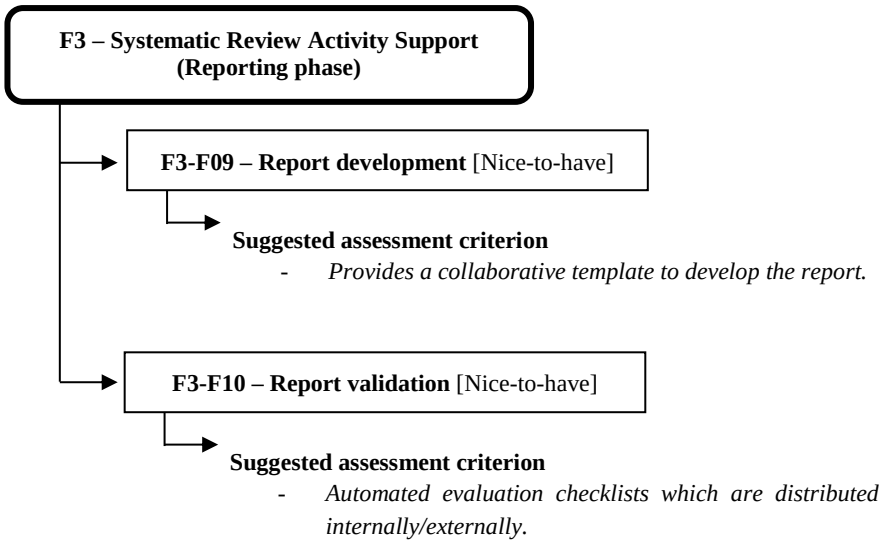


Figure 7-8. Feature set 3 (F3) Systematic review activity support (reporting phase)

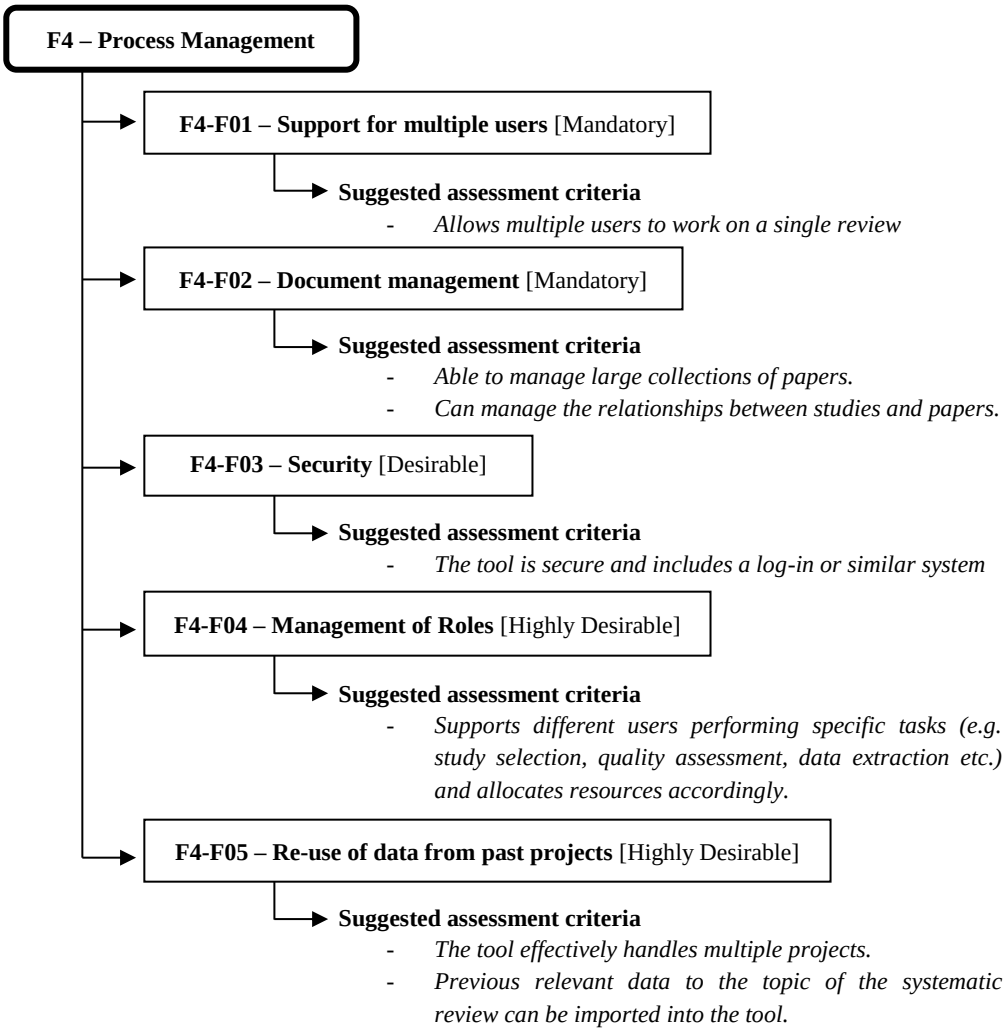


Figure 7-9. Feature set 4 (F4) Process management

Judgement scale and its interpretation

Is the feature present?	Score
Yes	1
Partly	0.5
No	0

JI1

Is the tool simple to install and setup?	Score
Yes	1
Some difficulties OR The tool could be installed, but there were a number of slight difficulties throughout the process.	0.5
No - The tool could be installed but the process was very difficult. OR No - The tool could not be installed.	0

JI2

Is the activity supported?	Score
Yes - Fully	1
Partly (Support is limited. Some aspects of the activity are not supported.)	0.5
No	0

JI3

Weightings

Importance	Multiplier
Mandatory (M)	*4
Highly Desirable (HD)	*3
Desirable (D)	*2
Nice to have (N)	*1

Level of importance of a feature

Feature Set	Weight
F1	0.1
F2	0.2
F3	0.4
F4	0.3

Feature set weighting

Equations

$$Percentage\ Score = \frac{Sum\ of\ Weighted\ Score}{Maximum\ Score} \times 100\% \quad (\mathbf{1})$$

$$Overall\ score = \frac{\sum_{i=1}^{4}(w_i TP_i)}{\sum_{i=1}^{4}(w_i)} \quad (\mathbf{2})$$

Calculating the scores

1. Weightings are assigned to each feature (i.e. levels of importance) and feature set.
2. A single judgement scale, which can be interpreted in one of three ways (JI1, JI2 and JI3), is used to score features. Using the suggested assessment criteria as a guide, where a feature is considered fully present it is awarded a raw score of 1, where it is partly present or partially supported it is awarded a raw score of 0.5 and where it is absent or minimally supported it is awarded a raw score of 0.
3. Weighted scores are calculated for each feature by multiplying the raw score by the importance weighting for that feature. These weighted scores are combined to determine percentage scores for each feature.
4. Using Equation 1, the maximum score for a feature set is assumed to be the sum of the weighted scores where all features in the set are fully present (or fully supported).
5. The overall score for each tool can be determined using Equation 2, where $\boldsymbol{w_i}$ is the weighting for the **i**$^{\mathbf{th}}$ feature and $\boldsymbol{TP_i}$ is the percentage score for the **i**$^{\mathbf{th}}$ feature set.

Figure 7-10. Score calculation

Table 7-7 displays all of the features (and feature sets) next to their assigned importance weighting and relevant interpretation of the judgement scale. As was the case with the feature analysis (see Section 3.3.3.3), this evaluation is still intended to assess support tools from the perspective of a collaborative systematic review undertaken in software engineering. Therefore, overall weightings remain high for feature sets 3 and 4 and are lower for feature sets 1 and 2. The total score possible for a tool is 50.

id	Feature Set	id	Feature	Feature Level of Importance	Interpretation of Judgement Scale	Feature set Importance Weighting
F1	Economic	F1-F01	No financial payment	HD	JI1	0.1
		F1-F02	Maintenance	HD	JI1	
F2	Ease of introduction and setup	F2-F01	Simple installation and setup	HD	JI2	0.2
		F2-F02	The tool is self-contained	HD	JI1	
F3	Systematic review activity support	F3-F01	Protocol development	D	JI3	0.4
		F3-F02	Protocol validation	D	JI3	
		F3-F03	Supporting the search	HD	JI3	
		F3-F04	Study selection and validation	HD	JI3	
		F3-F05	Quality assessment and validation	HD	JI3	
		F3-F06	Data extraction and validation	HD	JI3	
		F3-F07	Data synthesis	HD	JI3	
		F3-F08	Meta-analysis	N	JI1	
		F3-F09	Report write-up	N	JI3	
		F3-F10	Report validation	N	JI3	
F4	Process Management	F4-F01	Support for multiple users	M	JI1	0.3
		F4-F02	Document management	M	JI1	
		F4-F03	Security	D	JI1	
		F4-F04	Management of roles	HD	JI1	
		F4-F05	Reuse of data from past projects	HD	JI1	

Table 7-7 Features, assigned weightings and interpretation of judgement scale (version 1.3)

7.3.2.3 Results for SESRA

Table 7-8 presents the scores for *SESRA.*

Feature set 1

SESRA requires no financial payment to use and can be accessed from the development team's website[11]. To-date, the tool has not been particularly well maintained and whilst trying out the tool, a number of errors were encountered. The tool does, however, include a single point of contact to request assistance if necessary. Furthermore, new features and improvements to the tool are reportedly in development. *SESRA* scored 4.5 out of 6 for this feature set.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	4.5/6	75%
	F1-F02	HD	JI1	0.5	1.5		
F2	F2-F01	HD	JI1	0.5	1.5	4.5/6	75%
	F2-F02	HD	JI2	1	3		
F3	F3-F01	D	JI3	1	2	13.5/22	61%
	F3-F02	D	JI3	1	2		
	F3-F03	HD	JI3	0.5	1.5		
	F3-F04	HD	JI3	0.5	1.5		
	F3-F05	HD	JI3	0.5	1.5		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	0.5	1.5		
	F3-F08	N	JI1	0	0		
	F3-F09	N	JI3	1	1		
	F3-F10	N	JI3	1	1		
F4	F4-F01	M	JI1	1	4	7.5/16	47%
	F4-F02	M	JI1	0	0		
	F4-F03	D	JI1	1	2		
	F4-F04	HD	JI1	0.5	1.5		
	F4-F05	HD	JI1	0	0		
Total Score					**Overall % Score Using Feature Set Weightings**		
31/50					**61%**		

Table 7-8 Scores for *SESRA*

[11] http://sesra.net/

Feature set 2

Following a simple registration process, the tool is ready to use online at the development team's website. A supporting paper is available which presents the tool and explains aspects of its functionality. The tool's website hosts two video tutorials (currently in Portuguese only), which demonstrate how the tool supports aspects of the planning, conduct and report phases of a systematic review. A technical report is also available but, again, has not been translated to English. *SESRA* scored 4.5 out of 6 for this feature set.

