

THE SEARCH PHASE OF SOFTWARE ENGINEERING SYSTEMATIC LITERATURE
REVIEW: BARRIERS AND SOLUTIONS

by

AHMED AL-ZUBIDY

JEFFREY C. CARVER, COMMITTEE CHAIR
SUSAN VRBSKY
JEFF GRAY
RANDY SMITH
DAVID HALE

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2017

ABSTRACT

The Systematic Literature Review (SLR) is an approach for conducting literature reviews that provides less bias and more reliability than the more typical ad hoc approach. One of the first phases of the SLR process is to conduct a comprehensive search for current literature. Conducting this search on different sources of literature (i.e., digital libraries) is a manual and exhausting task that results in an SLR process that is more costly than necessary.

Therefore, the goals of this dissertation are to: (1) find empirical evidence about the SLR search problem and the current status of tool support; (2) understand the barriers and the tooling requirements to support the SLR search phase; and (3) develop and evaluate a solution to help reduce the barriers. To address these goals, this dissertation consist of three articles.

Article 1 describes the results from three empirical studies that produce a list of tooling requirements across the entire SLR process. The results of these studies identify numerous gaps between needs of SLR authors during the search phase and the current tool support.

Article 2 consists of an SLR and a survey, to identify the specific tool requirements to support the SLR search phase. The SLR produced a list of SLR search barriers that were reported in the literature. The survey of SLR authors confirmed the results of the SLR and expanded the list to include issues and requirements from the community beyond what is reported in the literature.

Article 3 describes the development and evaluation of the Searching Systematic Reviews Tool (SSRT) tool to address the problems identified in Articles 1 and 2. SSRT provides one interface to search multiple digital libraries at once, store the results, and remove duplicates. The article also describes the evaluation of SSRT and the future extensions of SSRT.

No dedication body text.

ACKNOWLEDGMENTS

I would like to thank everyone who supported me throughout this journey. I would like to specifically thank Dr. Jeffrey Carver for his supervision of the project, his guidance and support. I would also like to thank my committee members for their support and feedback Dr. Jeff Gray, Dr. Susan Vrbsky, Dr. David Hale, and Dr. Randy Smith. Special thanks to the Software Engineering Group at the University of Alabama for their feedback and support. Finally, I would like to acknowledge the support from the United States National Science Foundation that supported this project under grants NSF-1305395 and that supported me during my study and research under grant NSF-1525373.

CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iii
LIST OF TABLES	ix
LIST OF FIGURES	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Related Work	2
1.1.1 Systematic Literature Review	2
1.1.2 SLR Barriers	3
1.1.3 SLR Search Process	4
1.2 Research Objectives	5
1.3 Plan of Action	6
1.3.1 Article 1: Develop A Vision for SLR Tooling Infrastructure	6
1.3.2 Article 2: Identification of SLR Search Requirements	7
1.3.3 Article 3: Development and Evaluation of an SLR Search Support Tool	7
1.4 Contributions	8
CHAPTER 2 VISION FOR SLR TOOLING INFRASTRUCTURE: PRIORITIZING VALUE-ADDED FEATURES	10
2.1 Introduction	10
2.2 Requirements Sources	12
2.2.1 S1: Initial Identification of SLR Barriers	12

2.2.2	S2: Community Workshop on SLR Barriers	14
2.2.3	S3: Community Workshop on SLR Tool Needs	16
2.3	Functional Requirements	19
2.3.1	Mapping S1 Barriers to S2 Barriers	19
2.3.2	Mapping S2 Barriers to S3 Requirements	21
2.3.3	Prioritized List of Requirements	24
2.4	Current Tool Support	28
2.4.1	Tool Analysis Methodology	28
2.4.2	Tools Analyzed	29
2.4.3	Tool Analysis Results	30
2.5	Discussion	31
2.5.1	Observations	31
2.5.2	Suggestion for Infrastructure Developers	35
2.6	Threats to Validity	36
2.7	Summary and Future Work	37
2.8	Acknowledgments	38
CHAPTER 3 IDENTIFICATION OF REQUIREMENTS FOR TOOL SUPPORT FOR SLR SEARCH: SLR AND A SURVEY OF SLR RESEARCHERS		39
3.1	Introduction	39
3.2	Research Questions	40
3.2.1	Barriers in the SLR Search Phase	41
3.2.2	Current SLR Search Practice	42
3.2.3	Current and Desired Tools for SLR Search	43
3.2.4	Validation of Barriers in Literature	44
3.3	Methodology: SLR	44

3.3.1	Sources and Searching Strategy	45
3.3.2	Inclusion and Exclusion Criteria	46
3.3.3	Paper Selection	47
3.3.4	Quality Assessment	48
3.3.5	Data Extraction and Synthesis	48
3.4	Results: SLR	49
3.4.1	RQ1: What barriers do SLR authors face during the SLR search phase?	50
3.4.2	RQ2: What solutions do researchers employ to overcome the barriers described in RQ1?	52
3.5	Methodology: Survey	59
3.5.1	Survey Design	59
3.5.2	Pilot Tests	61
3.5.3	Participants Selection	61
3.5.4	Data Collection	63
3.5.5	Data Analysis	63
3.5.6	Demographics	64
3.6	Results: Survey	66
3.6.1	RQ1: What barriers do SLR authors face during the SLR search phase?	66
3.6.2	RQ2: What solutions do researchers employ to overcome the barriers described in RQ1?	66
3.6.3	RQ3: What are the most effective search strategies employed by researchers during the SLR search phase?	68
3.6.4	RQ4: Which tools do SE researchers use in the SLR search phase? What are the advantages and disadvantages of these tools?	68
3.6.5	RQ5: What functional requirements do SLR researchers desire in tools specifically in the SLR search phase?	68

3.6.6	RQ6: Which of the results from RQ1 match the barriers we found in the results of our SLR?	70
3.7	Discussion	74
3.7.1	Problems are Solvable	75
3.7.2	Existing Solutions and Workarounds	77
3.7.3	Tool Support is Still Needed	78
3.8	Threats to Validity	79
3.8.1	SLR	79
3.8.2	Survey	80
3.9	Conclusion and Future Work	81
3.10	Acknowledgment	82
CHAPTER 4 SSRT - A TOOL FOR SYSTEMATIC REVIEWS SEARCHING PHASE		83
4.1	Introduction	83
4.2	Background and Previous Work	84
4.2.1	SLR Search Phase Barriers and Tool Support	84
4.2.2	Our Previous Motivating Work	85
4.3	SSRT Tool	86
4.3.1	Query Converter (QC)	89
4.3.2	Query Search (QS)	90
4.3.3	Downloading Results (DR)	92
4.3.4	Back-end Storage (BS)	94
4.3.5	Removing Duplicates (RD)	95
4.3.6	Import/Export (IE)	96
4.3.7	Archive (AR)	98
4.3.8	Citations Management (CM)	99

4.4	Evaluation	100
4.4.1	Feature Analysis	100
4.4.2	Replicating an SLR	101
4.4.3	Evaluation by Other Users	107
4.5	Threats to Validity	108
4.6	Conclusion and Future Work	109
CHAPTER 5 DISSERTATION CONCLUSION AND PUBLICATIONS		110
REFERENCES		112
APPENDIX A INCLUDED PAPERS IN THE SLR		119
APPENDIX B TOOLS USED IN SLR SEARCH PHASE		137

LIST OF TABLES

2.1	The Most Costly SLR Phases	14
2.2	S2: Results - Top Ten Composite Barriers	15
2.3	S3: Results - Top Ten Composite Requirements (Note: a tie for tenth place)	18
2.4	Matching of Barriers (S2) to Requirements (S3)	23
2.5	Composites Re-Ranking	25
2.6	Detailed Requirements	25
2.7	Tools Analysis Example	30
2.8	Tools Analysis Results	32
3.1	Search and Selection Stage Requirements	42
3.2	Search Strings	45
3.3	Included Papers Distributions	48
3.4	Data Extraction Form - Specific to Research Questions	49
3.5	Problem Source: Study Types	51
3.6	Problem Source: Infrastructure vs. People	52
3.7	Problem Source: Author's Relationships to Problem	53
3.9	SLR Search Barriers in SE	54
3.9	SLR Search Barriers in SE	55
3.9	SLR Search Barriers in SE	56

3.9	SLR Search Barriers in SE	57
3.9	SLR Search Barriers in SE	58
3.9	SLR Search Barriers in SE	59
3.8	Reported Solutions	60
3.10	SLR Survey Questions	62
3.11	SLR Search Problems Identified by Researchers	67
3.12	SLR Search Solutions Identified by Researchers	69
3.13	Most Effective Search Strategies Identified by Researchers	70
3.14	Tools Used in SLR Search Phase Identified by Researchers	71
3.15	5 Highest Tools Used in SLR Search Phase Identified by Researchers	72
3.16	SLR Search Phase Tool Requirements Identified by Researchers	73
3.17	Top Ten SLR Search Problems Identified by the Literature	74
4.1	Top Ten Requirements	87
4.2	QC Addressed Requirements	90
4.3	QS Addressed Requirements	92
4.4	DR Addressed Requirements	92
4.5	BS Addressed Requirements	95
4.6	RD Addressed Requirements	95
4.7	IE Addressed Requirements	97
4.8	AR Addressed Requirements	98
4.9	CM Addressed Requirements	99
4.10	SSRT Feature Analysis Score	102
4.11	SLR Tools Feature Analysis Scores	103

4.12 Search Strings	105
A.1 List of papers included in the SLR	119
A.1 List of papers included in the SLR	120
A.1 List of papers included in the SLR	121
A.1 List of papers included in the SLR	122
A.1 List of papers included in the SLR	123
A.1 List of papers included in the SLR	124
A.1 List of papers included in the SLR	125
A.1 List of papers included in the SLR	126
A.1 List of papers included in the SLR	127
A.1 List of papers included in the SLR	128
A.1 List of papers included in the SLR	129
A.1 List of papers included in the SLR	130
A.1 List of papers included in the SLR	131
A.1 List of papers included in the SLR	132
A.1 List of papers included in the SLR	133
A.1 List of papers included in the SLR	134
A.1 List of papers included in the SLR	135
A.1 List of papers included in the SLR	136
B.1 Spreadsheet(Excel) - Purpose	137
B.2 Spreadsheet(Excel) - Limitations	138
B.3 Digital Libraries and Search Engines - Purpose	138
B.4 Digital Libraries and Search Engines - Limitations	138

B.5 Jabref - Purpose	138
B.6 Jabref - Limitations	139
B.7 Mendeley - Purpose	139
B.8 Mendeley - Limitations	139
B.9 Custom tool - Purpose	139
B.10 Custom tool - Limitations	139
B.11 StArt - Purpose	140
B.12 StArt - Limitations	140
B.13 End Note - Purpose	140
B.14 End Note - Limitations	141
B.15 RefWorks - Purpose	141
B.16 RefWorks - Limitations	141
B.17 RefWorks - Purpose	141
B.18 RefWorks - Limitations	141
B.19 RefWorks - Purpose	142
B.20 REviewER - Limitations	142
B.21 Zotero - Purpose	142
B.22 Zotero - Limitations	142
B.23 Sharelatex - Purpose	143
B.24 Google drive - Purpose	143
B.25 Google drive - Limitations	143
B.26 Publish or perish - Purpose	144
B.27 Reseachr - Limitations	144

B.28 SESRA - Purpose	144
B.29 SESRA - Limitations	144
B.30 NVivo - Purpose	145
B.31 GSResearch - Purpose	145
B.32 SLuRP - Purpose	145
B.33 SLuRP - Limitations	145

LIST OF FIGURES

1.1	SLR Process	2
1.2	Plan of Action Flow	9
2.1	Matching of S1 and S2 Barriers	22
3.1	Number of SLRs Each Participants Worked On	65
3.2	Participants Status and the Number of SLRs Worked On	65
3.3	Problems in Literature Matched to Problems in Survey	75
4.1	SSRT Main Screen	89
4.2	SSRT Design	89
4.3	Query Converter Home Screen	91
4.4	Results of Query Search	93
4.5	Download Options	94
4.6	Database Design	96
4.7	Export Options	97
4.8	Archive Home Screen	98
4.9	Citations Management Home Screen	99

CHAPTER 1

INTRODUCTION

The starting point of any research endeavor is identifying what work has already been done and identifies the gaps. No one should begin new research without first reviewing previous related work. Gaining this insight from the literature should be a structured process that produces reliable answers. To increase the reliability of the results, researchers began demanding evidence when answering research questions. In the 1970s the evidence-based movement began in the medical field [9]. Then in 1992, Sackett defined the systematic literature review (SLR) process in medicine [49]. This process is the one currently followed in the medical field.

The SLR process follows strict rules and guidelines. It is driven by the definition of specific, detailed research questions rather than broad questions. These questions are answered based on evidence published in the literature [35]. Researchers must identify and review all related information to answer the research questions in a systematic way. The rules and guidelines of the process reduce bias while conducting a systematic review [5, 31, 35].

More recently, in 2004, software engineering (SE) researchers have begun using the SLR approach. The increase in evidence-based SE studies suggests that such a process would be applicable to follow while conducting a literature review [31]. Since its introduction, the SLR approach has been widely accepted and adopted by members of the SE community. There has been an increasing number of published SLRs. A report in 2013 identified more than 116 published SLRs in SE venues [28].

Even with the increased adaptation of the SLR process for the SE community, it still



Figure 1.1: SLR Process

suffers from a number of barriers [24, 25, 32]. While those barriers occur in multiple phases of the SLR process, most of them are related to the search phase. Even though there are multiple tools that claim to support the entire SLR process [4, 6, 20, 27], the search phase is still largely unsupported by automated tools [42]. Those barriers lead to increase effort for conducting SLRs [24, 25, 41].

Therefore, the primary goal of this dissertation is to **understand and document the barriers related to the SLR search phase and build and evaluate a solution to overcome those barriers.**

1.1 Related Work

This section provides information on the SLR process, the barriers the SLR process faces, and the search phase barriers.

1.1.1 Systematic Literature Review

Based on Kitchenham’s guidelines for conducting SLRs in SE [35], the process of SLR goes into multiple stages. As shown in Figure 1.1 above, it is broken down into three major phases: Planning, Execution, and Documentation, with sub-phases in each major phase.

- Planning Phase: In this phase, a group of researchers would define the SLR protocol

that guides them to the completion of the SLR. The researchers build the protocol in iterative approach and should be reviewed by an external researcher to ensure its validity. For example, the protocol would include information on

- The motivation of conducting the SLR.
 - Research questions.
 - Search strategy and search sources.
 - Inclusion/exclusion criteria.
 - Quality assessment checklist.
 - Data extraction forms.
- Execution Phase: The researchers follow the protocol and the plan they defined in the previous phase. Team members conduct the tasks individually then compare their work together and discuss any conflicts in their results. Some of the tasks in this phase are:
 - Identifying relevant research based on their defined search strategy.
 - Selecting primary studies to be included in the SLR.
 - Evaluate the primary studies.
 - Extract the required information to answer the research questions.
 - Synthesize their findings.
- Documentation Phase: The final phase of the process is documenting and writing the report. The researchers work together to report their results and format the SLR document.