Feature set 3

SESRA includes a template for multiple users to develop a review protocol. The research questions, study selection criteria, search strings, quality assessment, data extraction process and data synthesis approach, can all be defined. The template builds a Word document which can be exported from the tool. Protocol validation is also supported. The protocol can be distributed to other researchers (who are registered with *SESRA*) for feedback. Comments about the protocol can be made and shared using the tool.

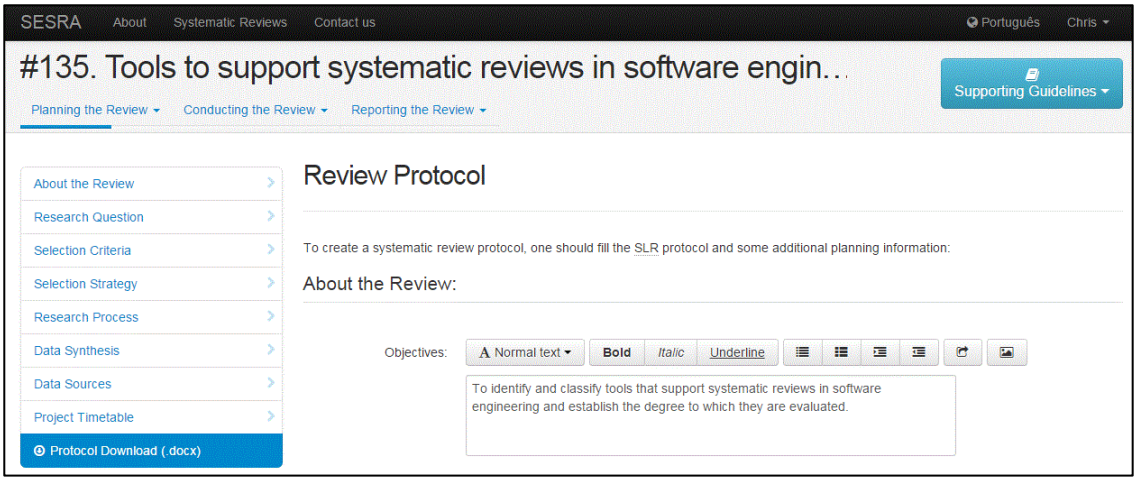


Figure 7-11. Screenshot of the tool's template for developing the protocol (*SESRA*).

SESRA provides partial support for the search. The tool offers some integration with IEEExplore and allows users to perform a preliminary search of the resource from within the tool. The main search, however, is not supported and must still be performed externally. *SESRA* supports

independent study selection amongst multiple researchers. However, this stage is considered only partially supported. The tool does not accommodate a multi-stage selection process effectively, nor did it allow studies to be included or excluded based on multiple criteria. *SESRA* flags any disagreements between user's selections and highlights an option to "request third party mediation" (see Figure 7-12). Selecting this option, however, failed to present an appropriate facility. *SESRA* also provides partial support for quality assessment. Users can opt to design and apply their own quality assessment criteria (see Figure 7-13) or choose from one of three predetermined instruments. The tool does not allow users to perform the scoring independently (i.e. scores are not blinded). Data extraction is also partially supported by *SESRA*. Users can design simple classification forms during the development of the protocol, which are used to extract and store quantitative and qualitative data from studies. Forms feel restrictive, however, and are difficult to modify. Analysis options are generally limited and meta-analysis is not supported. In particular, extracted data can (to-date) only be summarised and presented as a single table.

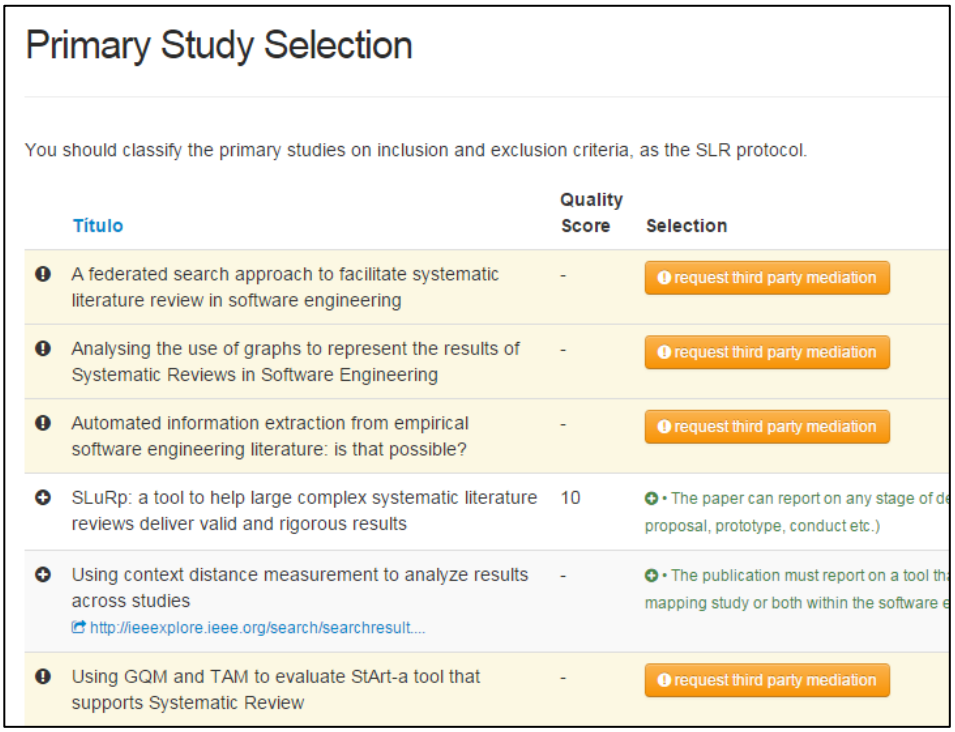


Figure 7-12. Screenshot of the tool's facility for resolving a conducting quality assessment score, inclusion or exclusion (*SESRA*)

SESRA provides support for writing and validating the final report. The main body of the report can be written within the tool and exported as a Word document. *SESRA* supports report validation in the same way it supports validating the protocol. *SESRA* scored 13.5 out of 22 for this feature set.

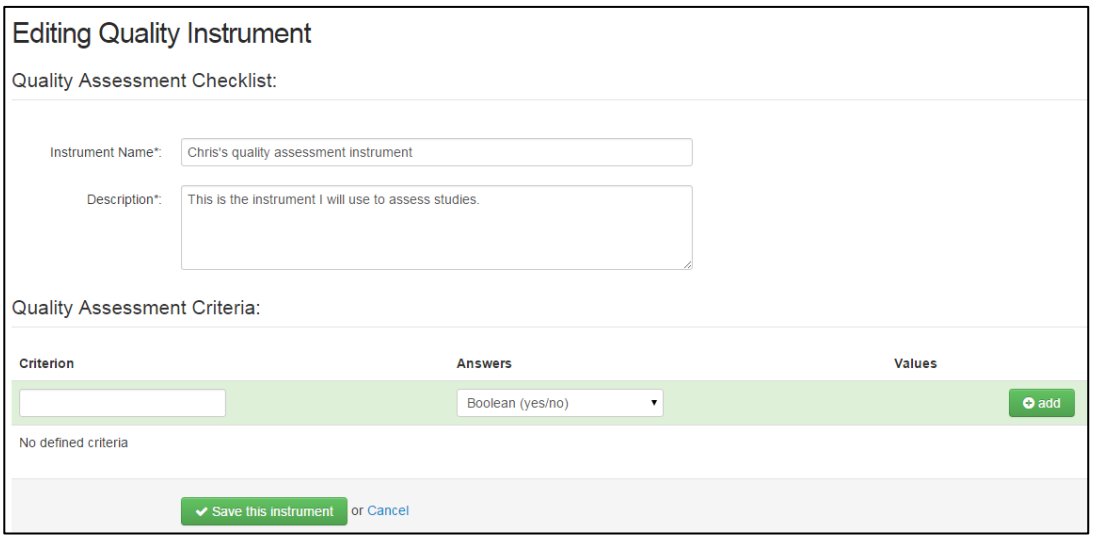


Figure 7-13. Screenshot of the tool's facility to create quality assessment criteria (*SESRA*)

Feature set 4

SESRA allows multiple users to work on a single review. Registered users can be added and removed from projects quickly and easily. Users can be assigned one of three roles; namely, "Team Researcher", "Mediator/Advisor" or "Other Stakeholder" (see Figure 7-14). The definition of these roles, however, is vague and their usefulness in the context of the tool is not clearly defined. Support for document management is also limited. Papers cannot be imported into the tool in bulk and must be manually imported one at a time. Once papers are stored, the tool provides minimal facilities to manage and organise them. Whilst the tool effectively handles multiple projects, reuse of past systematic review data is not supported. *SESRA* scored 7.5 out of 16 for this feature set.

Overall score

Using the calculation process shown in Figure 7-10 (particularly Equation 2) the overall score for *SESRA* is **61%.**

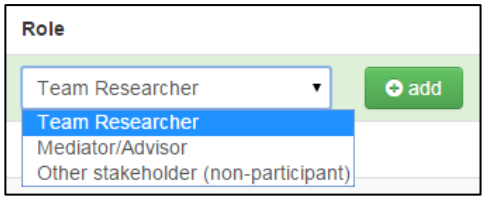


Figure 7-14. Screenshot showing the different roles that can be assigned to users (*SESRA*)

7.3.2.4 Updated results for SLuRp

As mentioned in Section 7.3.3.1, *SLuRp* has only been reassessed against features in version 1.3 of the evaluation framework which have changed since version 1.0. Its scores for features which have not changed remain the same as the tool's results in the previous feature analysis.