1.1.2 SLR Barriers

There are a number of barriers in the SLR process. A recent mapping study on SLR in SE identified a large number of barriers in multiple stages in the SLR process [32]. Some

barriers are related to the unstructured nature of the literature and to the SLR methodology itself. However, most of the barriers are related to the lack of tool support for the SLR process in SE. The searching phase is especially problematic. A survey of SLR authors to identify the barriers they face in the SLR process showed the three most time consuming phases of the SLR process are: searching of the digital libraries, study selection, and data extraction. These phases were also the ones most in need of tool support [8].

Hassler et al. conducted a community workshop where SLR authors identified and ranked barriers in the SLR process. The literature searching process was the top barriers that emerged in the workshop [25]. In a follow-up workshop, SLR authors identified and prioritized the functional requirements needed in an SLR tool support. The most required features were again related to the SLR searching phase [24].

A 2007 study [12] described the differences between the medical field and the SE field related to SLR. One of the differences identified in the paper is how medical researchers and SE researchers look for papers. As the medical field is more mature and has been using the SLR process for a long time, they have developed more infrastructure to support SLRs. However, there are not yet any resources dedicated to the SLR process in SE, for example a centralized information source for evidence.

1.1.3 SLR Search Process

Very few studies have been dedicated specifically to the SLR searching problem in SE. However, the studies that did examine the search phase, reported more problems, including: issues with the digital libraries, no standardized way to search all the resources, limitations on query building and inconsistency, irrelevant results, and the inability to obtain the results automatically [14, 34, 57].

The current SLR support tools for SE lack the ability to conduct automated searches. There are some studies that have identified tools in the SLR process of SE and evaluated those tools. The evaluations showed that none were able to conduct automated search, some did not even support search, and those that did support search require researchers to perform

the search manually and import their findings in the tool[41, 42].

There are two studies that proposed automated SLR tools. Molleri proposed an automated approach to support the entire SLR process [44]; however, it is only a proposal and does not address the searching phase. Kamdar also proposed an automated tool that helps in the searching phase of the SLR [30]; however, it is a very short proposal published recently in 2015. Therefore, there has not been any actual work or details on how to address the problem with a concrete solution.

The need for automated tool support for the entire SLR process is essential. However, previous work has shown that the search phase is one of the most problematic. Therefore, in this dissertation I focus on the search problem to understand it, evaluate it, design and implement a solution, and evaluate the solution to reduce the barriers. Reducing the barriers to the SLR search will reduce the effort needed to conduct SLRs, increase the number of published SLRs in SE, and increase the quality of those SLRs.

1.2 Research Objectives

First, the SLR process is very important for producing high quality research. As described previously, searching is one of the most important barriers that stands in the way of conducting SLRs. One of the highest barriers is the manual work to search and obtain the documents from digital libraries. Therefore, the first objective of this dissertation is to *identify empirical evidence to confirm the existence of the searching problems and the current status of tool support in the SLR process in SE (RO1)*.

Second, very few studies have focused on the barriers and the requirements in the SLR search phase. All of the barriers in the SLR search phase result in a large amount of manual work to complete this phase. The increased cost resulting from manual work make the searching phase important to study. Therefore, the second objective of this dissertation is to *understand the barriers and tool requirements for the SLR search phase (RO2)*.

Finally, based on the outcomes of RO1 and RO2, we can propose a solution to reduce

barriers in the search phase. Therefore, the third objective of this dissertation is to *build and evaluate a solution to reduce the barriers in the SLR search phase (RO3)*.

The primary goal of this dissertation is to understand the barriers that are related to the searching phase of the SLR process. Since this phase is still a manual work that takes a lot of time and effort, and the current tools still lack the support of automated search for literature. This dissertation's secondary goal is studying the possibility of solution that automate parts of the searching phase and an evaluation of the solution affect on the SLR process in SE.

1.3 Plan of Action

This dissertation consists of three articles that build upon each other.

1.3.1 Article 1: Develop A Vision for SLR Tooling Infrastructure

Article 1 describes the results of three empirical studies focused on identifying SLR process barriers and tool requirements throughout the SLR process. This article addresses the following goals:

- To develop and document online a prioritized list of functional requirements for SLR tools based on input from SLR authors within the SE community that can serve as a reference for SLR tool developers who wish to build or enhance SLR tools.
- To perform and document online our own analysis of how the current SLR tools address these requirements and have that analysis validated by the tool authors.

I plan to keep both online lists up to date as community needs change and as tool authors update existing tools or develop new tools.

The findings of this paper showed that the authors still face a lot of problems during the SLR search phase. Current tools do not provide much (if any) support specifically for the SLR search phase. Therefore, there is a need for more work to address the barriers faced by authors during the SLR search phase and to reduce the overall cost of conducting SLRs.

1.3.2 Article 2: Identification of SLR Search Requirements

Article 2 provides empirical evidence about barriers in the SLR search phase and requirements for tools to reduce those barriers. Because there are not a lot of studies that focus specifically on the SLR search phase, among the most problematic phases for SLR authors, it is important to understand the SLR search phase in more detail. This article consists of two studies, an SLR on the SLR search phase to identify barriers reported in the literature and a survey of SLR authors to identify barriers faced in practice. The combination of these results produced a list of tool requirements for the SLR search phase. The study achieved the following goals:

- Understand the SLR search barriers from the literature and researcher’s point of view.
- Provide a list of SLR search barriers while conducting SLR.
- Encourage SLR authors, researchers, and developers to provide solutions to reduce the barriers in the SLR search phase.

This study resulted in a list of tool requirements for the SLR search phase based on the literature and on the survey of 131 SLR authors. The study also describes the tools used in the SLR search phase, along with their advantages and disadvantages. This information helps researchers and tool developers understand how they can take advantage of existing tools, where appropriate, and identify points of potential collaboration and integration.

1.3.3 Article 3: Development and Evaluation of an SLR Search Support Tool

Article 3 describes the construction and evaluation of a tool to reduce the barriers identified in Articles 1 and 2. The SLR search barriers are primarily caused by limitations in the digital libraries, inadequate search interfaces, poor automated tool support, and the lack of a standardized way to search multiple sources. All those barriers can be addressed through appropriate tool support. This study achieved the following goals:

- Build a tool to address the SLR search phase barriers.

- Evaluate the proposed solution and measure its ability to reduce the SLR search phase barriers.
- Improve the SLR process and reduce the effort needed for the SLR search phase.

The results of this article is the Searching Systematic Reviews Tool (SSRT), a tool to support the SLR search phase. The tool enables researchers to use one interface to search multiple databases using standardized queries, download the results in batch mode, and automatically eliminate duplicates. Authors must currently perform these tasks manually, resulting in a lot of expended effort. The article also provides an informal evaluation of the tool that we have conducted and it shows the tool does reduce the manual work that is required in the SLR search phase.

Figure 1.2 illustrates how the articles fit together to address the research objectives.

1.4 Contributions

The main contributions of this dissertations are:

- Provides empirical evidence about the existence of and the importance of problems in the SLR search phase.
- A list of requirements to be used as a reference for SLR authors, researchers, and developers who want to build tools.
- Enables SLR authors to be more efficient by automating some of the manual work required during the search phase.
- Improves the quality of SLR by improving the quality of the search process through automation.

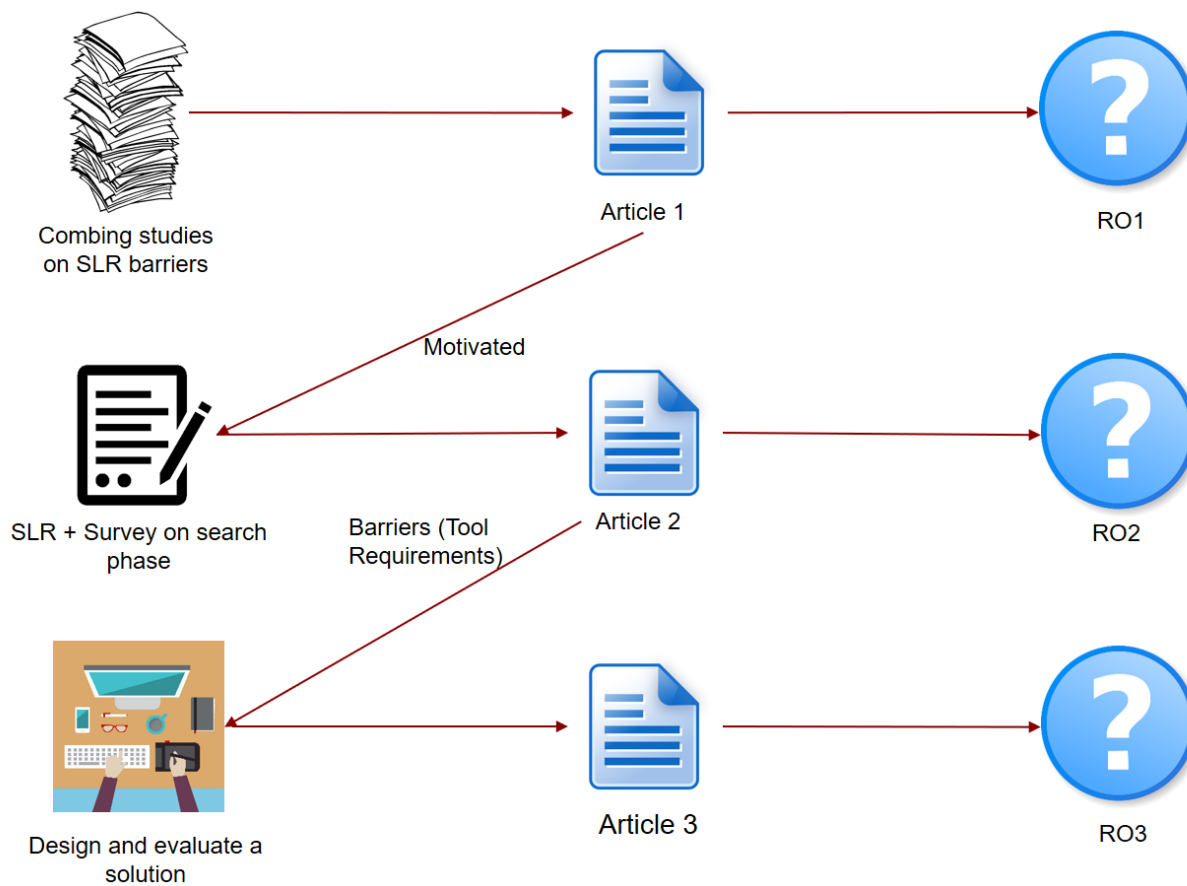


Figure 1.2: Plan of Action Flow

CHAPTER 2

VISION FOR SLR TOOLING INFRASTRUCTURE: PRIORITIZING VALUE-ADDED FEATURES

2.1 Introduction

Systematic literature review (SLR) is a formal, structured process to conduct literature reviews. SLRs are goal-driven activities that result in unbiased, repeatable, comprehensive literature reviews that evaluate and synthesize results of multiple primary studies [35]. Medical researchers have long depended on SLRs to synthesize study results and draw conclusions about medical interventions. Similarly, the prevalence of empirical research in software engineering (SE) has led to the same benefits. Increasingly, SE researchers are performing SLRs [31, 38]. There are more than 174 published SLRs in software engineering as have been reported in the literature [11, 28, 33, 37].

Unfortunately, the benefits of SLRs require a significant investment of time and effort [1, 8, 25, 48]. In addition, the SLR process has a number of barriers that limit the extent to which it can be used more broadly by the SE community. Many of these barriers span multiple SLR phases or results from inadequate tool support. Examples of barriers reported in the literature include: (1) difficulty planning search strings to accommodate the requirements of each digital library [35, 45, 47]; (2) difficulty identifying relevant papers and constructing search strings during the search and selection phases [7, 33]; (3) difficulty extracting data from papers due to inconsistent formatting [7]; (4) SE digital libraries and infrastructure do not adequately support SLRs compared with the medical field [12, 15, 47]; and (5) the overall amount of manual work required to complete the SLR process [7, 12, 33, 35]

Many of these barriers result from the fact that much of the SLR process requires time-consuming, manual tasks. Thus, proper tool support can reduce or eliminate many of these barriers and increase the number and quality of published SLRs. Previous studies have identified and examined various aspects of eleven existing SLR tools [24, 33, 41]. Of these, only four support the entire SLR process, StArt [27], SLuRP [6], SLR-Tool [20], and SLRTOOL [4]. The remainder of these tools mainly assist with text mining, which is useful in the selection phase of the SLR process. A recent evaluation of these tools against a set of features derived from the SLR process structure by Marsahall et al. [42] found that they still lack a number of important features, especially related to automation of the search process.

The goal of the research results represented in this paper was *to work with the SLR community to identify and prioritize a set of SLR tool requirements to address the existing barriers*. To that end, we have conducted a sequence of three studies of the SE SLR community to elicit SLR barriers and the consequent SLR requirements.

- **S1:** A survey of SLR researchers to identify the barriers faced when performing SLRs (especially by novice researchers) [8].
- **S2:** A community workshop to identify and prioritize the most important barriers in the SLR process [25].
- **S3:** A community workshop to identify and prioritize the most important tool features that would aid in the conduct of SLRs in SE [24].

Section 2.2 provides a brief overview of each study along with the results that are important to the overall goal of the work. The details of each study can be found in their respective publications.

The primary differences between our current study and the previous study by Marshall et al. [42] are (1) we include tools not evaluated by Marshall et al. (Parsifal [21] and SESRA [43]), (2) we include an updated version of StArt [18] in our evaluation that was not released

in time to be evaluated by Marshall et al., and (3) we analyze the tools against a different set of criteria, one generated based on input from the SLR community.

The primary contributions of this work are:

- A prioritized list of functional requirements for SLR tools based on input from SLR authors within the SE community. This list, which we will maintain online, will serve as a reference for SLR tool developers who wish to build or enhance SLR tools.
- An analysis of how the current SLR tools cover these requirements. We will continually update the results of this analysis online as developers create or evolve SLR support tools.

The remainder of this work is structured as follow. Section 2.2 gives an overview of the three studies used in this analysis. Section 2.3 reports the functional requirements. Section 2.4 reports the analysis of how the current SLR tools cover the requirements. Section 2.5 discusses our observations of the results. Section 2.6 describes the limitations of this study. Section 2.7 provides the summary and plans for future work.

2.2 Requirements Sources

We draw upon the results from three prior studies to identify the requirements for the envisioned SLR tooling infrastructure. This section provides an overview of each of those studies. For each study, the section describes: the goals, the participants, the process, the results, and the contribution towards the overall tool vision. The full details of each study have already been published elsewhere [8, 24, 25].

2.2.1 S1: Initial Identification of SLR Barriers

This study focused on identifying the most difficult and time-consuming phases of the SLR process. The full results appeared in the 7th *International Symposium on Empirical Software Engineering and Measurement* [8].

The Study Goal: The goal of this study was *to use experiences (both ours and members of the SLR community) to identify barriers that researchers, especially novice researchers, face when performing an SLR*. This study provided a starting point for our work on the barriers in the SLR methodology. We sought to identify both the type of barriers and the SLR phases in which they occur.

The Participants: This study contained two types of participants: 1) Eight PhD students who attended an Advanced Empirical Software Engineering course and were conducting an SLR as part of that course, and 2) Fifty-two respondents to a survey sent to authors who had published an SLR in a SE venue.