Table 7-9 presents the scores for *SLuRp*. Two scores (highlighted in bold) were updated.

Feature Set	Feature	Importance	Judgement Scale	Raw Score	Weighted Score	Feature Set Score	% Feature Set Score
F1	F1-F01	HD	JI1	1	3	6/6	100%
	F1-F02	HD	JI1	1	3		
F2	F2-F01	HD	JI1	**0.5**	**1.5**	1.5/6	27%
	F2-F02	HD	JI2	0	0		
F3	F3-F01	D	JI3	0	0	9/22	41%
	F3-F02	D	JI3	0	0		
	F3-F03	HD	JI3	0	0		
	F3-F04	HD	JI3	0.5	1.5		
	F3-F05	HD	JI3	1	3		
	F3-F06	HD	JI3	0.5	1.5		
	F3-F07	HD	JI3	0.5	1.5		
	F3-F08	N	JI1	0.5	0.5		
	F3-F09	N	JI1	1	1		
	F3-F10	N	JI3	0	0		
F4	F4-F01	M	JI1	1	4	16/16	100%
	F4-F02	M	JI1	1	4		
	F4-F03	D	JI1	1	2		
	F4-F04	HD	JI1	1	3		
	F4-F05	HD	JI1	**1**	**3**		
Total Score					**Overall % Score Using Feature Set Weightings**		
34/50					**61.8%**		

Table 7-9 Updated scores for *SLuRp*

After consolidating features to better assess a tool's overall installation and setup (see Section 7.3.1.1), *SLuRp*'s score improved for F2-F01. In its previous evaluation, *SLuRp* scored poorly for this feature. However, the tool did receive partial marks for 'reasonable system requirements' and 'installation guide' features (see Section 3.4.1.2). Since these factors are now included as suggested criteria to assess a tool's setup (see Figure 7-5), an improved score for F2-F01 was considered justified. Following changes made to version 1.0 of the evaluation framework, 'support for multiple projects' was strengthened to 'support for reusing data form past systematic reviews' (see Section 7.3.1.1). This feature was reassigned a highly desirable level of importance (see Section 7.3.1.2). *SLuRp* was previously only scored on its ability to handle multiple projects. However, the tool is also considered to provide useful support for reusing data from past systematic review projects. Papers, studies and extracted data can all be shared easily between multiple systematic reviews, managed inside the tool.

Overall score

Using the calculation process shown in Figure 7-10 (particularly Equation 2) the overall score for *SLuRp* is updated to **61.8%.**

7.3.2.5 Discussion of results

As shown in Table 7-10, *SESRA* achieved an overall score of **61%**. The tool's main strengths are its support for protocol development (and validation) and writing and validating the report. Its main weaknesses are inadequate document management facilities and a lack of support for reusing past systematic review data. It is noted that support for document management, in the context of the evaluation framework, is considered mandatory. Therefore, since *SESRA*, was not considered to provide full (or partial) support for this feature, the tool in its current state cannot be recommended. *SLuRp* achieved a very similar (updated) overall score of **61.8%**. The key strengths of the tool remain its support for quality assessment, collaboration and managing the overall process. Its

weaknesses still include a complex setup and lack of support for the search and protocol stages of a systematic review.

	F1 (scores out of 6)		F2 (scores out of 6)		F3 (scores out of 23)		F4 (scores out of 16)		Total (score out of 50)		Overall Score
SLuRp	6	100%	1.5	27%	9	41%	16	100%	34	68%	**61.8%**
SESRA	4.5	75%	4.5	75%	13.5	61%	7.5	47%	31	62%	**61%**

Table 7-10 Feature set scores and overall scores for *SESRA* and *SLuRp*

Within the constraints of the evaluation framework, *SLuRp* is still considered (to-date) the most suitable tool to support systematic reviews in software engineering. Whilst the development of *SESRA* suggests improvement for some areas of systematic review tool support (such as support for the planning and report phases), the tool's poor support for managing large numbers of papers and studies, which is a key aspect of undertaking a systematic review, is a significant drawback.

7.3.3 Summarised response to RQ2

What are the most important features of a systematic review tool in software engineering?

As part of this research project, features for an overall tool to support systematic reviews in software engineering have been established. 19 individual features (each with suggested criteria for their assessment) have been defined and organised into one of four feature sets (see Figure 7-6 to 7-9). The features (and suggested assessment criteria) are included as part of a flexible evaluation framework created to support the evaluation, selection and future development of tools. Based on the work undertaken to develop the framework, support for document management and collaboration (amongst multiple reviewers) are considered to be the most important features of a tool to support systematic reviews in software engineering. As shown in Table 7-7, tool support for multiple users (F4-F01) is classified as a mandatory feature. The importance of collaboration amongst multiple reviewers is emphasised throughout version 1.3 of the evaluation framework. In

particular, a positive assessment of a tool's support for many stages involved in a systematic review (e.g. protocol development, study selection, quality assessment data extraction etc.) is largely influenced by how well collaboration is supported. Similarly, tool support for document management (F4-F02) is also considered mandatory (see Table 7-7). Being able to effectively manage a large number of papers and studies (and handle the relationships between them) is considered an important factor toward achieving a successful, large-scale systematic review. Developers are recommended to prioritise these features when developing tools to support systematic reviews in software engineering. More recommendations to assist tool developers and potential tool users are provided in the following chapter (Chapter Eight).

7.4 Summary

In this chapter the findings from all of the research undertaken have been brought together. In particular, the results of the literature review, feature analysis and survey have been discussed in relation to the original aim and research questions. The development of an evaluation framework for an overall tool to support systematic reviews in software engineering has been presented and discussed. Changes made to the evaluation framework at various points in the project have been described and explained. Details of a validation exercise, where the framework was checked against the updated guidelines for performing systematic reviews in software engineering, has also been given. The most recent version of the evaluation framework (version 1.3) has been presented and further validated as part of another comparative evaluation. In the next chapter, a summary of the work and conclusions are provided. In addition, recommendations about the use and development of tools to support systematic reviews, along with suggestions for future work, are outlined.

Chapter Eight

Summary and Conclusions

This chapter provides a summary of the research undertaken and conclusions. Some final thoughts on the evaluation framework are presented and discussed. Recommendations for tool users and developers are also provided, along with suggestions for future work.

8.1 Summary and Conclusions of the Work Undertaken

The overall aim of this thesis was to investigate the usefulness and development of tools to support systematic reviews in software engineering. As part of this investigation, an evaluation framework for an overall tool would be developed. In this section, a summary of the work undertaken is provided along with conclusions.

In the early stages of this project, a literature review (taking the form of a mapping study) was undertaken to identify and classify special-purpose tools that support systematic reviews in software engineering and establish the degree to which they had been evaluated. The results of the literature review showed a small but encouraging growth of tools to support systematic reviews in software engineering. However, the majority of tools found were in the early stages of development (and usage) and had received limited evaluation. This meant there was very little primary data about the effectiveness of tools and, generally, only speculation over their potential. The findings of the literature review provided motivation for an independent evaluation of tools.

Four tools aiming to support the whole systematic review process in software engineering were independently evaluated as part of a feature analysis. To carry out this exercise, an initial version of the evaluation framework was developed. The components of version 1.0 of the evaluation framework were determined based on reported experiences of performing systematic reviews in software engineering, the results of the literature review, generic factors about tool evaluation in the literature and discussion between members of the evaluation team. The feature analysis focused on assessing how well tools provided support for each stage of a systematic review and managing the overall process. The results of the study identified strengths and weaknesses for each tool and identified the strongest (and weakest) candidate. Two refinements to the evaluation framework were made based on the results of and experience gained from undertaking the feature analysis. The framework was updated to version 1.1.

Following the feature analysis, two possible routes for future work were identified. The first was to circulate the framework within the evidence-based software engineering community for feedback

and refinement. However, this direction was decided against in favour of exploring tool support for systematic reviews in other domains, outside of software engineering. The latter would provide a deeper understanding of the technology being evaluated.

A survey was designed and implemented to interview 13 researchers across healthcare and social science about tools to support systematic reviews. A variety of tools were identified by participants and classified into one of seven categories. Reference management tools were mentioned most often by participants. Special-purpose tools including *RevMan* and *EPPI-Reviewer*, which aim to support the whole systematic review process, were the second most common. The strengths and weaknesses of these tools were identified and analysed. Participants were also presented with the feature set from version 1.1 of the evaluation framework. They were asked to rate the importance of features using the same weightings included as part of the framework's scoring process. Support for multiple users, data extraction and maintenance were the top three most important features classified by participants. Support for text analysis and the report phase of a systematic review were considered the least important. On comparing the importance levels of features identified by participants with ratings from a software engineering perspective; generally, there was a good level agreement. There were notable differences concerning support for meta-analysis, role management and security. A possible explanation for these differences is that the importance of these features is context-dependent. The survey reaffirmed that many problems relating to systematic reviews faced in other domains are similar to those faced by researchers in software engineering and that improved tools are needed. Two more modifications were made to the evaluation framework based on the results of the survey. The framework was updated to version 1.2.

During this project, a novel resource was developed to allow researchers to identify tools to support systematic reviews, based on their needs. *Systematic Review (SR) Toolbox* is a web-based, community driven catalogue of tools that support systematic reviews across multiple domains. This resource was developed in response to limited up-to-date information about what tools were currently available to support systematic reviews. The design of *SR Toolbox* is largely influenced by the work reported in this thesis. In particular, the classification criteria used to organise and

search for tools was mapped from categories used in the literature review and features from earlier versions of the evaluation framework. *SR Toolbox* has been well received by the research community and is actively used by research staff and students across multiple domains.

The guidelines for performing systematic reviews in software engineering have been updated. It was considered that some of the changes made to the guidelines may have implications for the evaluation framework. Therefore, version 1.2 of the evaluation framework was checked against the updated guidelines to ensure its suitability and relevance. Based on this exercise, four minor changes were made to some of the features in the framework. The framework was updated to version 1.3. As a final validation activity, version 1.3 of the evaluation framework was used in a second comparative evaluation. Using the framework, the strongest candidate from the feature analysis (*SLuRp*) was compared and evaluated against a new tool (*SESRA*) identified in the supplementary literature update. The overall score for both tools were similar. However, *SLuRp* was still considered the most suitable tool as it supported a higher number of mandatory features.