The Process: We asked the graduate students and the survey participants to describe the SLR process they followed, the difficulties they faced, the time they spent during the process, and the phases of the SLR that most needed tool support. We collected data differently from each participant type.

- PhD Students – wrote a report at the conclusion of their SLR describing their experiences, including the barriers faced while conducting the SLR.
- Survey respondents – completed an online survey to describe their SLR process in terms of: barriers faced, time spent, and phases most in need of tool support.

The Results: This study identified barriers in the SLR process in SE, especially those that are faced by novice researchers. The first result is the identification of the SLR phases that are most difficult, most time consuming, and most in need of tool support (Table 2.1). The other key results are:

- The paper identification process is mostly manual and requires a large amount of effort.
- Collaborative SLRs are not well-supported by tools.
- SLRs cannot evolve over time as a "living" document.

Table 2.1: The Most Costly SLR Phases

	Paper Selection	Data Extraction	Quality Assurance	Database Searching
Difficult	1	2	3	
Time Consuming*	2	1	3	3
Tool Support	2	1		3

*Note that there was a tie for the third most time consuming phase

The Contribution: This study identified the general barriers faced by SLR researchers in SE. Many of the barriers result from the lack of tool support, which causes authors to spend more effort conducting an SLR. Other barriers result from non-conformance to the SLR process. However, even these non-conformance issues can result from the lack of tool support. The outcome of this study motivated the need to understand the SLR process barriers in more detail.

2.2.2 S2: Community Workshop on SLR Barriers

This study focused on identifying and ranking the barriers faced by SLR authors in SE. The full results appeared in the *18th International Conference on Evaluation and Assessment in Software Engineering* [25].

The Study Goal: The results of S1 provided some high-level insight into the level of difficulty and effort required for the SLR phases. To provide a more detailed understanding of these barriers and an idea of their relative importance, we needed to conduct a more detailed study. The goal of this study was to *identify the barriers deemed most important by the members of the SE research community who conduct SLRs.*

The Participants: We conducted a workshop during the *2013 ACM/IEEE International Symposium on Empirical Software Engineering*. The workshop participants had all conducted one or more SLRs, including eleven faculty members and two PhD students. Other than one faculty member who was a PhD./MD. researcher in a medical school, all participants were from a computer science or software engineering department.

Table 2.2: S2: Results - Top Ten Composite Barriers

No.	Composite Barrier
1	Lack of Tool Support
2	Assessing Articles Quality
3	Synthesizing Primary Studies
4	Handling Data
5	Inadequate Search Engines
6	Connections with Industry Needs
7	Keywords and Terminologies
8	Reliability Measurements
9	Differences of SLR and Mapping Studies
10	Process Efficiency Assurance

The Process: We used the Nominal Group Technique [13] to identify and prioritize the SLR barriers. First, participants individually brainstormed the barriers they faced in the SLR process. The participants recorded each barrier on a separate sticky note. Next, the participants announced their barriers to the group and affixed them to a poster board corresponding to the SLR phase in which they occurred. Then, participants analyzed the barriers assigned to each phase to consolidate similar barriers and identify a set of composite barriers. Finally, the participants voted on the importance of each composite barrier.

The Results: The participants identified 100 barriers and grouped them into 37 composite barriers. The barriers related to search and selection were the highest rated barriers. Table 2.2 shows the top ten composite barriers. The table is sorted by the highest voted composite barriers.

After the workshop was completed, we performed a secondary qualitative analysis of the composite barriers. That analysis resulted in identification of four key themes among the barriers.

- SLR Process Barriers
 - The SLR protocol prescribes a sequential process rather than an iterative process
 - Meta-analysis is difficult
 - Researchers lack methods for result interpretation, generalization and framing

- Primary Study Barriers
 - Titles and abstracts are often misleading or irrelevant
 - Terminology and its use are not standardized
- Practitioner Community Barriers
 - Researchers have difficulty relating to industry needs
 - Researchers have difficulty conveying the value of a structured process
- Tooling Barriers
 - Electronic databases lack adequate search and retrieval facilities
 - Tools to support data extraction and management are inadequate or nonexistent

The Contribution: This study builds on the results of S1 by providing more detailed insight into the barriers based on the face-to-face interactions of the workshop participants. The workshop participants identified a large list of detailed barriers, which they grouped into higher-level composite barriers. These results show that the barriers occur across the entire SLR process, indicating a great need for tool support. Additionally, this study provided a rank ordering of the composite barriers that indicates the relative importance of each barrier. The outcome of this study motivated the need to understand the types of tool features or requirements necessary to address these barriers, which led to S3.

2.2.3 S3: Community Workshop on SLR Tool Needs

This study sought to expand on the previous studies to specifically identify the tool infrastructure that community members desired. The full results appeared in *Information and Software Technology* [24]. The group of participants were different in each of these studies. Thus, in this work we match these studies to each other to: A) Validate the needs covers the barriers, and B) Based on the three studies, provide a comprehensive list of requirements that SE researchers needs in SLR tool support.

The Goal of the Study: The results of S1 and S2 provided valuable insight into the barriers faced by SLR authors. To complement that information, we needed to better understand what gaps in tooling SLR authors would find most beneficial to close. Thus, the goal of this study was to *identify and prioritize tool features that would be beneficial when conducting an SLR in SE*.

The Participants: This workshop took place during the *18th International Conference on Evaluation and Assessment in Software Engineering*. The sixteen workshop participants included ten experience SLR authors (i.e. had completed at least three SLRs) and six novice SLR authors (i.e. had completed one or two SLRs). All attendees were interested in improving the SLR process by upgrading the available tool support infrastructure.

The Process: We again used the Nominal Group Technique [13] to identify and prioritize the SLR tool requirements. First, participants individually brainstormed ideas for tool requirements and recorded them on individual sticky notes. These requirements included new features, features that exist but need improvement, or features that exist and are well done. Then, as a group, the participants grouped the requirements into composite requirements around the different phases of the SLR process. Finally, the participants voted on which requirements were the most important.

The Results: The participants generated 97 primary requirements and grouped them into 28 composite requirements. Table 2.3 shows the top ten composite requirements, sorted by number of votes.

Analyzing the information from the workshop, the primary results of the workshop were:

- The highest ranked tool feature is integrated search, which enable the researchers to search multiple databases without having to perform separate searches.
- The highest ranked tool features cross SLR phases or focus on Search and Selection.
- The concept of collaboration appears in four tool features across four SLR phases, making collaboration the most desired feature.

Table 2.3: S3: Results - Top Ten Composite Requirements (Note: a tie for tenth place)

No.	Composite Requirement	Abbreviation (for Table 2.4)
1	Integrated Search	Int. Search
2	Collaboration Support	Collb. Support
3	Traceability	Traceability
4	Support Inclusion and Exclusion	Support Inc. & Exc.
5	Data Maintenance	Data Maint.
6	Automated Analysis	Auto. Analysis
7	Support Quality Assessment	Support Qual. Asmt.
8	Support Text Mining	Support Text Mining
9	Visualization	Viz.
10	Coding of Methods and Data	Coding Methods & Data
10	Storage of Studies	Study Storage

Comparing the results from the novices and the experts, we found:

- no overlap in top ranked tool features between the novice and expert SLR authors.
- expert researchers gave a much higher priority to Collaboration than novice researchers.
- novice researchers found task execution features (i.e., analyzing, tracing, time estimating, and collaboration on data extraction) to be of higher priority than expert researchers.
- expert researchers found tactical features (i.e., collaboration, protocol development and validation, and sharing) to be of higher priority than novice researchers.

Finally, evaluation of the current tool support indicated that:

- The two highest ranked tool features, Integrated Search and Collaboration, are not well supported by current tools.
- The third highest ranked tool feature, Traceability, is partially supported in current tools.
- The remaining five highest ranked tool features were either missing or needed improvement, with the exception of Data Maintenance.

The Contribution: This study is complementary to the results of S1 and S2 in that it provides detailed insights into the types of tool features desired by workshop participants. Similar to S2, the workshop participants identified a large list of detailed tool requirements, which they grouped into higher-level composite requirements.

2.3 Functional Requirements

This section describes the process of deriving a set of detailed requirements for an SLR tooling infrastructure. First, it maps the barriers identified in S1 to those identified in S2. Then it maps that list of barriers to the requirements identified in S3. The goal of these mappings is to provide validation and traceability from the barriers to the requirements. Finally, we provide the community with a list of detailed requirements that SE researchers would like to see in future tools to reduce the cost of conducting SLRs.

2.3.1 Mapping S1 Barriers to S2 Barriers

When gathering subjective, qualitative data, it is important to triangulate the results using different sources. The goal of this mapping is to ensure that the results gathered in S1 (from the survey and PhD students) is consistent with the results from S2 (the in-person workshop). If the results of S1 can be mapped to the results of S2, we have more confidence that they describe real barriers, rather than indiscriminate statements.

By analyzing the S1 results, we generated a list of barriers faced by the SE community when conducting SLRs. We grouped these barriers into three main themes.

1. **Digital Libraries Issues** – including issues such as: coverage of digital libraries, duplicate of references, inconsistent data in digital libraries, indexing issues, retrieving and managing papers, limited access, and filtering capabilities.
2. **Quality Assessment Issues** – including issues such as: difficulty assessing quality, insufficient resources, difficulty defining scope, and reusing existing assessments by others.

3. **Issues caused by not following the SLR protocol** – including issues such as: deviating from protocol, iterating in the process until reaching satisfied set of results, incomplete protocol descriptions, difficulties in resolving discrepancies, and difficulties of measuring agreement between researchers.

From the S2 results, we ranked the composite barriers based on the number of times the survey respondents mentioned them, we extracted the ten most important composite barriers (based on voting results), as follows.

1. **Lack of Tool Support (across all SLR phases)** – including barriers such as: lack of support for data extraction and analysis, lack of tools to help in traceability, lack of tools to help manage the data and documents, and lack of integration with other tools.
2. **Assessing Articles Quality** – including barriers such as: abstracts lack relevant information, misleading titles, and poorly structured primary studies (note that even though it focuses mostly on the quality of the articles, tool support can still help).
3. **Synthesizing Primary Studies** – including barriers such as: difficulties of mixing evidence from multiple studies, difficulties for synthesizing evidence across studies, and the lack of metrics to compare the similarity of studies.
4. **Handling Data** – including barriers such as: difficulties of summarizing and aggregating data, especially qualitative data.
5. **Inadequate Search Engines** – including barriers such as: finding duplicate studies in the digital libraries, search facilities not well-suited for SLR, and search string incompatibility across digital libraries.
6. **Connecting with Industry** – including barriers such as: the difficulty of mapping results between industry and academia, and the difficulty of producing SLRs that are valuable to industry.

7. **Keywords and Terminologies** – including barriers such as: the lack of unified, consistent SE terminology for key concepts, which make it difficult to evaluate the quality of search strings.
8. **Reliability Measurements** – including barriers such as: the difficulty of validating and measuring the quality of the studies, lack of quality assessment criteria, and the difficulty to have consistent assessment.
9. **Differences of SLR and Mapping Studies** – including barriers such as: the differences in the protocols and selection activities between SLRs and Systematic Mapping Studies.
10. **Process Efficiency Assurance** – including barriers such as: difficulty of process validation, lack of training on the SLR process, and the lack of reliability measures of validity.

As described in Section 2.2, each composite barrier was the result of combining a number of detailed barriers. To perform the mapping between S1 and S2, we returned to the detailed barriers (rather than the composite barriers). For each detailed barrier mentioned in S2, we looked for a corresponding detailed barrier in S1. After performing this mapping, we aggregated the results at the composite barrier level. In other words, if a detailed barrier from composite barrier A in S2 mapped to a detailed barrier from composite barrier B in S1, then we determined that barrier A maps to barrier B.

The results of the mapping, displayed in Figure 2.1, shows that each of the top ten composite barriers from S2 mapped back to at least one composite barrier from S1. In most cases the composite barriers in S2 provided more insight and details regarding the composite barriers identified in S1.

2.3.2 Mapping S2 Barriers to S3 Requirements

To ensure that the requirements produced during S3 were correct, we mapped those requirements to the barriers from S2. The goal of the mapping was to determine whether

Survey (S1): Barriers Composites

Workshop 1 (S2): Barriers Composites

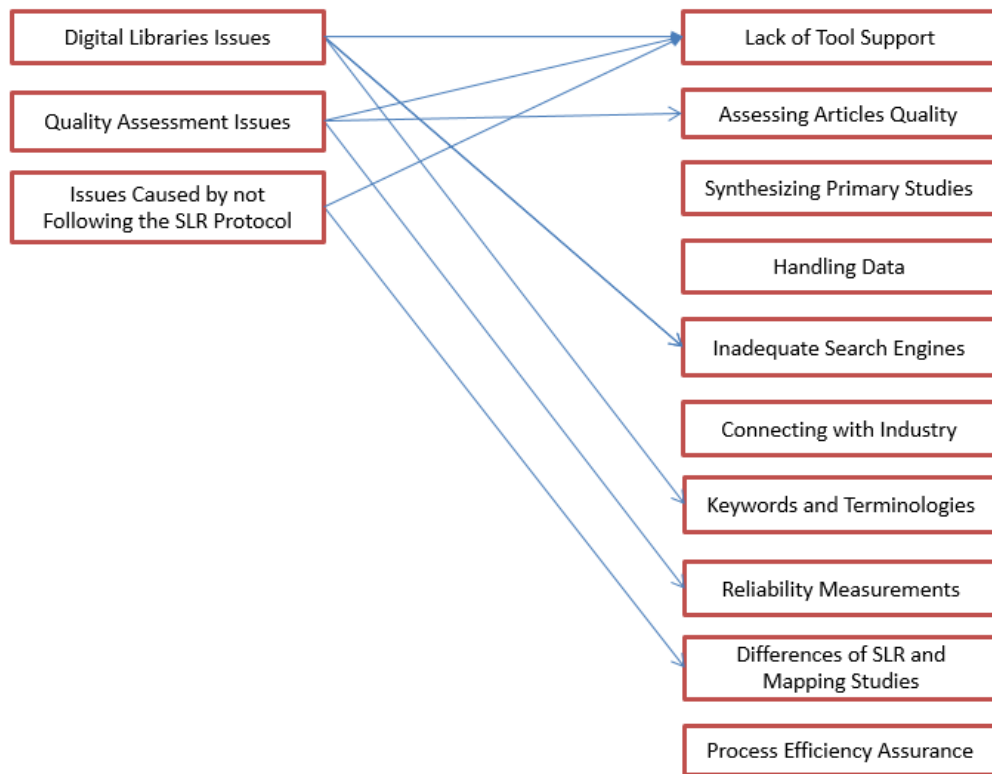


Figure 2.1: Matching of S1 and S2 Barriers

the S3 requirements were valid and covered all of the key barriers. Because the attendees of S2 and S3 were disjoint, if we can show congruence between the results, then we have additional evidence of the validity of the resulting set of requirements.

We performed this mapping similar to the mapping between S1 and S2. Each composite requirement consists of a number of detailed requirements. Using the ten highest rated requirements, we matched each detailed requirement with any detailed barriers that the requirement addressed. In other words, if a detailed requirement in composite requirement A matched a detailed barrier in composite barrier B, we said that composite requirement A addressed composite barrier B. Two researchers performed this matching independently and met to resolve any discrepancies. Table 2.4 displays the results.