This research project has provided valuable insight into the potential and usefulness of tools to support systematic reviews. The current state of tool support, however, is still in its infancy. Empirical evidence about the effectiveness of tools is still limited and, whilst many show promise, the majority are still in need of further independent evaluation before they can be recommended. The evaluation framework presented in this thesis aims to support the maturation of systematic review tools in software engineering. In particular, the framework aims to support user assessment and selection of tools and act as a template for future tool development.

Some final thoughts about the evaluation framework, including some potential future refinements, are provided in Section 8.2. Recommendations to support the future use and development of systematic review tools in software engineering, along with suggestions for future work, are given in Section 8.3.

8.2 Final Thoughts on the Evaluation Framework

In this section, some final thoughts about the evaluation framework and potential future refinements are presented and discussed.

The evaluation framework has been designed to be flexible and expandable. In particular, the evaluator is able (and encouraged) to adapt the components such as the features (and suggested assessment criteria), importance weightings and judgement scale depending on the context. This level of flexibility is considered necessary, since the framework is being used to evaluate tools which are still in their infancy (Babar *et al*., 2004). It is anticipated, therefore, that as tools continue to evolve, so too should the evaluation framework. For example, version 1.3 of the evaluation framework currently emphasises tool support for systematic review activities and managing the overall process (see Figure 7-10). However, as tools mature, other features might be included and weightings adjusted which prioritise support for different aspects (e.g. usability and/or accessibility). Similarly, the further advancement of the systematic review methodology in software engineering may also have future implications for the framework. As discussed in Section 7.3.1.3, updates to the 2015 guidelines for performing systematic reviews have already resulted in minor changes. It is anticipated that as systematic reviews continue to mature within the discipline, new issues surrounding their undertaking and, therefore, new requirements for tool features, will begin to surface.

It may also be useful to introduce a 'reference tool' as part of a future version of the framework's scoring process. Elsewhere, Collier *et al* (1999) developed an evaluation framework to assist with the assessment and selection of data mining software. As a key part of its scoring process, a candidate tool is scored against what is generally agreed upon as the most suitable data mining tool to-date (i.e. the 'reference tool'). Using the scale shown in Table 8-1, a score is calculated for each tool and then summated to produce a total score for each set of features. Due to the novelty of tools to support systematic reviews, a 'reference tool' was not able to be determined for previous versions of the evaluation framework. However, the results of the feature analysis and subsequent

comparative evaluation (presented in Section 7.3.2) suggest that *SLuRp* is currently the most suitable tool to support systematic reviews in software engineering. Therefore, *SLuRp* may be considered an appropriate reference tool for future evaluations.

Relative performance	Rating
Much worse than the reference tool	1
Worse than the reference tool	2
Same as the reference tool	3
Better than the reference tool	4
Much better than the reference tool	5

Table 8-1 Scoring a candidate tool against the 'reference tool' (Collier *et al.*, 1999)

Finally, other research groups are encouraged to build upon and enhance the evaluation framework. For example, Hassler *et al*. have used the features included in a previous version of the framework (Version 1.2) as a benchmark for an overall systematic review tool (Hassler *et al.*, 2016). The authors compare the features from Version 1.2 of the evaluation framework (see Table 6-14) with their results from a community workshop (designed to identify review tool requirements) and suggest a number of potential additions to the feature set. Looking ahead, similar work by other researchers to expand elements of the evaluation framework would also be beneficial.

8.3 Recommendations and Suggestions for Future Work

In this section, recommendations are provided for both potential tool users and developers about tools to support systematic reviews in software engineering. These recommendations (summarised in Table 8-2) have been made based on all of the research undertaken and reported in this thesis. Some suggestions for future related work are also given.

Recommendation One	At present, reviewers should consider using *SLuRp* to support systematic reviews in software engineering.
Recommendation Two	Reviewers (especially novices) should consider using the *Systematic Review Toolbox* as a resource for findings tools. Furthermore, developers are encouraged to use the resource to catalogue new tools.
Recommendation Three	Users and developers are encouraged to use the evaluation framework to help assess (and select) tools and as a platform for future development.
Recommendation Four	Reviewers are recommended to begin employing more special-purpose tools to support their systematic reviews and report their experiences back to the developer community.
Recommendation Five	Developers are recommended to prioritise support for collaboration and document management when developing tools to support systematic reviews in software engineering.

Table 8-2 Summary of recommendations to support the future use and development of systematic review tools in software engineering

There are tools currently available to support systematic reviews in software engineering. However, they are few in number, immature and not yet widely used. The majority of tools include text mining and visualisation techniques to support study selection, data extraction and data synthesis. A selection of tools which offer support for the whole systematic review process are also available. Of these types, *SLuRp* is (to-date) recommended to users as the most suitable overall tool to support systematic reviews in software engineering. In other disciplines, where tool support for systematic reviews is more mature, established tools including *RevMan* (used primarily in healthcare) and *EPPI-Reviewer* (used primarily across areas of social science) offer support for many systematic review activities and include collaborative working. Since many problems relating to systematic reviews faced in other disciplines are similar to those faced in software engineering, it

may be that some tools in other domains could be useful within software engineering too. However, more work needs to be undertaken to investigate their suitability before any recommendations can be made. To help identify and select appropriate tools, potential users are recommended to query the *Systematic Review Toolbox* (more information about this resource is reported in Chapter Four) and use the evaluation framework as a guide (see Section 7.3.2).

Developers are encouraged to use the evaluation framework to evaluate their tools and act as a foundation for future development. As shown in Table 7-7 (and discussed in Section 7.3.3), tool support for collaboration and document management are the only features in the evaluation framework considered mandatory. These aspects of a tool are, therefore, considered priority features for development. It is recommended that developers consider support for collaboration and facilities for document management at the very start of a tool's development. In addition, developers are encouraged to submit their existing and future tools to the *Systematic Review Toolbox*[1] for users to find easily.

To help support the maturation of systematic review tools in software engineering, reviewers are encouraged to employ more special-purpose tools to support their systematic reviews and report their experiences to developers. As mentioned in Section 7.2.4, more feedback from active users about the effectiveness of tools would provide valuable information to the development and research community. As tool support matures, it is suggested that further work to investigate their use, development and effectiveness, be undertaken. The work reported in this thesis has provided a platform for new research in the topic area to be undertaken.

Based on the findings of this research, a case study to compare the conduct of a full systematic review using an overall tool, with a (traditional) manual approach, is recommended. Further research to investigate the relationship between tool support and the user's knowledge and experience with systematic reviews, is also suggested. It is anticipated, for example, that users with less experience with, or knowledge of the systematic review methodology (e.g. students and novice

[1] http://systematicreviewtools.com/addtool.php

researchers), will have different requirements for tools (and thoughts about the importance of particular features) than more experienced researchers. The flexibility of the evaluation framework aims to accommodate different types of user, by allowing features and importance weightings to be adjusted (see Section 8.2). However, further research to gain a clearer understanding of this relationship, is recommended. In this work, tool support for systematic reviews has been covered quite broadly. However, much of the focus of the investigation and; in particular, the evaluation framework, is placed on tools that aim to support the overall systematic review process (or at least the majority of the stages) in software engineering. It is suggested, therefore, that future work be undertaken to investigate tools which support particular aspects or stages of a systematic review only. In particular, a compatibility study is recommended. A compatibility study is a type of evaluation used to investigate how combinations of different tools work together (Brown & Wallnau, 1996). In the context of systematic review tools, it would be interesting to determine the extent to which different tools (which support different parts of a systematic review) interfere with each other or, whether they can be used together effectively. In addition, investigating how well special-purpose tools integrate with more established, general purpose systems (e.g. reference managers, word processors, spreadsheet packages etc.), would also be beneficial.

References

Aleti, A., Buhnova, B., Grunske, L., Koziolek, A., & Meedeniya, I. (2013). Software architecture optimization methods: a systematic literature review. *Software Engineering, IEEE Transactions on, 39*(5), 658-683.

Ali Babar, M., & Kitchenham, B. (2007). Assessment of a framework for comparing software architecture analysis methods. In *Proceedings of the 2007 International Conference on Evaluation and Assessment in Software Engineering*.

Allen, C., & Richmond, K. (2011). The Cochrane Collaboration: International activity within Cochrane Review Groups in the first decade of the twenty-first century. *Journal of Evidence-Based Medicine*, *4*(1), 2-7.

Austin, L. L., & Toth, E. L. (2011). Exploring ethics education in global public relations curricula: Analysis of international curricula descriptions and interviews with public relations educators. *Public Relations Review*, *37*(5), 506-512.

Babar, M. A., & Gorton, I. (2004). Comparison of scenario-based software architecture evaluation methods. In *Proceedings of the 2004 Asia-Pacific Software Engineering Conference* (pp. 600-607).

Babar, M. A., & Zhang, H. (2009). Systematic literature reviews in software engineering: preliminary results from interviews with researchers. In *Proceedings of the 2009 International Symposium on Empirical Software Engineering and Measurement* (pp. 346-355).

Babar, M. A., Zhu, L., & Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods. In *Proceedings of the 2004 Australian Software Engineering Conference, 2004* (pp. 309-318).

Badampudi, D., Wohlin, C., & Petersen, K. (2015). Experiences from using snowballing and database searches in systematic literature studies. In *Proceedings of the 2015 International Conference on Evaluation and Assessment in Software Engineering* (pp. 17-26).

Baltussen, R., Youngkong, S., Paolucci, F., & Niessen, L. (2010). Multi-criteria decision analysis to prioritize health interventions: capitalizing on first experiences. *Health Policy*, *96*(3), 262-264.

Beecham, S., OLeary, P., Richardson, I., Baker, S., & Noll, J. (2013). Who are we doing global software engineering research for?. In *Proceedings of the 2013 International Conference on Global Software Engineering* (pp. 41-50).

Belton, V., & Stewart, T. (2002). *Multiple criteria decision analysis: an integrated approach*. Springer Science & Business Media.

Bezerra, F., Favacho, C. H., Souza, R., & de Souza, C. (2014) Towards supporting systematic mappings studies: an automatic snowballing approach. In *Proceedings of the 2014 Brazilian Symposium on Databases.*

Bowes, D., Hall, T., & Beecham, S. (2012). SLuRp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In *Proceedings of the 2012 International Workshop on Evidential Assessment of Software Technologies* (pp. 33-36).