Table 2.4: Matching of Barriers (S2) to Requirements (S3)

Reqs. Barriers	Int. Search	Collab. Support	Trace- ability	Support Inc. & Exc.	Data Maint.	Auto. Analys- is	Support Qual. Asmt.	Support Text Mining	Viz	Coding Method & Data	Study Storage
Lack of Tool Support	X	X	X	X	X	X	X	X	X	X	X
Assessing Article Quality		X					X	X			
Synthe- sizing Primary Studies		X	X		X	X		X	X	X	
Handling Data		X	X		X	X		X	X	X	
Inade- quate Search Engine	X	X		X				X			X
Connect- ing with Industry		X									
Keywords & Termino- logies	X	X		X			X	X			
Reliabi- lity Measure- ments		X	X				X				
Difference of SLR & Mapping Studies		X		X						X	
Process Efficiency Assurance		X	X				X				

2.3.3 Prioritized List of Requirements

The ranking in the previous section was based only on the votes during S3. Using information about the priority of the barriers (S2) and the requirements (S3), we reranked the top requirements from S3 taking into account the importance of the barriers the requirements addressed. We calculated the new ranking as follows:

- We assigned each composite barrier and each composite requirement a value from 1-10, based on its rank within its individual workshop (i.e, the top barrier and top requirement each received 10 points, the second barrier and requirement each received 9, and so on).
- In cases where a barrier was addressed by more than one requirement, we divided the barrier's points evenly among those requirements. This practice gave more importance to a requirement than if it was the only one that addressed a particular barrier. For example, if requirement 3 and 4 both addressed barrier 1, then requirement 3 and 4 each received half, or 5, of the 10 points belonging to barrier 1. In the case where only one requirement addressed a barrier, that requirement received all of the barrier's points.
- Finally, to get the final score for each requirement, we summed the points given to the requirement based on its ranking and the points given to the requirement based on the barriers it addressed.

For example, the *Data Maintenance* requirement is ranked as number 5 in S3 (giving it 5 points). It addresses barriers *Lack of Tool Support* ranked as number 1 in S2 (worth 10 points), *Synthesizing Primary Studies* ranked as number 3 in S2 (worth 8 points), and *Handling Data* ranked as number 4 in S2 (worth 7 points). *Lack of Tool Support* is addressed by 10 requirements, so its 10 points are divided among those ten requirements, leaving 1 point for *Data Maintenance*. Likewise, *Synthesizing Primary Studies* is addressed by 7 requirements, giving each one 1.1 points and *Handling Data* is addressed by 7 requirements,

Table 2.5: Composites Re-Ranking

Ranking	Requirement	Final Score
1	Collaboration Support	24
2	Integrated Search	13
3	Traceability	12.4
4	Support Text Mining	11.1
5	Support Inclusion and Exclusion	10.6
6	Support Quality Assessment	10.1
7	Data Maintenance	9.1
8	Automated Analysis	8.1
9	Visualization	5.1
10	Coding of Methods and Data	4.7
11	Storage of Studies	3.2

*Note that there was a tie for the tenth and eleventh most voted composite requirements in the workshop

giving each one 1 point. Therefore, the total score for *Data Maintenance* is its original 6 points plus the points from its barriers (1 + 1.1 + 1) totaling 9.1. After performing this computation for all requirements, *Data Maintenance* moved down to requirement number 7. Table 2.5 shows the ranking of the composite requirements based on this approach.

The results of the combination of the results from S1, S2 and S3 provide researchers and tool builders with a prioritized list of SLR tool requirements desired by the community. The list of requirements contains 112 requirements across all phases of the SLR process.

Table 2.6 provides a list of the top 11 composite requirements (along with the detailed requirements that comprise them). The full list of requirements is available on an online repository (<http://carver.cs.ua.edu/Studies/SLR-Requirements/>). We will maintain this repository as future research is done in this area.

Table 2.6: Detailed Requirements

#	Consolidated Requirements	Requirement
1	Collaboration Support	Support for distributed review teams
		Support for collaborative SLR

#	Consolidated Requirements	Requirement
		Support for conflict resolution
2	Integrated Search	<p>Search and storage of data other than PDFs</p> <p>Portal to search stored literature</p> <p>Integration with databases services</p> <p>Automated search in digital libraries</p> <p>Support for collaborative search</p> <p>Automated search and storage of studies in digital libraries</p> <p>Importing search results from different databases</p> <p>Unified portal to search literature in common digital libraries</p> <p>Cross search of digital libraries</p> <p>Support for search iterations</p> <p>Support piloting search string</p>
3	Traceability	<p>Map relationships between papers</p> <p>The ability to take snapshots and roll back features</p> <p>Consistent documentation of results across the process</p> <p>Support for tracing extracted data back to source study</p>
4	Support Text Mining	<p>Natural language processor for included studies</p> <p>Automated search of similar studies based on defined keywords</p> <p>Textual analysis support</p> <p>Keywords definition support</p> <p>Automated keyword/phrases extraction, topic analysis in primary studies</p> <p>Support search automation for evidence in study</p>

#	Consolidated Requirements	Requirement
5	Support Inclusion and Exclusion	Semi-Automated inclusion/exclusion of studies
		Automated support to map snowballing of references
		Support pilot of inclusion/exclusion
		Semi-Automated to conduct snowballing
		Support for inclusion/exclusion criteria management
6	Support Quality Assessment	Support quality assessment of studies, protocol, and process activities.
7	Data Maintenance	Reference management
		Document management
		Data management
		Support backup and maintenance of data for longterm evolution of research
8	Automated Analysis	Coding Support
		Automated analysis and summary of coding
		Automated classification based on coding
		Automated reports generation of studies meta-data
		Automated reports generation of statistical data, list of included/excluded stuides, and criteria
		Support graph generation
9	Visualization	Visualization support of cluster of papers
		Visualization support of qualitative findings
		Generate presentations using LaTeX
10	Coding of Methods and Data	Support encode study methodology
		Support categories management
		Quantitative data capture

#	Consolidated Requirements	Requirement
		Support codes management and results
		Qualitative data capture
		Support pilot of data extraction
		Support PDF annotation
11	Storage of Studies	Automated storage of PDFs
		Management of stored studies
		Ability to import external studies not from digital libraries

2.4 Current Tool Support

Researchers have developed a number of tools to support various aspects of the SLR process. In order to determine whether there is a need for additional tool support, we analyzed these tools using the requirements identified in the previous section. This section describes our tool analysis methodology, the tools analyzed, and the results of the analysis.

2.4.1 Tool Analysis Methodology

The goal of this analysis is not to judge the usefulness or value of the various tools, nor to rank the tools. We realize that tool developers have chosen which aspects of the process to support based on various factors. Conversely, the goal of this analysis is to determine whether there are any tools (or combinations of tools) that provide complete coverage of the SLR tool requirements provided by the members of the SLR community. For the sake of space, the paper describes the analysis results only for the eleven requirements included in Section 2.3. The analysis results for the complete requirements set can be found in our online repository (<http://carver.cs.ua.edu/Studies/SLR-Requirements/>), where we will continually update the results as new tools and new tool versions are released.

To perform this analysis, we chose the *feature analysis* method [39]. This method is

one of the evaluation defined in the DESMET methodology used for evaluating tools and methods [36]. We chose this methodology because its goal is to report information about the suitability of a tool within a specific context, rather than trying to rank the tools. In addition, this method is a well-established evaluation method in software engineering [23, 26, 42]

For each tool analyzed (see Section 2.4.2), we attempted to perform the tasks described by each composite requirement (including all of the detailed requirements contained within the composite requirements). In each case, we determined whether the tool supported the detailed requirement fully, partially, or not all. Then we generated a *total coverage score* for each tool as follows:

- We gave each of the composite requirements (from Table 2.5) a score from 10 - 1 (based on the ordering).
- We then divided the composite requirement's score evenly among its detailed requirements (i.e., Detailed requirement weight = Composite weight/the number of detailed requirements).
- We assigned a coverage score to each detailed requirement of *fully*, *partially*, or *none*.
- For each tool, we then computed a composite score as the sum of the scores of the detailed requirements.
- Finally, we computed a coverage percentage for each composite requirement, and averaged the composites percentages to compute the total coverage score for each tool.

Table 2.7 shows an example of how the calculation was done for tool_X for the collaboration requirement composites.

2.4.2 Tools Analyzed

Based on a search of the literature, we identified twelve tools that provide some type of support for the SLR process. Because our goal in this work is to analyze how well tools support the entire SLR process, we excluded those tools that focused on one phase only.

Table 2.7: Tools Analysis Example

#	Req. Composite	Req.	Req. Weight	Composite Weight	Tool X Feature	Tool X Feature Score	Tool X Composite Score	%
1	Collaboration Support	Support for distributed review teams	3.3	10	Full	3.3	4.9	49%
		Support for collaborative SLR	3.3		Partial	1.6		
		Support for conflict resolution	3.3		None	0		

This exclusion left six tools which provide support for the entire SLR process: StArt [27], SLuRP [6], SLR-Tool [20], SLRTOOL [4], SESRA [43], and Parsfial [21]. Our analysis differs from a previous tool evaluation by Marshall et. al [42]

- It uses a different list of features, which is based upon the results of the community studies described earlier,
- It analyzes a newer version of the StArt tool released since the initial study, and
- It adds two tools: Parsfial and SESRA.

The evaluation of the tools at first was done by the first author. However, to ensure validity of the evaluation we sent the evaluation results for the tools authors. The tools authors confirmed the validity of our evaluation and modified the items that we have misjudged. Then all the authors discussed the evaluation and ensured its validity.

2.4.3 Tool Analysis Results

Table 2.8 shows the coverage score for each tool on each composite requirement. The columns are sorted by the overall coverage of the tools. The coverage percentage for each tool for each composite requirement is calculated based how much of the detailed requirements

the tool covers. The final row of the table provides a weighted average of the coverage percentage of the individual requirements in the top 11 composite requirements. The detailed results of each tool are in the online repository at (<http://carver.cs.ua.edu/Studies/SLR-Requirements/>). This analysis represents the current state of the tools. We will keep this analysis updated on the webpage as tools change.

Based on the data in the table, we can make a few interesting observations. First, the coverage of the composite requirements varies greatly. For example, *Collaboration Support* and *Data Maintenance* are well-supported in multiple tools, while *Coding of Methods and Data* is either unsupported or only very partially supported. For most of the other requirements, the results are mixed. Second, the coverage of the tools varies greatly. For example, the newer tools (StArt, Parsifal, and SESRA) have a higher overall coverage score than the other tools. Third, there is no one tool that is always best. While StArT, Parsifal, and SESRA have generally the best coverage, they are not always the best for any individual requirement. For example, *Automated Analysis* and *Visualization* are supported better by SLuRP. Therefore, there is currently no one tool (or combination of tools) that provide support for all the functionality desired by the SLR community. The next section provides additional discussion of the results.

2.5 Discussion

This section first describes a series of observations from our the analysis of the study results. Then, it provides a vision for an integrated framework that will support the needs of the SLR community.

2.5.1 Observations

As we conducted the underlying research, we made seven general observations.

Requirements are everywhere We have conducted three studies on the SLR barriers and requirements. In this work we have integrated the results of these three studies and built one list of functional requirements that SE community needs in tools to support the

Table 2.8: Tools Analysis Results

#	Consolidated Requirements	StArt	Parsifal	SESRA	SLuRP	SLRTOOL	SLR-Tool
1	Collaboration Support	100%	100%	100%	100%	66%	0%
2	Integrated Search	26%	62%	53%	17%	17%	17%
3	Traceability	100%	75%	75%	25%	50%	50%
4	Support Text Mining	23%	30%	7%	14%	14%	14%
5	Support Inclusion and Exclusion	60%	30%	30%	20%	30%	30%
6	Support Quality Assessment	100%	50%	50%	50%	50%	50%
7	Data Maintenance	100%	100%	100%	100%	100%	100%
8	Automated Analysis	50%	50%	50%	66%	33%	33%
9	Visualization	60%	60%	60%	90%	60%	60%
10	Coding of Methods and Data	14%	0%	0%	14%	14%	0%
11	Storage of Studies	66%	66%	66%	66%	66%	66%
Total		68%	63%	59%	48%	44%	32%

SLR process. These requirements are spread over the all phases of the SLR process which means there are still barriers that need to be addressed in all the phases of the SLR process. Some SLR phases have a higher number of detailed requirements than others. For example, while the Search phase contains the most detailed requirements (45), all other phases contain at least 9 detailed requirements. In addition, the requirements focus on different types of issues. For example, some address limitations in the current infrastructure while others address limitations in current tool support. We believe addressing these requirements will increase the ability of the SE researchers to perform more and faster SLRs.

Gaps in current tools The results of the tool analysis shown in Table 2.8 indicates that each tool has different strengths. None of the existing SLR tools cover all the im-

portant requirements elicited from the SLR community. The six tools we analyzed each cover 32-68% of the most important requirements. To better understand the coverage, we examined how well each tool covers all of the detailed requirements within the composite requirements. Due to space, we posted the results of this analysis on a webpage (<http://carver.cs.ua.edu/Studies/SLR-Requirements/>). That analysis showed that most of the composites requirements cannot be fully covered by a single tool. For example, to cover all of the detailed requirements inside the *Integrated Search* composite requirement, an author must use both StArt and SESRA. While SESRA has a higher overall coverage for *Integrated Search*, StArt covers some of the requirements that SESRA misses. Also, some of the detailed requirements are partially covered by multiple tools. For example, Parsifal only searches some online article databases, while SESRA searches different databases. StArt observation emphasizes the need for a mechanism to integrate the functionality provided by multiple tools. While the current tools perform their functionality well, the overall benefit would be greater if they could easily work together.

Collaboration is needed The SLR process by its very nature is collaborative. To reduce the effects of single-researcher bias, multiple researchers must collaboratively execute the SLR process by reviewing and synthesizing research from the published literature. Our work identified requirements highlighting the need for collaboration across all phases of the SLR process, including: protocol development, search, selection, conflict resolution, data extractions, synthesis, and report writing. Therefore, there is a need to facilitate collaboration both among co-located teams (e.g., PhD students and their advisor) and among distributed teams (e.g., researchers from different countries).

Automation is missing The SLR process is time consuming due to the large number of manual steps. While existing tools support various aspects of the SLR process, they do not eliminate the need for manual effort. Automating aspects of the SLR process will reduce the amount of manual work SLR authors must perform. The SE community needs tools to automate the process. Most of the identified requirements describe the automation of SLR

steps across the entire SLR process. Specifically, the requirements include automation for: building protocols, planning the SLR, search execution, study selection, data extraction, and generating graphs and reports.

Reuse and evolution With the increasing number of SLRs performed by SE researchers, it is likely that the same paper will be relevant to multiple SLRs. Because there is no central repository for storage of extracted data, each team of researchers must repeat the data extraction process. Persistent data storage of extracted information would both reduce effort, by enabling a researcher to extract only the additional data relevant to the new research question(s), and facilitate collaboration, by allowing researchers to identify others working on similar topics. In addition, storage of data in a central repository will facilitate the goal of making SLRs into living documents that can evolve as new research is published. Researchers could more easily integrate new findings with the existing results by taking advantage of the access to stored data.

Integration of current tools As mentioned in the discussion of the tool analysis, each tool has its own strengths. However, data integration among these tools so that the the results of one can be easily used as input to another is missing. To fully benefit from the strengths of each tool, researchers should be able to move data among them with little manual "translation" effort. Because this ability is current lacking, researchers are either locked-in to a particular tool and the functionality provides or avoid using a tool because of the lack of portability of the data.

Newer tools The analysis in Table 2.8 shows that the new tools (SESRA and Parsifal) and the updated version of StArt do a more thorough job of addressing community needs by covering more requirements. These three tools have the highest overall coverage of the community requirements. The updated version of StArt has made it highest ranked tool in terms of requirements coverage. This result indicates that tool authors are building more robust tools that can provide support to SLR authors. Even with these advances, there is still work work to be done to fully cover all of the important SLR requirements.

2.5.2 Suggestion for Infrastructure Developers

Based on the studies described in this paper and the observations across studies, we believe the community will benefit from an enhanced SLR infrastructure. By infrastructure we do not mean a single tool, it is a combination of existing and new tools that all work together to support the SLR process. Where all the tools exchange input/output with each other using the main infrastructure. We believe the open infrastructure should address the following based on the observation of the study:

- **Automation:** To reduce the cost of the SLR, there is a need to reduce the amount of manual work and automate or semi-automate tasks in the SLR process.
 - Provides support to automate the SLR process and reduce the manual tasks across all the SLR phases.
 - Support automated guidance.
 - Support cross-phase coordination and revised planning as well as iteration within and among SLR Phases.
 - Maintain queries, protocols, article links, and meta-data.

- **Collaboration:** To reduce bias, one of the key goals of an SLR, there is a need for multiple researchers to work together. Functionality to support collaboration should address the needs of multiple researchers working on SLRs.
 - Support project collaboration and coordination of work teams (comprised of multiple researchers, geographically dispersed, and working asynchronously).
 - Support monitoring of progress and quality.
 - Support revised planning.
 - Reduces the effects of single-researcher bias in the SLR process by providing conflict resolution tools.

- **Reuse and Evolution:** To keep the results of SLRs relevant, there is a need for the ability to update them as new research emerges. In addition, researchers should be able to reuse the work done on previous SLRs when performing a new SLR.
 - Support future extensions through persistence of SLR templates and data.
 - Support data interchange across all SLR phases.
 - It should provides persistent storage of extracted data to reduce effort, and enabling researchers to extract only the additional data relevant to the new research question(s), as well as facilitating collaboration by allowing researchers to identify others working on similar topics.
 - Provide central storage of data that will facilitate the goal of making SLRs into "living" documents that can evolve as new research is published. In addition, Researchers can more easily integrate new findings with the existing results by taking advantage of the access to stored data.

- **Integration and Openness:** There are a number of existing tools that support aspects of the SLR process well. However, it is not easy to move data from one tool to another. Therefore, integration of existing tools is important.
 - Support existing tools through workflow integration and data interchange wrappers.
 - Support new tools through both wrappers and add-in integration.

2.6 Threats to Validity

The first threat to validity relates to our methodology for mapping the results of S1, S2, and S3. It is possible that we interpreted a barrier or requirement differently than the participant intended. To reduce the effect if this threat, two researchers performed the analysis independently and met to resolve any discrepancies and produce the final mapping.

The second threat to validity is that the final requirements list could omit some important requirements. There could be additional requirements desired by members of the SLR community who did not participate in our studies. To mitigate this threat, we sent the survey to all published SLR authors. In addition, the participants in S2 and S3 were almost completely disjoint, lending additional validity to the obtained results.

Finally, our selection of tools to analyze could pose a validity threat. While we examined all published literature and talked with SLR authors to identify all existing SLR tools for SE, it is possible that tools exist which have not been published or are not known within the SLR community. We think that this situation is highly unlikely because of the number of SLR authors involved in our studies. In fact, we did identify one tool from our discussions with authors that was not published in the literature, Parsifal (included in our study).

2.7 Summary and Future Work

The SLR process, which is widely used in SE, still suffers from a number of barriers that result in time-consuming, manual work. Most of these barriers are due to the lack of tool support. In this paper, we worked with the SLR community to develop a list of SLR tool requirements by combining the results of three empirical studies: a community survey, a community workshop on barriers, and a community workshop on requirements. We then analyzed the existing SLR tools to see how well they covered this list of requirements.

The results of this work show that, while there are good tools that support SLR tasks, there is still a gap in the functionality provided by these tools. The current level of functionality provided is less than what is desired by the SLR community. The list of requirements identified in this paper can serve as reference for SLR tools developers and researchers to produce solutions that address the most important requirements. Because most SLRs are collaborative, collaboration support is one of the most required features, across all phases of the SLR process. Adding support for collaboration to existing tools would be of great benefit to the SLR community.

As future work, we will explore these requirements in more detail and begin building them into our own tool set. We will also continue to analyze SLR tools as they are released to update our online community resource.

2.8 Acknowledgments

The authors would like to thank the workshop participants for their inputs. We acknowledge partial support from the United States National Science Foundation under grant NSF-1305395.

CHAPTER 3

IDENTIFICATION OF REQUIREMENTS FOR TOOL SUPPORT FOR SLR SEARCH: SLR AND A SURVEY OF SLR RESEARCHERS

3.1 Introduction

Literature reviews are foundational for scientific endeavors because they enable researchers to synthesize the current knowledge about a topic, assess the feasibility of new research, and ground new research in existing literature. The Systematic Literature Review (SLR) method, originally developed for medical research, provides specific rules and guidelines to identify, assess, and synthesize literature [31]. By following the SLR process, a researcher can produce a literature review that is formal, repeatable, and most importantly, unbiased. Seeing these benefits in medical research, software engineering (SE) researchers adopted the practice of SLRs in 2004 [38]. SLRs are beneficial to researchers, by providing clear motivation for new work, and to practitioners, by providing comprehensive evidence to guide decision-making [5, 46].

However, most researchers do not find the task of performing an to be simple. SLRs are time- and effort-intensive due to the large amount of required manual work [1, 17, 48]. In addition, researchers face a number of barriers when performing SLRs. These barriers result from limitations in digital library search facilities, lack of complete tool support, and limited infrastructure to support SLRs in SE [24, 25, 33, 55]. As the popularity of SLRs in SE increases, it becomes increasingly important to overcome or reduce these barriers [3].

One of the critical barriers and most time-consuming phases of the SLR process is searching the digital libraries to select the most appropriate studies (papers) for inclusion in the

review [8, 24, 25, 37]. The presence of these barriers often leads researchers to conduct SLRs in non-systematic manner, which goes against a key reason for performing the SLR and reduces the overall quality of the review. Therefore, the goal of this paper is to *to aid researchers in performing SLRs by understanding how researchers currently perform the SLR search phase, the barriers they encounter during the search phase, and their tooling needs.*

To address this objective, we performed two studies focused on the SLR search phase. In both studies, we gathered information about search problems, solutions employed, and use of tools. In the first study, we conducted our own SLR of SLRs published between 2005 and 2015. In the second study, to validate the literature findings and ensure that we captured the voice of the larger SLR community, we conducted a survey of published SLR authors. After reporting on the results of each study, we compare the results to draw overall conclusions about the SLR search phase.

The primary contributions of this paper are:

- A concrete list of problems in the SLR search phase;
- A list of requirements from which researchers and tool builders can draw; and
- An analysis of the strengths and weaknesses of the tools used by SLR authors.

In Section 3.2, we provide background information to motivate the research questions. In Section 3.3, we discuss the methodology we followed to conduct the SLR on searching. In Section 3.4, we present the results of the SLR on searching. In Section 3.5, we discuss the survey methodology. In Section 3.6, we describe the survey results. In Section 3.7, we discuss the study findings and the current tools support. In Section 3.8, we enumerate the threats to validity. Finally, in Section 3.9, we explain the conclusions and our future work.

3.2 Research Questions

This section describes the research questions that drove the design of the SLR and the survey. For each research question, we provide a motivation based upon relevant published

literature. We identify three sets of research questions: (1) questions related to the barriers faced by SLR authors during the search phase; (2) questions related to the current practice of SLR authors when searching for papers; and (3) questions related to SLR search tools, both those SLR authors currently use and those SLR authors would like to see developed.

3.2.1 Barriers in the SLR Search Phase

Even though the SLR process is valuable to the SE community, authors still face a number of barriers, especially during the search phase. A comprehensive SLR published in 2013 found a number of papers that discuss barriers caused by the digital libraries themselves and by the lack of a fully-automated search capability [33]. Similarly, in Carver et al.'s survey of SLR authors to identify and rank the SLR barriers, they found that the SLR search phase was one of the most time consuming, most in need for tool support, and most difficult to conduct [8]. Other studies that address SLR search barriers find that the issues are caused by various factors, including:

- Digital libraries have limited support for the SLR process and the infrastructure is not suited for SLRs in SE [7, 12, 14, 16, 56, 57];
- Digital libraries return a large number of results, many of which are irrelevant, resulting in a large amount of manual effort to identify relevant studies [17];
- Researchers have difficulty defining keywords and terms when building search strings due to the lack of formality and consistent terminology in SE [8, 50, 54]; and
- The chances of multiple researchers using the same terminology is less than 20% [19, 52];

In addition to these findings from the literature, we have conducted surveys [8] and community workshops [24, 25] to identify barriers and tool needs directly from members of the SLR community. In the workshops, we gathered SLR experts attending the ESEM and the EASE conferences to identify and prioritize SLR barriers and SLR tool requirements. During the first community workshop, the attendees rated the lack of automated support for searching the literature and identifying relevant studies as one of the top barriers. Overall, the

workshop attendees indicated that the SLR process suffers from the lack of appropriate tool support across all phases. During the second workshop, the attendees identified, classified, and prioritized the requirements for SLR tool support. Of the 91 requirements identified, the cluster related to searching was the largest and received the second-highest priority. Table 3.1 lists the specific SLR search requirements along with the overall percentage of votes received across all requirements.

Table 3.1: Search and Selection Stage Requirements

Requirements	Number of Votes	Percentage
Integrated Search	16	11%
Study Selection	9	6%
Semantic Search	6	4%
Study Storage	5	3%
Collaboration	4	3%
Export of References	0	0%

As a result of the perceived importance of SLR search and the barriers researchers face during this process, we pose the following research questions, to be answered by both the SLR and the survey:

***RQ1:** What barriers do SLR authors face during the SLR search phase?*

***RQ2:** What solutions do researchers employ to overcome the barriers described in RQ1?*

3.2.2 Current SLR Search Practice

Due, at least in part, to the difficulties with the SLR search process, researchers have investigated other ways of performing the search. One common approach is *snowballing*, that is following the citations of included papers to identify additional relevant papers. Interestingly, while the snowballing approach results in a different set of papers than the traditional database search approach, the overall SLR conclusion are similar using either approach [29, 51].

Conversely, other studies have found differences between searching and snowballing. For example, database searching has higher recall but lower precision than snowballing [19].

Snowballing led to a larger and richer set of papers than the traditional database search [2]. The benefits of snowballing are strong enough that one author even published guidelines for using snowballing in the SLR search phase [53].

Snowballing is an example of an alternate searching strategy researchers use to handle the barriers. It is important to gather information that will help us better understand how SLR authors currently perform SLR search and what other methods they use to handle the barriers. Therefore, we pose the following research question to be answered by the survey:

RQ3: *What are the most effective search strategies employed by researchers during the SLR search phase?*

3.2.3 Current and Desired Tools for SLR Search

Kitchenham et al. identified a list of tools that claim to support the SLR process. However, they also report that current tool support is not sufficient and there is still a need for better tool support [33]. Similarly, in the community workshops described earlier, SLR authors the workshop attendees prioritized the barriers related to search as being among the most important and most in need of tool support [24, 25]. Other researchers report similar findings about the limited tool support for the SLR search phase [1, 7, 16, 41, 42].

To better understand how SLR authors currently use tools, even those not directly designed for SLRs, and to provide a list of requirements for tool builders, we pose the following questions, to be answered by the survey:

RQ4: *Which tools do SE researchers use in the SLR search phase? What are the advantages and disadvantages of these tools?*

RQ5: *What functional requirements do SLR researchers desire in tools specifically in the SLR search phase?*

3.2.4 Validation of Barriers in Literature

Finally, in order to triangulate the results and obtain the most consistent picture, we need to compare the results from different data sources. As described earlier, this study consists of a SLR as well as a survey of SLR authors. It is important to ensure that the barriers captures in the literature truly reflect the barriers faced by SLR authors. Furthermore, it is possible that SLR authors face barriers that they do not report in their SLR papers (since the goal of the SLR paper is to report the results of the SLR rather than to report barriers in the process) Therefore, we pose the final research question, to be answered by both the SLR and the survey:

***RQ6:** Which of the results from RQ1 match the barriers we found in the results of our SLR?*

3.3 Methodology: SLR

We planned and conducted this SLR based on published guidelines in software engineering [31, 35, 38]. We designed the SLR to specifically address RQ1 and RQ2 (Section 3.2.1). The main goals of this SLR are (1) to identify specific problems faced by SLR authors during the search phase, (2) to identify solutions authors employed to address those problems, and (3) to identify any tools authors use to support the search phase.

To achieve these goals we searched for three types of papers. First, we searched for published SLRs that contained a description of a problem encountered in the SLR search phase. Second, we searched for meta-papers that studied the SLR process to understand its strengths and weaknesses from the perspective of software engineering researchers. Third, we searched for papers that described SLR tools, SLR tool features, experiences with SLR tools, or evaluations of SLR tools. These papers are likely to describe barriers in the SLR process that motivated the creation of the tool along with a description of how the tool provides a solution to those barriers.

Accomplishing these goals will help us to establish the current state-of-the-practice re-

Table 3.2: Search Strings

SQ#	Search String	Motivation
SQ1	(Systematic Literature Review OR Systematic Review OR SLR OR Systematic Mapping) AND (Software Engineering) AND (Searching OR Automated Search OR Search OR Digital Library) AND (Problem OR Barrier OR Issue OR Answer OR Solution)	Identify problems with SLR search in published SLR papers or SLR meta-papers
SQ2	(Systematic Literature Review OR Systematic Review OR SLR OR Systematic Mapping) AND (Software Engineering) AND (Tool OR Tooling OR Tools)	Identify tool support for SLR search

lated to SLR search and to identify areas in need of additional research and/or tool-building to help the software engineering community become more effective and efficient in conducting SLRs.

3.3.1 Sources and Searching Strategy

To find the relevant papers to answer RQ1 and RQ2, we constructed two search strings (Table 4.12). The goal of the first search string is to identify papers that describe problems in the SLR search phase (either in SLR papers or in SLR meta-papers). The goal of the second search string is to identify papers related to SLR tools.

We executed these queries on the set of digital libraries which are commonly used in software engineering SLRs: the ACM Digital Library, IEEExplore, ScienceDirect, Scopus, and Engineering Village [35, 40]. In addition to being the most commonly used, these digital libraries index high quality journals and conferences. Each of these digital libraries has the advanced search capabilities to properly execute the queries. In addition to the results of the search, we included a paper identified in our previous work, which was not found during the search process because it was not indexed by any of the digital libraries. Finally, we used snowballing to include any relevant papers referenced by those found in our search.