Bozdağ, C. E., Kahraman, C., & Ruan, D. (2003). Fuzzy group decision making for selection among computer integrated manufacturing systems. *Computers in Industry, 51*(1), 13-29.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Systems and Software, 80*(4), 571-583.

Brooks Jr., F. P. (1987). No silver bullet: essences and accidents of software engineering. *IEEE Computer 20*(4), 10-19.

Brown, A. W., & Wallnau, K. C. (1996). A framework for evaluating software technology. *IEEE Software, 13*(5), 39-49.

Budgen, D., & Brereton, P. (2006). Performing systematic literature reviews in software engineering. In *Proceedings of the 2006 International Conference on Software Engineering* (pp. 1051-1052).

Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. (2008). Using mapping studies in software engineering. In *Proceedings of the 2008 Psychology of Programming Interest Group Proceedings of PPIG* (pp. 195-204).

Carver, J., Hassler, E., Hernandes, E & Kraft, N. (2013) Identifying barriers to the systematic literature review process. In *Proceedings of the 2013 International Symposium on Empirical Software Engineering and Measurement* (pp. 203-212).

Cegala, D. J. (2011). An exploration of factors promoting patient participation in primary care medical interviews. *Health Communication, 26*(5), 427-436.

Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., & Shekelle, P. G. (2006). Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. *Annals of internal medicine, 144*(10), 742-752.

Clough, P., & Nutbrown, C. (2012). *A student's guide to methodology*. SAGE Publications

Cochran, J. K., & Chen, H. N. (2005). Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis. *Computers & Operations Research, 32*(1), 153-168.

Cohen, A. M., Adams, C. E., Davis, J. M., Yu, C., Yu, P. S., & Smalheiser, N. R. (2010). Evidence-based medicine, the essential role of systematic reviews, and the need for automated text mining tools. In *Proceedings of the 2010 ACM International Health Informatics Symposium* (pp. 376-380).

Collier, K., Carey, B., Sautter, D., & Marjaniemi, C. (1999). A methodology for evaluating and selecting data mining software. In *Proceedings of the 1999 International Conference on Systems Sciences* (pp. 1-11).

Colombo, E., & Francalanci, C. (2004). Selecting CRM packages based on architectural, functional, and cost requirements: empirical validation of a hierarchical ranking model. *Requirements engineering, 9*(3), 186-203.

Cook, D. J., Mulrow, C. D., & Haynes, R. B. (1997). Systematic reviews: synthesis of best evidence for clinical decisions. *Annals of internal medicine, 126*(5), 376-380.

Cruzes, D., Mendonça, M., Basili, V., Shull, F., & Jino, M. (2007). Using context distance measurement to analyze results across studies. In *Proceedings of the 2007 International Symposium on Empirical Software Engineering and Measurement* (pp. 235-244).

Cruzes, D., Mendonça, M., Basili, V., Shull, F., & Jino, M. (2007). Automated information extraction from empirical software engineering literature: is that possible?. In *Proceedings of the 2007 International Symposium on Empirical Software Engineering and Measurement* (pp. 491-493).

da Silva, F. Q., Santos, A. L., Soares, S., França, A. C. C., Monteiro, C. V., & Maciel, F. F. (2011). Six years of systematic literature reviews in software engineering: an updated tertiary study. *Information and Software Technology*, *53*(9), 899-913.

Davis, L., & Williams, G. (1994). Evaluating and selecting simulation software using the analytic hierarchy process. *Integrated Manufacturing Systems*, *5*(1), 23-32.

de Zwaan, M., Hilbert, A., Swan-Kremeier, L., Simonich, H., Lancaster, K., Howell, L. M., & Mitchell, J. E. (2010). Comprehensive interview assessment of eating behaviour 18–35 months after gastric bypass surgery for morbid obesity. *Surgery for Obesity and Related Diseases*, *6*(1), 79-85.

Deb, S., Chaplin, R., Sohanpal, S., Unwin, G., Soni, R., & Lenotre, L. (2008). The effectiveness of mood stabilizers and antiepileptic medication for the management of behaviour problems in adults with intellectual disability: a systematic review. *Journal of Intellectual Disability Research*, *52*(2), 107-113.

Dejaeger, K., Verbeke, W., Martens, D., & Baesens, B. (2012). Data mining techniques for software effort estimation: a comparative study. *IEEE Transactions on Software Engineering,*, *38*(2), 375-397.

Denscombe, M. (2014). *The good research guide: for small-scale social research projects*. McGraw-Hill Education (UK).

DePanfilis, D., & Zlotnik, J. L. (2008). Retention of front-line staff in child welfare: a systematic review of research. *Children and Youth Services Review*, *30*(9), 995-1008.

du Plessis, A. L. (1993). A method for CASE tool evaluation. *Information & Management*, *25*(2), 93-102.

Dujmović, J., & Kadaster, M. (2002). A technique and tool for software evaluation. In *Proceedings of the 2002 International Conference Software Engineering Applications*, (pp. 374, 246).

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: a systematic review. *Information and Software Technology*, *50*(9), 833-859.

Dybå, T., Kampenes, V. B., & Sjøberg, D. I. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, *48*(8), 745-755.

Dybå, T., Kitchenham, B., & Jørgensen, M. (2005). Evidence-based software engineering for practitioners. *IEEE Software*, *22*(1), 58-65.

Elamin, M. B., Flynn, D. N., Bassler, D., Briel, M., Alonso-Coello, P., Karanicolas, P. J., & Montori, V. M. (2009). Choice of data extraction tools for systematic reviews depends on resources and review complexity. *Journal of Clinical Epidemiology*, *62*(5), 506-510.

Elliott, J., Sim, I., Thomas, J., Owens, N., Dooley, G., Riis, J., Wallace, B., Thomas, J., Noel-Storr, A., Rada, G., Struthers, C., Howe, T., MacLehose, H., Brandt, L., Kunnamo, I., Mavergames, C. #CochraneTech: technology and the future of systematic reviews[editorial] (2014). *Cochrane Database of Systematic Reviews*.

Felizardo, K. R., Andery, G. F., Paulovich, F. V., Minghim, R., & Maldonado, J. C. (2012). A visual analysis approach to validate the selection review of primary studies in systematic reviews. *Information and Software Technology*. *54*(10), 1079-1091.

Felizardo, K. R., MacDonell, S. G., Mendes, E., & Maldonado, J. C. (2012). A systematic mapping on the use of visual data mining to support the conduct of systematic literature reviews. *Journal of Software 7*(2), 450-461.

Felizardo, K. R., MacDonell, S. G., Mendes, E., & Maldonado, J. C. (2012). A systematic mapping on the use of visual data mining to support the conduct of systematic literature reviews. *Journal of Software 7*(2), 450-461.

Felizardo, K. R., Nakagawa, E. Y., MacDonell, S. G., & Maldonado, J. C. (2014). A visual analysis approach to update systematic reviews. In *Proceedings of the 2014 International Conference on Evaluation and Assessment in Software Engineering* (pp. 4-13).

Felizardo, K. R., Nakawaga, E. Y., Feitosa, D., Minghim, R., & Maldonado, J. C. (2010). An approach based on visual text mining to support categorization and classification in the systematic mapping. In *Proceedings of the 2010 International Conference on Evaluation and Assessment in Software Engineering* (pp. 1-10).

Felizardo, K. R., Riaz, M., Sulayman, M., Mendes, E., MacDonell, S. G., & Maldonado, J. C. (2011). Analysing the use of graphs to represent the results of systematic reviews in software engineering. In *Proceedings of the 2011 Brazilian Symposium on Software Engineering* (pp. 174-183).

Felizardo, K. R., Salleh, N., Martins, R. M., Mendes, E., MacDonell, S. G., & Maldonado, J. C. (2011). Using visual text mining to support the study selection activity in systematic literature reviews. In *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement* (pp. 77-86).

Fernández-Sáez, M. G. Bocco, F.P. Romero. (2010). SLR-tool - a tool for performing systematic literature reviews. In *Proceedings of the 2010 International Conference on Software and Data Technologies* (pp. 157-166).

Geersing, G. J., Bouwmeester, W., Zuithoff, P., Spijker, R., Leeflang, M., & Moons, K. G. (2012). Search filters for finding prognostic and diagnostic prediction studies in Medline to enhance systematic reviews. *PLoS One*, *7*(2), e32844.

Ghafari, M., Saleh, M., & Ebrahimi, T. (2012). A federated search approach to facilitate systematic literature review in software engineering. *International Journal of Software Engineering and Applications 3*(2), 13-24.

Gomm, R. (2004). *Social research methodology*. New York: Palgrave Macmillan.

Grimán, A., Pérez, M., Mendoza, L., & Losavio, F. (2006). Feature analysis for architectural evaluation methods. *Journal of Systems and Software*, *79*(6), 871-888.

Grimshaw, J. M., & Russell, I. T. (1993). Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. *The Lancet*, *342*(8883), 1317-1322.

Haddaway, N. R., & Pullin, A. S. (2014). The policy role of systematic reviews: past, present and future. *Springer Science Reviews*, *2*(1-2), 179-183.

Hakim, C. (2000). *Research design: successful designs for social and economic research*. Psychology Press.

Harris, P. E., Cooper, K. L., Relton, C., & Thomas, K. J. (2012). Prevalence of complementary and alternative medicine (CAM) use by the general population: a systematic review and update. *International Journal of Clinical Practice*, *66*(10), 924-939.

Hassler, E., Carver, J. C., Hale, D., & Al-Zubidy, A. (2016). Identification of SLR tool needs–results of a community workshop. *Information and Software Technology*, *70*, 122-129.

Hearn, J., & Higginson, I. J. (1998). Do specialist palliative care teams improve outcomes for cancer patients? a systematic literature review. *Palliative Medicine*, 12(5), 317-332.

Hedberg, H., & Lappalainen, J. (2005). A preliminary evaluation of software inspection tools, with the DESMET method. In *Proceedings of the 2005 International Conference on Quality Software* (pp. 45-52).

Hernandes, E., Zamboni, A., Fabbri, S., & Di Thommazo, A. (2012). Using GQM and TAM to evaluate StArt–a tool that supports systematic review. *CLEI Electronic Journal*, *15*(1), 13-25.