Initially, we targeted SQ1 more narrowly so that it identified only the papers that described problems with SLR searching. Unfortunately, this approach resulted in very few

identified papers (as few as 0 papers in some digital libraries). Therefore, to ensure that we found all relevant papers, we expanded the search string so that it found papers that described any problems in the SLR process. From this larger set, we could identify those that describe a problem specifically with search.

As this paper is about barriers in the searching process, we can report two barriers that we faced ourselves. First, regarding SQ1, the word *issue* is a reserved word in the Scopus digital library. To make the search execute properly, we had to put quotation marks around that word for the Scopus digital library. Second, in the ACM Digital Library, when we removed keywords "Answer OR Solution", we obtained two additional papers that were relevant to our search. The remainder of the ACM results remained unchanged regardless of the presence of those two keywords.

3.3.2 Inclusion and Exclusion Criteria

Because we were looking for some relatively specific information in each paper, we purposely made our search strings broad so as not to prematurely exclude relevant papers. As a result, we expected a large number of search results. To properly cull these results, we defined a good set of inclusion and exclusion criteria. We used these criteria to ensure all included papers are related to the research questions and provide valuable information to help answer those questions.

Inclusion Criteria

- Papers that report problems with the search phase of the SLR process in software engineering.
- Papers that report tools or tool features that support SLRs in software engineering and address one or more problems related to automated search and retrieval of results.
- Papers that report solutions or suggestions for solutions to problems faced during the SLR search phase in software engineering.

Exclusion Criteria

- Studies not related to the research questions.
- Papers that are not in English.
- Abstracts, proposals, posters, or short papers.
- Duplicated studies.
- Papers that are not peer-reviewed.

3.3.3 Paper Selection

After executing the searches, we used the following process to identify the primary studies for inclusion in the review. The initial search queries returned 1,385 papers. The *deduplication phase* reduced that number to 811 papers. During the *title elimination phase* we reviewed the paper titles and removed papers that were clearly not related to the research questions, i.e. papers from outside of software engineering and papers that did not reporting systematic reviews or systematic mapping studies. This step reduced the number of papers under consideration to 266. Because we were looking for specific information within the reporting of an SLR (rather than the overall results of the published SLRs), the *abstract elimination phase* was not very helpful. In reviewing the abstracts, we were only able to eliminate 7 papers, bring the total down to 259 for full text review. In both the *title elimination phase* and the *abstract elimination phase*, we took a very conservative approach by only eliminating papers that we were certain were not related to the research questions. The *full text reading phase* eliminated 173 papers that did not provide any insight into the searching problem, leaving 86 papers for inclusion in the SLR.

The first author performed the primary analysis. The second author validating a random subset of 20% of the papers. Of those, the authors had minor disagreements on 7% of the papers. We easily resolved these disagreements as they were the result of misreading not a misunderstanding. The first author checked the remainder of the papers carefully to ensure there were no other misreadings.

Table 3.3: Included Papers Distributions

Venue	Type	# Publications	%
Information and Software Technology (IST)	Journal	37	42%
Journal of Software and Systems (JSS)	Journal	8	9%
ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	Conference	6	6%
Journal of Software: Evolution and Process (JSEP)	Journal	3	3%
Empirical Software Engineering (EMSE)	Journal	3	3%
International Conference on Evaluation and Assessment in Software Engineering (EASE)	Conference	2	2%
<i>Venues with single papers</i>		28	35%
Total		86	100%

The resulting papers came from a relatively small number of venues. Table 3.3 lists the venues that provided two or more papers in the final set. Note that 35% of the papers came from venues that contributed only one paper each.

3.3.4 Quality Assessment

For this paper, we did not conduct the type of quality assessment common in most SLRs. Unlike the typical SLR which seeks to gather published evidence and synthesize it into an overall conclusion, our goal was to find instances of methodological problems and their solutions described by these papers. We primarily examined the methodology and lessons learned sections to identify these problems. Therefore, we did not conduct a typical quality assessment because we were not trying to weigh the reliability of different pieces of evidence.

3.3.5 Data Extraction and Synthesis

To ensure that we extracted the same information from each paper, we defined a Data Extraction Form. Based on the research questions, we identified the items in Table 3.4 as being relevant to our research questions. After extracting data from all papers, we synthesized it to answer each research question, as discussed in the next section.

Table 3.4: Data Extraction Form - Specific to Research Questions

RQ	Data Item	Description
RQ1	Problem	Description of the searching or storage problems
RQ1	Problem Cause	The source of the searching problem, i.e. the digital library, lack of tool support, or people.
RQ1	Relationship Between Study and Problem	Whether the paper is focused on studying the searching problem or just reporting it.
RQ2	Solution	Description of solution(s) or suggested solution to overcome the problems.
RQ2	Solution Effect	Does the solution affect the DB, a Tool, or People?
RQ2	Solution Evaluation	Whether the proposed solution was evaluated.
RQ2	Tool Name	If a tool is discussed.
RQ2	Tool Functionality	Tool functionalities to overcome the stated problem. If none, then a description of what was done to overcome the problem.
RQ2	Relationship Between Tool and Author	Whether the paper authors created the tool, just used the tool, or are evaluating the tool?
RQ2	Tool Limitation	Any limitations that prevent the researchers from performing searching or storage as desired using the tool.
RQ2	Tool Availability	Whether tool is available and whether it is free.
RQ2	Tool Link	Link to the tool if available.

3.4 Results: SLR

We executed our search queries in November 2015. This section presents the results and findings from the literature search. The section is organized around the two research questions defined in Section 3.2. Due to space limitations, we do not provide the list of included papers here. The complete list, along with the corresponding papers IDs, can be found here: http://http://carver.cs.ua.edu/Data/2016/SLR_Search/. The list is also available in the appendix of the dissertation.

3.4.1 RQ1: What barriers do SLR authors face during the SLR search phase?

We analyzed each of the included papers to extract the specific problem(s) identified regarding the SLR search phase. Column 3 in Table 3.9 lists each of the specific problems identified in the literature. We grouped these problems into five higher-level problem categories. The number in parentheses after each problem category indicates the number of specific problems that we grouped into that category.

- P1: Problems with building and manipulating the search string (2)
- P2: Problems with digital library front-ends including interfaces, displaying results, guidelines, and filters (9)
- P3: Problems with the digital libraries back-ends including syntax, algorithms, rules, and optimization (12)
- P4: Problems with lack of tool support to automate the searching process (3)
- P5: Problems related to researchers, including experience, skills, and limited access to papers (3)

To further understand the problems and their importance, we examined the sources of the problems in more detail. First, we analyzed whether each problem was identified during the course of conducting an SLR or was identified through some type of deeper analysis of the SLR process. If most problems are uncovered during the course of conducting SLRs, as opposed to only being found via a deep analysis of the process, then solving these problems will likely be of immediate benefit to SLR authors. For each problem, we categorized it as (1) an experience or lesson learned from an SLR (SLR), (2) research on the SLR methodology or process (MP), (3) research on SLR tool support (Tool), (4) research specifically on SLR barriers (Barriers), or (5) research specifically about the SLR searching problem (Searching). Table 3.5 shows that the majority of the problems originated in the course of conducting

Table 3.5: Problem Source: Study Types

Study Type	# of Studies	%	Studies
SLR	62	72%	S1, S3, S4, S5, S6, S8, S10, S11, S12, S13, S15, S16, S19, S22, S23, S25, S26, S27, S28, S31, S32, S34, S35, S36, S37, S38, S39, S40, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S59, S60, S61, S62, S63, S64, S65, S66, S68, S69, S70, S74, S75, S76, S77, S78, S81, S82, S83, S84, S85
MP	10	11%	S2, S9, S14, S24, S30, S33, S41, S42, S67, S71
Tool	6	6%	S43, S44, S72, S73, S79, S86
Searching	5	5%	S7, S20, S21, S58, S80
Barriers	3	3%	S17, S18, S29

SLRs. Therefore, addressing these problems should have direct benefit to a broad range of researchers.

Second, we examined whether the searching problems originated in the existing infrastructure (i.e., digital libraries, lack of tool support, or unstructured literature) or from the SLR researchers themselves. This information will guide researchers on how to best solve the problems. Table 3.6 shows the origin of the searching problems. Please note that because some studies report multiple problems, they will appear in multiple table rows. This information provides insight into the source of the problems, which guides the development of appropriate solutions. People-related problems require better guidelines and education. Other types of problems require solutions that provide automation or remove barriers.

Third, we examined the relationship between the paper author and the reported problem. We categorized each problem as either being reported by researchers in the course of conducting an SLR (Report) or by researchers specifically studying the problem (Study). Similar to the first analysis (Table 3.5) this information helps determine the extent of the potential impact of any solutions to the searching problems. The difference is that this analysis focuses on whether the paper was studying the searching problem or just reporting it,

Table 3.6: Problem Source: Infrastructure vs. People

Problem Source	# Studies	Study
Digital Libraries	82	S1, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23, S24, S25, S26, S27, S28, S29, S31, S32, S33, S34, S35, S36, S37, S38, S39, S40, S41, S42, S43, S44, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S58, S59, S60, S61, S62, S63, S64, S65, S66, S67, S68, S69, S70, S71, S73, S74, S75, S76, S77, S78, S79, S80, S81, S82, S83, S84, S85
Lack of Tool Support	6	S2, S17, S18, S19, S21, S29
People	4	S17, S29, S30, S58
Unstructured Literature	2	S17, S29
NA	2	S72, S86

whereas the previous analysis focused on the type of study that uncovered the problem. The results of this analysis, shown in Table 3.7, indicates that most problems were discovered in the course of conducting an SLR. The results also show a relatively small number of studies conducted to specifically study the problems in the SLR process.

3.4.2 RQ2: What solutions do researchers employ to overcome the barriers described in RQ1?

As shown in the response to RQ1, we identified a large number of problems related to the SLR search phase. This question gathers information about how researchers have solved or overcome those problems. Note that all of the identified solutions are not formalized, concrete procedures. Some solutions are mere workarounds to allow the researchers to complete the SLR, because the problems could not be easily solved. Based on our analysis, we grouped the solutions into six categories. For each solution, we first describe the solution and then discuss the potential weakness of it.

- **Iterative Search:** Repeatedly searching the digital library with different versions of a search string until an adequate search string is identified. *Problem:* by non-

Table 3.7: Problem Source: Author's Relationships to Problem

Type of Study	# Studies	Study
Report	70	S1, S3, S4, S5, S6, S8, S9, S10, S11, S12, S13, S14, S15, S16, S19, S22, S23, S24, S25, S26, S27, S28, S30, S31, S32, S33, S34, S35, S36, S37, S38, S39, S40, S41, S42, S45, S46, S47, S48, S49, S50, S51, S52, S53, S54, S55, S56, S57, S59, S60, S61, S62, S63, S64, S65, S66, S68, S69, S70, S71, S74, S75, S76, S77, S78, S81, S82, S83, S84, S85
Study	15	S2, S7, S17, S18, S20, S21, S29, S43, S44, S58, S67, S72, S73, S79, S80
NA	1	S86

systematically adding or removing search terms, the resulting set of papers may either be deficient, in that it is missing relevant papers, or overly large with many irrelevant papers, increasing the time required to process them.

- Alternative Sources:** When a digital library had some type of limitation, researchers used an alternate digital library. For example, researchers might search Google Scholar instead of the preferred digital library assuming (incorrectly) that Google Scholar indexes all literature. *Problem:* Even though there is some overlap among digital libraries, the overlap is not always 100%. This approach may lead researchers to omit papers that were in the original digital library.
- Different Search Strings:** To deal with the different interfaces and search rules of each digital library, researchers often use different search string for each digital library. *Problem:* This solution is non-systematic because each digital library is searched with different search terms. The power of SLR is to be systematic in all the stages.
- Tool:** The researchers used a tool to help overcome some barriers. *Problem:* Papers do not typically specify exactly how the tool is used or which barriers the tool addresses.

- **Other:** Other solutions include:
 - Guideline for searching digital libraries to retrieve better results
 - Using the QUASI searching approach to retrieve more relative papers
 - Conducting manual search on the digital libraries
 - Suggesting the use of information retrieval to overcome the limitations of digital libraries algorithms

- **None:** The paper reported no solution to the identified problems.

Table 3.8 provides additional information about each of these solutions. First, it shows which problem source(s) the solution impacts, i.e. DB, Tool, or People. Second, it shows the type of study from which the solution was drawn. This information is important to understand whether the solution came directly from SLRs or from studies about SLRs. Third, it shows which of the problems identified in RQ1 are addressed. Finally, it shows which studies employed the solution.

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
1	P1	Limitation on the search string	Iterative Search	S1, S23, S31, S41, S84
2	P1	Limitation on complex strings in DL	Iterative Search	S25
			Tool	S43
			Other	S37, S80
			None	S48, S67, S79

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
3	P2	Better approaches to display the results from DL are required	Iterative Search, Alternative Sources	S24
4	P2	Inconsistent filters for results across DL	Different Search Strings	S76
5	P2	Unable to conduct automated search on the interfaces of the DL	Tool	S2, S44
			None	S14, S21, S29
6	P2	Better interfaces to search DL are required	Other	S7, S80
			Alternative Sources	S24
			Iterative Search	S24
			None	S12, S14, S21, S79
7	P2	DL don't search singular and plural of terms	Iterative Search	S68
			Other	S7
8	P2	Some DL searches are not user definable only pre-established fields allowed	Other	S7
9	P2	The number of results is limited within the same page and limited to select all the results	Other	S7

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
10	P2	Inconsistent search of fields across DL	Other	S7
11	P2	Better guidelines are required for some DL to understand how to search them	Other	S7
12	P3	Difficult to integrate with DL	Other	S7
			None	S17
13	P3	Incomplete article abstract or full text in some DL	Other	S7
14	P3	DL index small group of journals and conferences	Other	S7
15	P3	DL are not utilized for SLR and SM	Alternative Sources	S9
16	P3	A lot of duplicated results	Other	S7
			None	S19
17	P3	Different syntax across DL (Need standards)	Iterative Search	S3, S4, S5, S6, S8, S10, S13, S15, S24, S31, S32, S40, S42, S45, S46, S47
			Alternative Sources	S9, S24, S55

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
			Different Search Strings	S27, S28, S49, S50, S51, S52, S53, S54, S57, S59, S61, S62, S63, S65, S69, S70, S71, S74, S75, S76, S78, S81, S82, S83, S85
			Other	S80
			None	S14, S21, S36, S56, S58, S67
18	P3	Special characters are not always allowed	None	S11
19	P3	DL backend issues	Other	S7, S20
20	P3	Some DL missing full text document	Other	S80
21	P3	Some DL do not search full text	Iterative Search	S42, S55, S57
			Alternative Sources	S55, S57
			None	S34, S35, S56, S58, S66

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
22	P3	Same DL produce different results based on the order of the words	Iterative Search, Alternative Sources	S24
			Different Search Strings	S38
			None	S60
23	P3	DL return a lot of irreverent results	Iterative Search, ,	S41
			Alternative Sources	S26
			None	S58
24	P4	DL search is time consuming	None	S18, S17
25	P4	No support for collaborative search	None	S17
26	P4	Unable to download large batch of results	Alternative Sources	S16
			Iterative Search	S64
			Other	S7
			None	S21, S33, S73, S77
27	P6	DL search require experiences in search	Other,	S20
			None	S58

Table 3.9: SLR Search Barriers in SE

#	Group	Problem	Solution	Study
28	P6	Limited access to papers	Alternative Sources	S26
			None	S29
29	P6	Unable to convert research questions to search strings	Iterative Search	S30
			None	S29

3.5 Methodology: Survey

In this section, we describe the methodology we employed to answer our research questions. To gather information about barriers, workarounds, and tool support in the SLR search phase, we designed and executed a survey of SLR authors. In the following we describe survey design, pilot testing of the survey, the participant selection process, data collection, and the data analysis process.

3.5.1 Survey Design

Using the research questions, we designed a survey instrument that would allow respondents to provide relevant information about problems and solutions in the SLR search phase. We designed the survey questions around four major themes. The four major themes along with their justification are:

- *Experience*: This demographic information allows us to measure and classify survey respondents based upon their experience with performing SLRs. We measured experience as the number of SLRs conducted. We classified respondents who had published only one SLR as novices. We classified respondents who had published more than one SLR as experts. This theme address the demographic questions.

Table 3.8: Reported Solutions

Solution	Solution Impact	Type of Study	Problems Addressed	# Studies	Study
Iterative Search	DB, People	SLR, MP	P1, P2, P3, P4, P5	27	S1, S3, S4, S5, S6, S8, S10, S13, S15, S22, S23, S24, S25, S30, S31, S32, S39, S40, S41, S42, S45, S46, S47, S55, S57, S68, S84
Different Search Strings	DB	SLR, MP	P2, P3	25	S27, S28, S38, S49, S50, S51, S52, S53, S54, S59, S61, S62, S63, S65, S69, S70, S71, S74, S75, S76, S78, S81, S82, S83, S85
Alternative Sources	DB	SLR, MP, BS	P2, P3, P4, P5	7	S7, S9, S16, S24, S26, S55, S57
Tool	Tool	Tool, MP	P1, P2	3	S2, S43, S44
Other	DB, People, Tool	SLR, BS	P1, P2, P3, P4, P5	5	S7, S20, S37, S80
NA	NA	SLR, MP, BG, BS, Tool	P1, P2, P3, P4, P5	24	S11, S12, S14, S17, S18, S19, S21, S29, S33, S34, S35, S36, S48, S56, S58, S60, S64, S66, S67, S72, S73, S77, S79, S86

- *Search Phase Barriers*: The main goal of the study is to identify the barriers in the search phase of the SLR process. The questions for this theme focus on identifying the problems authors face during the SLR search phase. This information is also helpful for identifying potential tool requirements. This theme address research questions RQ1.
- *Search Phase Solutions*: Due to the presence of barriers in the search phase, we wanted to understand the solutions authors had applied to overcome those barriers during their SLRs. This information is important because semi-systematic solutions could reduce the quality of the resulting SLRs. The answers to these questions can also suggest potential tool requirements to capture any best practices. This theme address research

questions RQ2, RQ3.

- *Search Phase Tool Support*: Finally, to concretely identify specific requirements for tool support, this them focused on two concerns. First, we wanted to gather the SLR authors' assessment of the tools they currently used for conducting SLR search. Second, we wanted to gather specific requirements that SLR authors would like to see embodied in the features provided by tools that support the SLR search phase. This theme address research questions RQ4, RQ5.

In Table 3.10 we list the survey questions. For each question, we provide a mapping to the Research Questions and a description of the possible answer choices. We designed the survey with both closed-ended and open-ended questions to gather as much information as possible about the barriers. We designed the survey to identify the problems in the search phase, to provide a snapshot of SLR authors' perceptions, and to discover relationships among the barriers, solutions, and experience.

3.5.2 Pilot Tests

To ensure the survey was valid and comprehensive relative to our research questions, we conducted two pilot tests. Using these pilot tests, we could refine any questions that were unclear or irrelevant. In the first pilot test, we asked SLR experts to review the research questions and resulting survey questions. We used their feedback to clarify the questions. In the second pilot, we had software engineering PhD students, with experience conducting SLRs, complete the survey based on their experiences. These student provided us with feedback about the survey to help us ensure they were properly interpreting the survey questions. The feedback resulted in minor changes including rephrasing few items in some questions and rearranging some answer choices..

3.5.3 Participants Selection

To gather valid data, we needed to ensure that the survey respondents had some experience with performing SLRs, otherwise their feedback would be less valuable. We identi-

Table 3.10: SLR Survey Questions

#	RQ*	Question	Answer Choices
Q1	D	How many software engineering SLRs have you Started (but not completed)	[#]
		How many software engineering SLRs have you Completed (but not published)	[#]
		How many software engineering SLRs have you Published	[#]
Q2	D	For the SLRs you have conducted/what was your status? (Please check all that apply)	[Undergraduate Student , Graduate Student , Faculty , Other]
Q3	D	Which software engineering topic(s) were the focus of your SLR(s)?	
Q4	RQ1	Have you faced any problems in the SLR search phase?	[Yes, No]
Q5	RQ1	If yes, please describe these problems	
Q6	RQ2	If you faced any problems, did you develop a solution or work-around?	[Yes, No]
Q7	RQ2	If yes, please describe the solution or work-around	
Q8	RQ2	If you solved them, did you evaluate your solution?	[Yes, No]
Q9	RQ2	If yes, what method or approach did you use to evaluate your solution(s)?	
Q10	RQ4	Did you use any tool to help in the SLR search phase?	[Yes, No]
Q11	RQ4	If yes, please provide the tool names?	
Q12	RQ4	If you used any tool(s), for each tool, please indicate why you choose the tool and which features were useful. (Note: If you used more than one tool, please indicate the tool for each response)	
Q13	RQ4	If you used any tool(s), If you found any limitations in the tool(s), please describe them. (Note: If you used more than one tool, please indicate the tool for each response)	
Q14	RQ5	Do you think the SLR search phase needs better tool support?	[Yes, No]
Q15	RQ5	If yes, which features would you like to see to better support the SLR search phase?	
Q16	RQ3	What are your most effective strategies for searching literature?	

*Note "D" refers to demographic questions

fied the subject pool by mining the papers from a set of SE conference and journals that commonly publish SLRs. The venues we mined were: *ESEM (Empirical Software Engineering and Measurement)*, *EASE (Evaluation and Assessment in Software Engineering)*, *ESE (Empirical Software Engineering)*, *IST (Information and Software Technology)*, and *IEEE Transactions on Software Engineering*. We restricted our search to papers published after 2011 to ensure that we included only those authors who had published and SLR recently enough that would be able to recall their experiences. After extracting all authors from the identified papers, we used Excel to identify duplicates in the emails and the names of the authors. After removing duplicates, we were left with a list of 638 SLR authors as our pool of potential survey respondents.

3.5.4 Data Collection

Using an online survey tool, Qualtrics, which ensures the anonymity of the respondents, we sent the link to the online survey to all 638 identified authors in September 2016. Of those 638 authors, 72 emails bounced, leaving a pool of 566 potential respondents. We sent a reminder to the list of 566 potential respondents two weeks after the original email. Once the response rate slowed, we closed the survey (approximately a month later). From the pool of 566 authors, 297 clicked on the survey link (52%), 191 began the survey (33%), and 131 completed the survey (23%).

We only analyzed the responses of those who completed the survey. Because some qualitative questions were optional, not all the participants answered all questions. We were still able to analyze their responses to the questions they answered.

3.5.5 Data Analysis

We worked together to analyze the qualitative data from the open-ended questions. Because we did not have a predefined set of codes, we used an open coding process to reduce any preconceived biases [10, 22]. Using NVivo, we coded the results in four batches (as described below). After each batch of answers, we met to compare our results and ensure consistency in the process.

As an initial attempt to understand the data, each author independently coded all questions for the first 10 responses. Each of us created our own set of codes based upon our interpretation of the responses. We then met to agree upon a consistent set of codes. Using those codes, we each recoded the first 10 responses. We then met to discuss and resolve any discrepancies. We repeated this process with the next 20 responses, adding new codes as needed. After meeting to resolve discrepancies, we then coded the next 30 responses. After meeting to resolve discrepancies, we coded the last 81 responses. Finally, we met to resolve any discrepancies. In the end, we both agreed upon all of the codes assigned to all responses.

3.5.6 Demographics

This subsection provide information about the demographics of the survey respondents. To get a sense of the participants experience in SLR we asked the participants to indicate the number of SLRs they have started but not completed, completed but not published, and published. As the number of SLR gives slight insight on the experience level in SLRs.

- The total number of SLRs published by the survey respondents is 370 (2.8 per author). The total number of SLRs completed but not published by the survey respondents is 149 (1.1 per author). The total number of SLRs started but not completed by the survey respondents is 149 (1.1 per author).
- All the respondents had worked on at least one SLR, with a number of respondents having a high level of experience. Figure 3.1 shows the total number of SLRs for each of the participants where: Total = Started but not completed + completed but published + published.
- Most of the survey respondents were faculty members. Figure 3.2 shows the participants status and the total number of SLRs worked on by each type of participant.

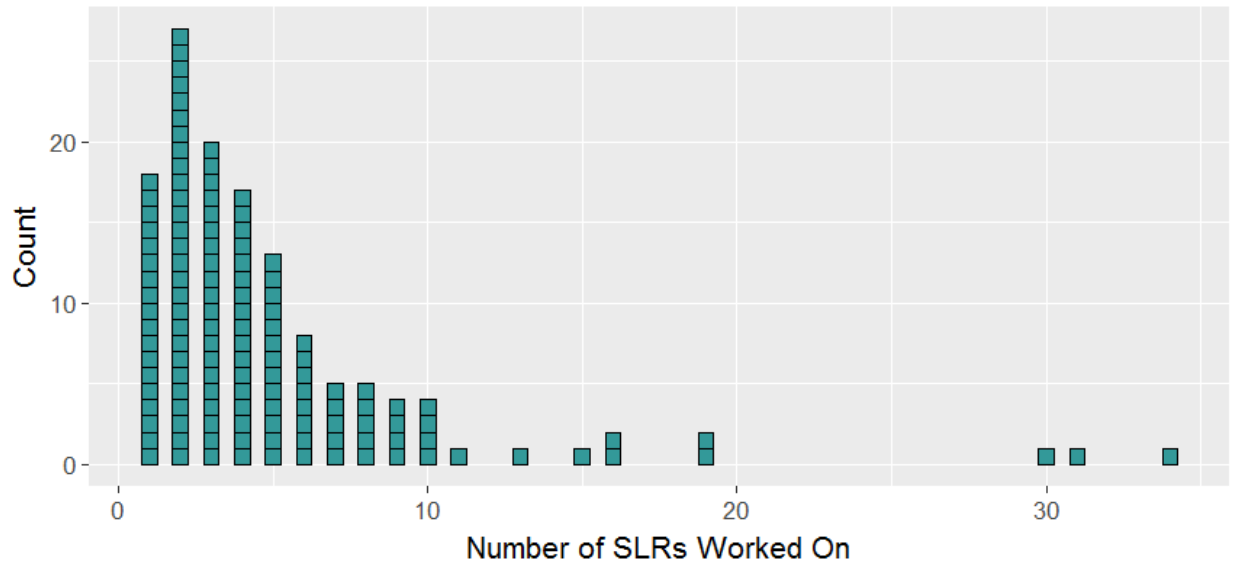


Figure 3.1: Number of SLRs Each Participants Worked On

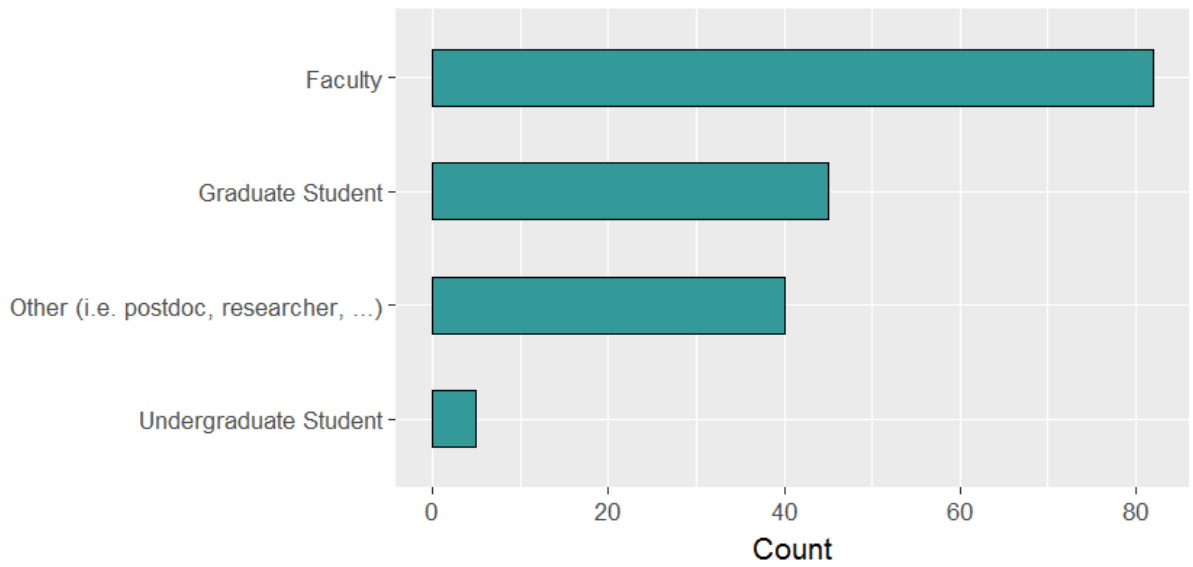


Figure 3.2: Participants Status and the Number of SLRs Worked On

3.6 Results: Survey

In this section, we present the survey results. In the next section, we discuss the implications of those results. We organize this section around each of the research questions.

3.6.1 RQ1: What barriers do SLR authors face during the SLR search phase?

To answer this research question we asked the participants if they faced problems during the search phase. Out of the 131 participants, 99 indicated that there were problems in the search phase. Of the remaining 32 who said they did not face problems in the search phase, 19 of them still indicated there was a need for better tool support for the searching phase.

The 99 participants who faced problems in the searching phase answered a follow-up question to describe the specific problem(s) they faced. We coded, analyzed, and grouped the 192 responses into 29 categories, as shown in Table 3.11. The most common problem in the search phase is that digital libraries have inconsistent syntax requirements for the search strings. This problem not only results in increased effort and time to build each specific search string, but it also might lead to differences in the meanings of the search string causing the quality of the SLR to drop. For example, as one respondent stated ” *Some of them [the digital libraries] were not consistent in handling logical operators such as AND or OR, making it difficult to achieve an interesting precision in the outcome ... [and] ... resulted in a lots of additional effort filtering the papers.*” Another example is that something as simple as capitalization of words can change how different digital libraries interpret the queries. Note that the category “Not Related” contained responses that did not belong to any other category and were not described clearly enough for us to meaningfully understand them.