Higgins, J. P. (2008). Cochrane handbook for systematic reviews of interventions. Chichester: Wiley-Blackwell.

Hlupic, V. (1997). Simulation software selection using SimSelect. *Simulation*, *69*(4), 231-239.

Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, *53*(5), 521-534.

Hossain, E., Babar, M. A., & Paik, H. Y. (2009). Using Scrum in global software development: a systematic literature review. In *Proceedings of the 2009 IEEE Conference on Global Software Engineering* (pp. 175-184).

Hove, S. E., & Anda, B. (2005). Experiences from conducting semi-structured interviews in empirical software engineering research. In *Proceedings of the 2005 International Symposium on Software Metrics* (pp. 1-10).

Howell, D., & Kaplan, L. (2015). Statewide survey of healthcare professionals: management of patients with chronic noncancer pain. *Journal of Addictions Nursing*, *26*(2), 86-92.

Husereau, D., Boucher, M., & Noorani, H. (2010). Priority setting for health technology assessment at CADTH. *International Journal of Technology Assessment in Healthcare*, *26*(3), 341-347.

Imtiaz, S., Bano, M., Ikram, N., & Niazi, M. (2013). A tertiary study: experiences of conducting systematic literature reviews in software engineering. In *Proceedings of the 2013 International Conference on Evaluation and Assessment in Software Engineering* (pp. 177-182).

Iyer, J., & Richards, D. (2004). Evaluation framework for tools that manage requirements inconsistency. In *Proceedings of the 2004 Australian Workshop on Requirements Engineering.*

Jadhav, A. S., & Sonar, R. M. (2009). Evaluating and selecting software packages: a review. *Information and Software Technology, 51*(3), 555-563.

Jalali, S., & Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the 2012 International Symposium on Empirical Software Engineering and Measurement* (pp. 29-38).

Johnston, J. M., Leung, G. M., Tin, K. Y., Ho, L. M., Lam, W., & Fielding, R. (2004). Evaluation of a handheld clinical decision support tool for evidence-based learning and practice in medical undergraduates. *Medical Education*,*38*(6), 628-637.

Jones, S. S., Rudin, R. S., Perry, T., & Shekelle, P. G. (2014). Health information technology: an updated systematic review with a focus on meaningful use. *Annals of Internal Medicine*, *160*(1), 48-54.

Kamdar, B. B., Shah, P. A., Sakamuri, S., Kamdar, B. S., & Oh, J. (2015). A novel search builder to expedite search strategies for systematic reviews. *International Journal of Technology Assessment in Healthcare*, 1-3.

Karlsson, C., Taylor, M., & Taylor, A. (2010). Integrating new technology in established organizations: a mapping of integration mechanisms. *International Journal of Operations & Production Management*, *30*(7), 672-699.

Karunananthan, S., Wolfson, C., Bergman, H., Béland, F., & Hogan, D. (2009). A multidisciplinary systematic literature review on frailty: overview of the methodology used by the Canadian Initiative on Frailty and Aging. *BMC Medical Research Methodology*, 9(1), 68.

Khalifelu, Z. A., & Gharehchopogh, F. S. (2012). Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation. *Procedia Technology*, *1*, 65-71.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Technical Report*, Keele University.

Kitchenham, B. A. (1996). Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods. *ACM SIGSOFT Software Engineering Notes*, *21*(1), 11-14.

Kitchenham, B. A. (1997). Evaluating software engineering methods and tools, part 7: planning feature analysis evaluation. *ACM SIGSOFT Software Engineering Notes*, *22*(4), 21-24.

Kitchenham, B. A., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *EBSE Technical Report*.

Kitchenham, B. A., & Jones, L. (1997). Evaluating software engineering methods and tool part 5: the influence of human factors. *ACM SIGSOFT Software Engineering Notes*, *22*(1), 13-15.

Kitchenham, B. A., Budgen, D., & Brereton, O. P. (2010). The value of mapping studies: a participant observer case study. In *Proceedings of the 2010 International Conference on Evaluation and Assessment in Software Engineering* (pp. 25-33).

Kitchenham, B. A., Dybå, T., & Jørgensen, M. (2004). Evidence-based software engineering. In *Proceedings of the 2004 International Conference on Software Engineering* (pp. 273-281).

Kitchenham, B. A., Dybå, T., & Jørgensen, M. (2004). Evidence-based software engineering. In *Proceedings of the 2004 International Conference on Software Engineering* (pp. 273-281).

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, *55*(12), 2049-2075.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and Software Technology*, *51*(1), 7-15.

Kitchenham, B., Brereton, P., & Budgen, D. (2012). Mapping study completeness and reliability-a case study. In *Proceedings of the 2012 International Conference on Evaluation and Assessment in Software Engineering*. (pp. 126-135).

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. *Information and Software Technology*, *52*(8), 792-805.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-Based Software Engineering and Systematic Reviews* (Vol. 4). CRC Press.

Le Blanc, L. A., & Jelassi, M. T. (1989). DSS software selection: a multiple criteria decision methodology. *Information & Management*, *17*(1), 49-65.

Le Blanc, L. A., & Korn, W. M. (1992). A structured approach to the evaluation and selection of CASE tools. In *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's* (pp. 1064-1069).

Leslie, S. J., Hartswood, M., Meurig, C., McKee, S. P., Slack, R., Procter, R., & Denvir, M. A. (2006). Clinical decision support software for management of chronic heart failure: Development and evaluation. *Computers in Biology and Medicine*, *36*(5), 495-506.

Lethbridge, T. C., Sim, S. E., & Singer, J. (2005). Studying software engineers: data collection techniques for software field studies. *Empirical Software Engineering*, *10*(3), 311-341.

Lin, H. Y. (2006). *Data warehouse system evaluation and selection decisions*. Doctoral Dissertation, Taiwan.

Major, L., Kyriacou, T., & Brereton, P. (2014). The effectiveness of simulated robots for supporting the learning of introductory programming: a multi-case case study. *Computer Science Education*, *24*(2-3), 193-228.

Malheiros, V., Hohn, E., Pinho, R., & Mendonca, M. (2007). A visual text mining approach for systematic reviews. In *Proceedings of the 2007 International Symposium on Empirical Software Engineering and Measurement*, (pp. 245-254).

Marshall, C., & Brereton, P. (2013) Tools to support systematic literature reviews in software engineering: a mapping study. In *Proceedings of the 2013 International Symposium on Empirical Software Engineering and Measurement* (pp. 296-299).

Marshall, C., & Brereton, P. (2015). Systematic review toolbox: a catalogue of tools to support systematic reviews. In *Proceedings of the 2015 International Conference on Evaluation and Assessment in Software Engineering* (pp. 23-26).

Marshall, C., Brereton, P., & Kitchenham, B. (2014). Tools to support systematic reviews in software engineering: a feature analysis. In *Proceedings of the 2014 International Conference on Evaluation and Assessment in Software Engineering* (pp. 139-148).

Marshall, C., Brereton, P., & Kitchenham, B. (2015). Tools to support systematic reviews in software engineering: a cross-domain survey using semi-structured interviews. In *Proceedings of the 2015 International Conference on Evaluation and Assessment in Software Engineering* (pp. 26-31).

Mergel, G. D., Silveira, M. S., & da Silva, T. S. (2015). A method to support search string building in systematic literature reviews through visual text mining. In *Proceedings of the 2015 Annual ACM Symposium on Applied Computing* (pp. 1594-1601).

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). *Qualitative data analysis: A methods sourcebook*. SAGE Publications.

Misra, S. K. (1990). Analysing CASE system characteristics: evaluative framework. *Information and Software Technology*, *32*(6), 415-422.

Moher, D., & Tsertsvadze, A. (2006). Systematic reviews: when is an update an update?. *The Lancet*, *367*(9514), 881-883.

Molléri, J. S., & Benitti, F. B. V. (2015). SESRA: a web-based automated tool to support the systematic literature review process. In *Proceedings of the 2015 International Conference on Evaluation and Assessment in Software Engineering* (pp. 24-33).

Mulrow, C. D. (1994). Rationale for systematic reviews. *BMJ: British Medical Journal*, *309*(6954), 597.

Ngai, E. W., & Chan, E. W. C. (2005). Evaluation of knowledge management tools using AHP. *Expert Systems with Applications*, *29*(4), 889-899.

Nikoukaran, J., & Paul, R. J. (1999). Software selection for simulation in manufacturing: a review. *Simulation Practice and Theory*, *7*(1), 1-14.

Nutt, D. J., King, L. A., & Phillips, L. D. (2010). Drug harms in the UK: a multicriteria decision analysis. *The Lancet*, *376*(9752), 1558-1565.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., & Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, *4*(1), 5.

Oates, B. J. (2006). *Researching Information Systems and Computing*. Sage Publications.

Octaviano, F. R., Felizardo, K. R., Maldonado, J. C., & Fabbri, S. C. (2014). Semi-automatic selection of primary studies in systematic literature reviews: is it reasonable?. *Empirical Software Engineering*, 1-20.

Pai, M., Zwerling, A., & Menzies, D. (2008). Systematic review: T-cell–based assays for the diagnosis of latent tuberculosis infection: an update. *Annals of Internal Medicine*, *149*(3), 177-184.

Parrish, A. M., Yeatman, H., Iverson, D., & Russell, K. (2012). Using interviews and peer pairs to better understand how school environments affect young children's playground physical activity levels: a qualitative study. *Health Education Research*, *27*(2), 269-280.

Patel, N., & Hlupic, V. (2002). A methodology for the selection of knowledge management (KM) tools. In *Proceedings of the 2002 International Conference on Information Technology Interfaces.*

Patton, M. Q. (1990). *Qualitative Evaluation and Research Methods* . SAGE Publications.

Patton, M. Q. (2005). *Qualitative Research*. John Wiley & Sons, Ltd.

Perez, M., & Rojas, T. (2000). Evaluation of Workflow-type software products: a case study. *Information and Software Technology*, *42*(7), 489-503.

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: an update. *Information and Software Technology*, *64*, 1-18.

Potter, J., & Hepburn, A. (2005). Qualitative interviews in psychology: problems and possibilities. *Qualitative Research in Psychology*, *2*(4), 281-307.

Powell, J., Inglis, N., Ronnie, J., & Large, S. (2011). The characteristics and motivations of online health information seekers: cross-sectional survey and qualitative interview study. *Journal of Medical Internet Research*, *13*(1) e20.

Preuveneers, D., & Novais, P. (2012). A survey of software engineering best practices for the development of smart applications in ambient intelligence. *Journal of Ambient Intelligence and Smart Environments*, *4*(3), 149-162.

Radjenovic, D., Herico, M., Torkar, R., & Zivkovic, A. (2013). Software fault prediction metrics: a systematic literature review. *Information and Software Technology*, *55*(8), 1397-1418.

Ramampiaro, H., Cruzes, D., Conradi, R., & Mendona, M. (2010). Supporting evidence-based software engineering with collaborative information retrieval. In *Proceedings of the 2010 International Conference on Collaborative Computing: Networking, Applications and Worksharing* (pp. 1-5).

Rea, L. M., & Parker, R. A. (2014). *Designing and conducting survey research: A comprehensive guide*. John Wiley & Sons.

Riaz, M., Sulayman, M., Salleh, N., & Mendes, E. (2010). Experiences conducting systematic reviews from novices' perspective. In *Proceedings of the 2010 International Conference on Evaluation and Assessment in Software Engineering* (pp. 1-10).

Robson, C. (2011). *Real World Research*. 3rd Edition. John Wiley & Sons.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131-164.

Saaty, T. L. (1988). *What is the analytic hierarchy process?* (pp. 109-121). Springer Berlin Heidelberg.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences, 1*(1), 83-98.

Sackett, D. L., Rosenberg, W. M., Gray, J. A., Haynes, R. B., & Richardson, W. S. (1996). Evidence based medicine: what it is and what it isn't. *BMJ: British Medical Journal*, 312(7023), 71.

Santos, R. E., & Da Silva, F. Q. (2013). Motivation to perform systematic reviews and their impact on software engineering practice. In *Proceedings of the 2013 International Symposium on Empirical Software Engineering and Measurement* (pp. 292-295).

Sarkis, J., & Talluri, S. (2004). Evaluating and selecting e-commerce software and communication systems for a supply chain. *European Journal of Operational Research*, *159*(2), 318-329.

Schulte, A. G., Buchalla, W., Huysmans, M. C., Amaechi, B. T., Sampaio, F., Vougiouklakis, G., & Pitts, N. B. (2011). A survey on education in cariology for undergraduate dental students in Europe. *European Journal of Dental Education*, *15*(s1), 3-8.

Shemilt, I., Simon, A., Hollands, G. J., Marteau, T. M., Ogilvie, D., O'Mara-Eves, A., & Thomas, J. (2014). Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Research Synthesis Methods*, *5*(1), 31-49.

Shieh, R. S. (2012). The impact of Technology-Enabled Active Learning (TEAL) implementation on student learning and teachers' teaching in a high school context. *Computers & Education*, *59*(2), 206-214.

Skoglund, M., & Runeson, P. (2009). Reference-based search strategies in systematic reviews. In *Proceedings of the 2009 International Conference on Evaluation and Assessment in Software Engineering*.

Stamelos, I., & Tsoukias, A. (2003). Software evaluation problem situations. *European Journal of Operational Research*, *145*(2), 273-286.

Staples, M., & Niazi, M. (2007). Experiences using systematic review guidelines. *Journal of Systems and Software*, *80*(9), 1425-1437.

Sun, Y., Yang, Y., Zhang, H., Zhang, W., & Wang, Q. (2012). Towards evidence-based ontology for supporting systematic literature review. In *Proceedings of the 2012 International Conference on Evaluation and Assessment in Software Engineering*, (pp. 171-175).

Sutton, A. J., Donegan, S., Takwoingi, Y., Garner, P., Gamble, C., & Donald, A. (2009). An encouraging assessment of methods to inform priorities for updating systematic reviews. *Journal of Clinical Epidemiology*, *62*(3), 241-251.

Thokala, P., & Duenas, A. (2012). Multiple criteria decision analysis for health technology assessment. *Value in Health*, *15*(8), 1172-1181.

Thomas, J., McNaught, J., & Ananiadou, S. (2011). Applications of text mining within systematic reviews. *Research Synthesis Methods*, *2*(1), 1-14.

Tomassetti, F., Rizzo, G., Vetro, A., Ardito, L., Torchiano, M., & Morisio, M. (2011). Linked data approach for selection process automation in systematic reviews. In *Proceedings of the 2011 International Conference on Evaluation and Assessment in Software Engineering* (pp. 31-35).

Torres, J. A. S., Cruzes, D. S., & do Nascimento Salvador, L. (2012). Automatic results identification in software engineering papers. Is it possible?. In *Proceedings of the 2012 International Conference on Computational Science and Its Applications* (pp. 108-112).

Torres, J., Cruzes, D., & Salvador, L. Automatically locating results to support systematic reviews in software engineering (2013). In *Proceedings of the 2013 Iberoamerican Conference on Software Engineering.*

Tsafnat, G., Dunn, A., Glasziou, P., & Coiera, E. (2013). The automation of systematic reviews. *BMJ: British Medical Journal*, *346*, f139.

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., & Coiera, E. (2014). Systematic review automation technologies. *Systematic Reviews*, *3*(1), 74.

Tyndale, P. (2002). A taxonomy of knowledge management software tools: origins and applications. *Evaluation and Program Planning*, *25*(2), 183-190.

Vanderlinde, R., & van Braak, J. (2010). The gap between educational research and practice: views of teachers, school leaders, intermediaries and researchers. *British Educational Research Journal*, *36*(2), 299-316.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J., Schmid, C. H., Bertram, L., & Trikalinos, T. A. (2012). Toward modernizing the systematic review pipeline in genetics: efficient updating via data mining. *Genetics in medicine*,*14*(7), 663-669.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*.

Wohlin, C., Runeson, P., Neto, P. A. D. M. S., Engström, E., do Carmo Machado, I., & De Almeida, E. S. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, *86*(10), 2594-2610.

Zelkowitz, M. V., & Wallace, D. (1998). Validating the benefit of new software technology. *Software Quality Practitioner*, *1*(1).

Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, *53*(6), 625-637.

Zhou, Y., Zhang, H., Huang, X., Yang, S., Babar, M. A., & Tang, H. (2015). Quality assessment of systematic reviews in software engineering: a tertiary study. In *Proceedings of the 2015 International Conference on Evaluation and Assessment in Software Engineering*.

Appendix A1 – Known papers used to validate the search

Paper ID	Title	Paper Ref.
P10	SLuRp: A Tool to Help Large Complex Systematic Literature Reviews Deliver Valid and Rigorous Results	Bowes *et al.*, 2012
P05	SLR-Tool – A Tool for Performing Systematic Literature Reviews	Fernández-Sáez *et al*., 2010
P01	A Visual Text Mining Approach for Systematic Reviews	Malheiros *et al*., 2007
P02	An Approach Based on Visual Text Mining to Support Categorization and Classification in the Systematic Mapping	Felizardo *et al*., 2010
P03	Using Visual Text Mining to Support the Study Selection Activity in Systematic Literature Reviews.	Felizardo *et al*., 2011
P04	A Visual Analysis Approach to Validate the Selection Review of Primary Studies in Systematic Reviews	Felizardo *et al*., 2012
P07	Using Context Distance Measurement to Analyze Results across Studies	Cruzes *et al*., 2007
P08	Automated Information Extraction from Empirical Software Engineering Literature: Is that Possible?	Cruzes *et al*., 2007
P09	Automatic Results Identification in Software Engineering Papers. Is it Possible?	Torres *et al*., 2012
P11	Analysing the Use of Graphs to Represent the Results of Systematic Reviews in Software Engineering	Felizardo *et al*., 2011
P06	Extracting Information from Experimental Software Engineering Papers	Cruzes *et al.*, 2007

Appendix A2 – Excluded papers

Paper ID	Title	Paper Ref.
P06	Extracting Information from Experimental Software Engineering Papers	Cruzes *et al.*, 2007
P13	Synthesizing Evidence in Software Engineering Research	Cruzes & Dybå, 2010
P17	An Empirical Assessment of A Systematic Search Process for Systematic Reviews	Zhang *et al.*, 2011
P18	Supporting Systematic Reviews using Text Mining	Ananiadou *et al.*, 2009
P19	A Systematic Mapping on the Use of Visual Data Mining to Support the Conduct of Systematic Literature Reviews	Felizardo *et al.*, 2012
P20	Visual Analysis of Data from Empirical Studies	Garcia *et al.*, 2004
P21	Reference-based Search Strategies in Systematic Reviews	Skoglund & Runeson, 2009

Appendix A3 – Quality assessment results

Paper ID	Research	Aim	Context	Research Design	Recruitment Strategy	Control Group	Data Collection	Data Analysis	Relationship	Findings	Value	Applicable	Total Quality Score
P01	1	0.5	1	1	0.5	1	1	0.5	0	1	1	11	**8.5**
P02	1	1	1	1	1	1	1	0.5	0.5	1	1	11	**10**
P03	1	1	0	1	0.5	1	1	0.5	0.5	1	1	11	**8.5**
P04 (Study 1)	1	1	1	1	0	1	0.5	0	0	0.5	0.5	11	**6.5**
P04 (Study 2)	1	1	1	1	0.5	1	1	0.5	0	1	1	11	**9**
P07	1	1	1	0.5	1	1	1	1	1	0.5	0.5	11	**9.5**
P09	1	1	0.5	1	0.5	1	1	0.5	N/A	1	0.5	10	**8**
P11	1	1	1	1	1	1	1	1	0	1	1	11	**10**
P12	1	1	0.5	1	0.5	1	1	0.5	0	0.5	0.5	11	**7.5**
P14	1	0.5	1	1	0.5	1	1	1	N/A	1	0.5	10	**8.5**
P15	1	0.5	1	0	0.5	N/A	1	1	0	0.5	0.5	10	**6**

Appendix A4 – Email invitation sent to participants

"Dear [Participant Name],

I'm Chris, A PhD Student (Keele University) researching tool support for systematic reviews. We're aiming to develop an evaluation framework for an end-to-end tool which supports the entire systematic review process within software engineering.

So far, we've performing a mapping study which identified a range of tools to support the systematic review process. In addition, we have developed a preliminary framework and used this to evaluate a selection of current 'overall' support tools.

To progress this work, I'd like to speak with people who have performed systematic reviews in other domains where the method is also used and is more established.

As someone that has experience with the systematic review methodology, would you be interested in participating in a short interview? The interview would focus on the role of systematic reviews within your domain, known tools that are used to support systematic reviews and your personal experience undertaking systematic reviews.

Look forward to hearing from you.

Best wishes,

Chris."

Appendix A5 – Interview preparation sheet

Keele University

Interview Preparation Document (Information Sheet)

Study Title

Tool Support for Systematic Reviews in Software Engineering

Aims of the Research

The aim of this interview is to gather information about the availability, use, potential and effectiveness of automated tools, which provide support for systematic reviews.

How long will the interview take?

The interview should take no more than one hour to complete.

What will I be asked about?

The interview will focus on discussing your thoughts and experience using tools to support the conduct of a systematic review. However, we are also interested in learning about the systematic review process particularly within your discipline. Questions will be asked on the following topics:

- The role of systematic reviews within your discipline.
- Known tools that are used to support the conduct of systematic reviews within your domain.
- Your personal experience undertaking systematic reviews (with/without the help of tools).

How will the information about me be used?

The data collected will contribute toward the development of a refined framework, for an overall tool to support SRs.

Who will have access to the information about me?

The only people who will have access to the data collected are the members of the research team conducting this study. This includes Christopher Marshall (PhD Researcher), Prof Pearl Brereton (Lead Supervisor) and Prof Barbara Kitchenham (Second Supervisor). All data will be made anonymous during the analysis process for future reports and research projects. Notes taken during the interview process will be stored on a password protected computer. Audio recordings (providing you have agreed to for the interview to be recorded) will be stored in a locked filing cabinet.

Who is funding the research?

This research is partially supported by Keele University's Environmental, Physical Sciences and Applied Mathematics (EPSAM) Research Institute.

What if there is a problem?

If you have a concern about any aspect of this study, you may wish to speak to the researchers who will do their best to answer your questions. You should contact Christopher Marshall on +44(0)1782 733593 or c.marshall@keele.ac.uk. Alternatively, if you do not wish to contact this researcher you may contact Prof Pearl Brereton (lead supervisor) on +44(0)1782 733079 or o.p.brereton@keele.ac.uk

If you remain unhappy about the research and/or wish to raise a complaint about any aspect of the way that you have been approached or treated during the course of this study please write to Nicola Leighton who is the University's contact for complaints regarding research at the following address:

Nicola Leighton
Research Governance Office
Research & Enterprise Services
Dorothy Hodgkin Building
Keele University
ST5 5BG
E-mail: n.leighton@uso.keele.ac.uk
Tel: +44 (0)1782 733306.

Contact for Further Information

Christopher Marshall
PhD Researcher
School of Computing and Mathematics
Keele University
ST5 5BG
E-mail: c.marshall@keele.ac.uk
Tel: +44 (0)1782 733593

Thank you for agreeing to be interviewed.

Appendix A6 – Consent form 1

Keele University

CONSENT FORM

Title of Project: Tool Support for Systematic Reviews in Software Engineering

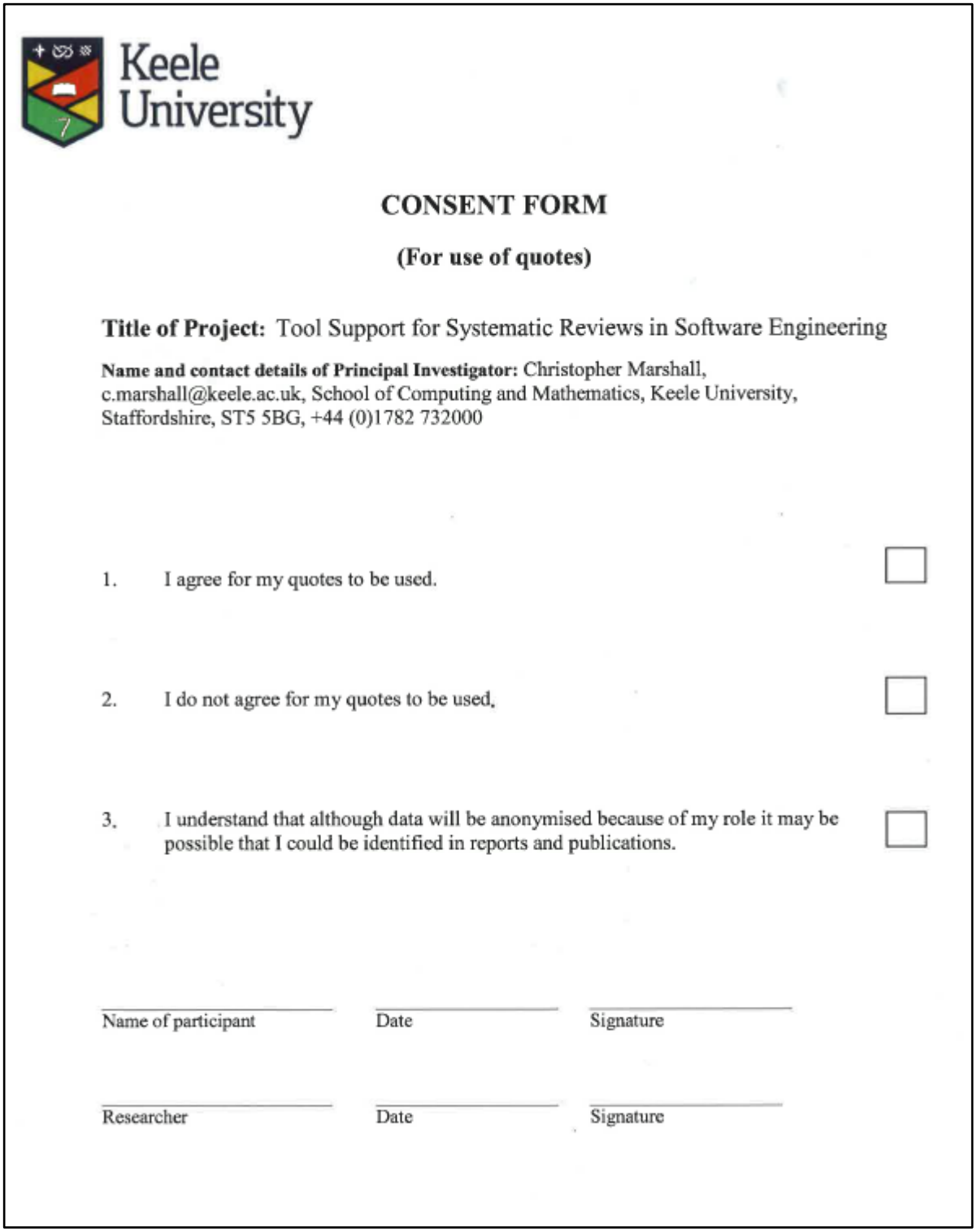
Name and contact details of Principal Investigator: Christopher Marshall, c.marshall@keele.ac.uk, School of Computing and Mathematics, Keele University, Staffordshire, ST5 5BG, +44 (0)1782 732000

		Please tick the box if you agree with the statement
1.	I confirm that I have read and understood the interview preparation sheet for the above study and have had the opportunity to ask questions.	☐
2.	I understand that my participation is voluntary and that I am free to withdraw at any time.	☐
3.	I agree to take part in this study.	☐
4.	I understand that data collected about me during this study will be anonymised before it is submitted for publication.	☐
5.	I understand that although data will be anonymised because of my role it may be possible that I could be identified in reports and publications	☐
6.	I agree to the interview being audio recorded.	☐
7.	I agree to allow the dataset collected to be used for future research projects.	☐
9.	I agree to be contacted about possible participation in future research project.	☐

Name of participant	Date	Signature
Researcher	Date	Signature

For Focus Groups/Interviews*
If you consent to participate in this study, it should be drawn to your attention that the researcher has a professional obligation to act upon any aspects of poor practice and/or unprofessional behaviour that may be disclosed during the research activity. Researchers should use the appropriate reporting mechanisms if they have witnessed or experienced poor practice and/or unprofessional behaviour.

Appendix A7 – Consent form 2 (use of quotes)



Keele University

CONSENT FORM

(For use of quotes)

Title of Project: Tool Support for Systematic Reviews in Software Engineering

Name and contact details of Principal Investigator: Christopher Marshall, c.marshall@keele.ac.uk, School of Computing and Mathematics, Keele University, Staffordshire, ST5 5BG, +44 (0)1782 732000

1. I agree for my quotes to be used. ☐

2. I do not agree for my quotes to be used. ☐

3. I understand that although data will be anonymised because of my role it may be possible that I could be identified in reports and publications. ☐

Name of participant	Date	Signature
Researcher	Date	Signature

Appendix A8 – Ethical approval confirmation letter

Keele University

RESEARCH AND ENTERPRISE SERVICES

22nd May 2014

Christopher Marshall
School of Computing and Mathematics
Keele University

Dear Christopher,

Re: Tools to support systematic reviews in software engineering: A survey using semi-structured interviews

Thank you for submitting your revised application for review. I am pleased to inform you that your application has been approved by the Ethics Review Panel. The following documents have been reviewed and approved by the panel as follows:

Document	Version	Date
Summary of Proposal	1.0	April 2014
Information Sheets	1.0	April 2014
Consent Form	1.0	April 2014
Consent Form for the use of quotes	1.0	April 2014
Interview Topic Guides	1.0	April 2014

If the fieldwork goes beyond the date stated in your application, you must notify the Ethical Review Panel via the ERP administrator at uso.erps@keele.ac.uk stating ERP2 in the subject line of the e-mail. If there are any other amendments to your study you must submit an 'application to amend study' form to the ERP administrator stating ERP2 in the subject line of the e-mail. This form is available via http://www.keele.ac.uk/researchsupport/researchethics/

If you have any queries, please do not hesitate to contact me via the ERP administrator on uso.erps@keele.ac.uk stating ERP2 in the subject line of the e-mail.

Yours sincerely

Dr Bernadette Bartlam
Chair – Ethical Review Panel

CC RI Manager
Supervisor

Research and Enterprise Services, Keele University, Staffordshire, ST5 5BG, UK
Telephone: + 44 (0)1782 734466 Fax: + 44 (0)1782 733740