3.6.2 RQ2: What solutions do researchers employ to overcome the barriers described in RQ1?

Of the 99 participants who faced problems in the search phase, 50 indicated that they had developed a solution or workaround to their problem(s). We grouped these solutions into 16 categories. The most common solution was ” *Splitting the search string into various*

Table 3.11: SLR Search Problems Identified by Researchers

#	Problem	# Reports
P1	Different syntax across libraries	31
P2	Choosing search strings	21
P3	Inconsistent terminology	15
P4	Search string limitations	13
P5	Filter limitations	10
P6	Downloading references	10
P7	Inconsistent results - within a search engine	8
P8	Limited access to full text	8
P9	Irrelevant results	8
P10	Choosing correct database	7
P11	Duplicated Papers	6
P12	Designing SLR	5
P13	Inconsistent results - across search engines	4
P14	Difficult to combine results from different libraries	4
P15	Sorting results	4
P16	Different Interfaces Across Databases	4
P17	Evaluating quality of search string	3
P18	High Number of Returned Results	3
P29	Incomplete results	2
P20	Study organization	2
P21	Reproducing results	1
P22	Determining type of paper	1
P23	Database changes over time	1
P24	Choosing research questions	1
P25	Missing Abstracts	1
P26	Difficult to Find Replicate Results	1
P27	Different Export Format	1
P28	Limited Number of Requests to DB	1
P29	Not Related	16

components and then performing various searches on the same database.” Table 3.12 shows the solutions that were reported by the authors, along with the how the authors evaluated their solution (when provided). Similar to the previous item, we have a “Not Related” category.

3.6.3 RQ3: What are the most effective search strategies employed by researchers during the SLR search phase?

Most respondents answered this question (102/131). We grouped the 154 responses into 15 categories, as shown in Table 3.13. Most of the researchers just use regular search in digital libraries using keywords and search strings.

3.6.4 RQ4: Which tools do SE researchers use in the SLR search phase? What are the advantages and disadvantages of these tools?

The respondents reported a large number of tools used to assist in the search process. Out of the 131 participants, 53 of them said they used a tool to help in the SLR search phase. Out of a total of 24 tools reported, only 4 are specific to the SLR process in SE: StArt, SLuRP, SESRA, and SLR Tool. The most used tool is Spreadsheet software e.g. Excel, Google Sheets, etc. Table 3.14 shows the names of the tools reported in the survey along with the number of times each tool was reported.

In addition to identifying the tools authors are using to support the SLR process, it is important to know why the authors chose these tools. We asked the respondents to identify the purposes and the limitations of each tool. Table 3.15 shows the five most commonly reported tools along with their advantages and disadvantages. Due to the length of the information about each tool, the rest of the tools are available online at <http://carver.cs.ua.edu/Data/2017/SLR-Tools/>. The list is also available in the appendix of the dissertation.

3.6.5 RQ5: What functional requirements do SLR researchers desire in tools specifically in the SLR search phase?

The answer to this question provides the most important outcome of our work as it had the goal of providing a list of tooling requirements that authors, software developers, and researchers can take in consideration when building tools to support the SLR search phase.

Table 3.12: SLR Search Solutions Identified by Researchers

#	Solution	Reported by	Evaluation	Reported by
S1	Iteratively Manipulate Search String and Multiple Searches	10	Check IRR, Compare to papers from trial and error	1,1
S2	Manual Work and Analysis	8	Expert judgment, Check IRR	1,1
S3	Obtain papers from other sources	4	Manual	1
S4	Trial and error	3	Check if known papers are displayed	2
S5	Snowballing	2		
S6	Read Documentation	2	Expert judgment	1
S7	Developed a tool	2	Controlled experiment	1
S8	Google Drive	1		
S9	Spreadsheet	1		
S10	Pilot Search String	1		
S11	Enrich the Paper	1		
S12	Standardize format	1		
S13	Restricting venues that are searched	1		
S14	Statistical random sampling	1	Different samplings techniques	1
S15	Develop gold set	1		
S16	Other	9	Expert checks the results	1

Table 3.13: Most Effective Search Strategies Identified by Researchers

#	Search Strategies	Reported by
SS1	Keywords and DB search	42
SS2	Snowballing	42
SS3	Search specific venues	9
SS4	Manual search	7
SS5	Use known papers and previous SLRs	7
SS6	Trial and error	6
SS7	Pilot test search	5
SS8	Search iteration	4
SS9	Search specific authors	3
SS10	Search strings keywords iteration	3
SS11	Use lists and filters	3
SS12	Expert knowledge	3
SS13	Define good search string	3
SS14	Search by title	1
SS15	Other	16

When asked if they needed better tooling support for the SLR search process, 102 out of 131 respondents indicated the need for better tools. The participants identified 167 requirements, that we grouped into 39 categories. Table 3.16 shows the requirements that were reported by the authors along with the number of occurrence of each one.

3.6.6 RQ6: Which of the results from RQ1 match the barriers we found in the results of our SLR?

We wanted to validate the SLR search phase problems that we found in the literature. As we mentioned previously in this study, we conducted an SLR to identify the SLR search problems in the literature. In the SLR we identified 29 problems reported in the literature between 2005 and 2015. We are checking the problems that appeared in the literature if they are reported by the researchers in the survey as well. This validation ensures the following:

- Understand which of the problems we found in the literature are still valid.

Table 3.14: Tools Used in SLR Search Phase Identified by Researchers

#	Tool	# Reported
T1	Spreadsheet (Excel)	22
T2	DL search engines	10
T3	Jabref	9
T4	Mendeley	9
T5	Custom tool	7
T6	StArt*	6
T7	End Note	5
T8	RefWorks	3
T9	Text mining	2
T10	Google scholar	2
T11	REviewER	2
T12	Zotero	2
T13	SLR Tool*	1
T14	Sharelatex	1
T15	Google drive	1
T16	Dropbox	1
T17	MS Word	1
T18	Publish or perish	1
T19	Reseachr	1
T20	SESRA*	1
T21	NVivo	1
T22	GSResearch	1
T23	SLuRP*	1
T24	Other	1

*Specific Tool for SLR in SE

- We wanted to know if the importance of the issues is the same in the literature and researchers.
- Check if the survey revealed more issues that we did not catch in the literature.

To answer this question, we compare the top ten problems identified in the literature to the top ten problems reported by the survey respondents. Table 3.17 shows the top ten problems reported in the literature. As we show in Figure 3.3, seven of the top ten problems

Table 3.15: 5 Highest Tools Used in SLR Search Phase Identified by Researchers

#	Tool	Purposes	Limitations
T1	Spreadsheet (Excel)	Organize, studies, Group data, Removing duplicates, Data extraction, Analysis, Track process, visualization, Collaboration, Inclusion and exclusion support, Classify data, Paper identification.	Not adapted for SLRs, Manual analysis, Cannot store PDFs, Data extraction, Printing graphs, Integration with other tools, Snowballing is not supported.
T2	DL search engines	Retrieve studies, Other, Remove Duplicates, Searching, Accurate, Comprehensive.	Search string differences across tools, Other, Manual analysis, Difficult snowballing, Too many false positive, Search string limitations, complex queries.
T3	Jabref	Organize studies, Availability, Filtering capabilities, Removing duplicates, Tailorable, Handle differently formatted references, Commonly used, Manage references, Annotation, Ease of use, Organize notes.	Difficult to customize, Defining search string, Search feature does not work, Complex tailoring is difficult, No support for defining search strings, Other.
T4	Mendeley	Organize studies, Organize notes, Generate bib file, Unknown, Coding, Study classification, Storage of results, Export references	No Customizable UI to store information, Not adapted for SLRs, No space to store paper, Removing duplicates is difficult, PDF annotation is independent, Tracing papers throughout SLR phases, Sync problems, Paper organization
T5	Custom tool	Data extraction, Summarization, Studies classification	Other, Manual work, Obscure

Table 3.16: SLR Search Phase Tool Requirements Identified by Researchers

#	Requirement	# Survey	# Literature
R1	Search multiple databases in standardized query	9	10, 8
R2	Removing duplicates	8	
R3	Filtering capabilities	7	
R4	Merging results from DL	6	New
R5	Synonym recommendation	6	New
R6	Repository for papers	6	
R7	Standardized export	6	New
R8	Download papers	6	7
R9	Automatic determination of relevance	5	New
R10	Support snowballing	5	
R11	Automatic generation of search terms	5	New
R12	Support collaboration	4	
R13	Papers categorization	4	New
R14	Iterative search	3	
R15	Retrieve more relevant results	3	2
R16	Search string definition	3	5
R17	Support visualization and graphical analysis	3	
R18	Papers summarization	2	New
R19	Managing notes	2	New
R20	Import search results to one tool	2	New
R21	Validation of data	2	New
R22	Archiving	2	New
R23	Tagging	2	New
R24	Support for searching process	2	4
R25	Aggregate data	2	New
R26	Extracting tables and figures	1	New
R27	Version control	1	New
R28	Automated analysis	1	
R29	Reference management	1	
R30	Meta data management	1	New
R31	Search papers based on similarity	1	New
R32	Directory for SE venues	1	New
R33	Semantic enrichment	1	New
R34	Import external resources	1	
R35	Suggesting articles based on keywords	1	New
R36	Easy to use	1	New
R37	Tracing studies sources	1	New
R38	Notification of new papers	1	New

Table 3.17: Top Ten SLR Search Problems Identified by the Literature

#	Problem
LP1	Different syntax across digital libraries
LP2	Some digital libraries do not search full text
LP3	Better interfaces to search digital libraries are required (Inconsistent interfaces even within the same digital libraries)
LP4	Unable to download large batch of results automatically
LP5	Limitation on complex strings in digital libraries
LP6	Limitation on the search string
LP7	Unable to conduct automated search on the interfaces of the digital libraries
LP8	Same digital library produce different results based on the order of the words
LP9	Digital libraries return a lot of irreverent results
LP10	Limited access to papers

identified in the SLR map to problems in the top ten reported by the survey respondents. Of the other three, LP3 and LP7 match to lower ranked survey problems, while LP2 did not match any survey problems, but did match a survey requirements. Of the three problems identified in the survey that do not map to the top ten from the SLR, SP4 matches a lower ranked requirements from the literature, while SP2, and SP3 are new requirements that did not appear in the literature.

The ability to match the problems reported in the literature to those reported by survey respondents confirms that the issues are still valid and need to be resolved. The matching also reveals some problems reported by survey respondent that were not reported in the literature. Taken together, these lists provide a comprehensive set of key requirements.

3.7 Discussion

In this section, we discuss the results described in the previous section to provide some overall implications of our findings. We can make a number of general observations about the SLR search process regarding the problems, the solutions, the requirements, and the tool support.

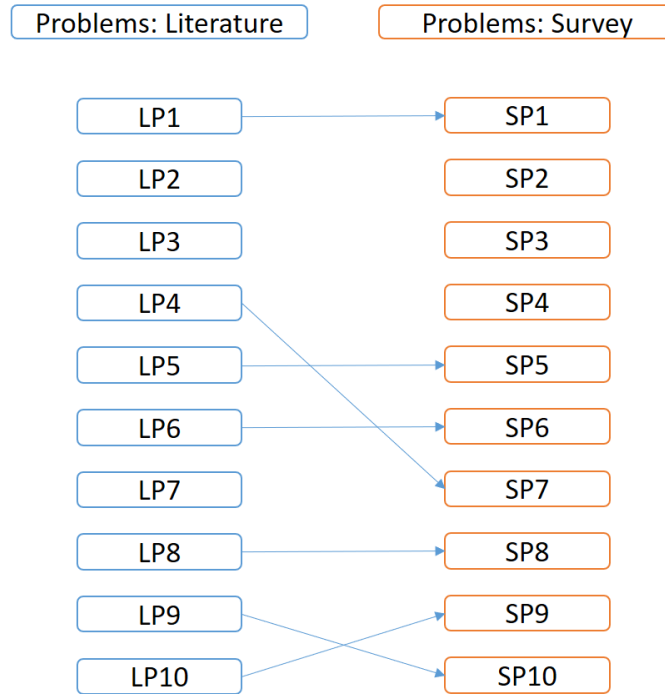


Figure 3.3: Problems in Literature Matched to Problems in Survey

3.7.1 Problems are Solvable

According to the literature and the survey results, 75% of the authors in the survey still face issues in conducting the SLR search process. A closer examination of the issues indicates that the problems are not inherent in the SLR methodology itself, but rather are the result of factors external to the methodology. In the following discussion, we identify three types of solutions to the problems, which all involve either better communication, better tool support, or both. As a reminder, in this paper we focus only on the problems related to the SLR search phase. Problems in other phases are outside the scope of this study and may have different solutions.

Digital Libraries

These problems originate directly with the digital libraries themselves and cannot be solved by a third-party tool. The solution to these problems is better communication with the

digital library providers. For example, one of the most common problems was that search engines provide inconsistent results, that is the same search query returns different results each time. The only way to solve this problem and other that are beyond the control of third-party tool developers is through better communication with the digital library problems.

The search process is critical to the success of an SLR because if the SLR authors do not start with the correct set of papers, they will not achieve the correct results. Therefore, problems that occur during the search phase have implications throughout the SLR process. For example, when a digital library search returns a large number of irrelevant results, the authors must spend extra time examining these results to have confidence in the final set of papers. This extra effort not only slows down the search phase, but it also affects the selection and data extractions phases.

There are currently no real solutions to overcome most of these problems. In cases where some problems could be address through tool support, the digital libraries often block (or make extremely difficult) automation, which prevents developers from creating tools to mask some of the issues. As respondents explained "*publishers limit the automatic download of their paper likely because of "... the detection of robot action by the search engines.*" The primary goal of the digital libraries is to provide researcher with easy access to its content. While we understand that the sheer size of these libraries will make any changes difficult, these changes are essential for higher-quality SLRs.

Automation Support

Another set of problems are those that increase the time required for the search phase due to the necessity of manual work by the SLR authors. Unlike the previous set of problems, these problem can be addressed through additional tool support to automate the corresponding tasks. As one respondent stated "*Basically most of the problems is (sic) to automate the process.*" While some of the reported problems are more difficult to automate than others, the SE SLR process still lacks automation for the small tasks like converting the search string

syntax to the appropriate format(s) for each digital library, importing/exporting studies in a standardized format, and recommending synonyms to create more comprehensive search strings. While these issues vary in implementation difficulty, their benefits will outweigh the cost of building them because they will reduce the overall cost of the SLR search phase.

SLR Authors

This category of problems are those that must be solved by the SLR authors themselves. In some cases, rather than needing more tool support or improved digital libraries, the authors need additional training or guidelines to help them overcome the barriers. Some of examples of these problems include "*How do you define the search terms, which data bases to include, how to search in the most efficient way*", "*No standards for conducting a search for studies. For example when to stop with sampling studies, etc.*", and "*Finding relevant venues/journals for the specific domain*". The existing technical guidelines plus publications about evidence-based software engineering and systematic reviews are a good resources for these type of issues [35, 40]. In addition, novice authors can benefit by consulting with more experienced SLR authors who can provide assistance in resolving many of these issues.

3.7.2 Existing Solutions and Workarounds

From the literature point of view 55% of the problems did not have a reported solution. As described in the answer to RQ2, even when a solution was proposed, it was often non-systematic. The real value of an SLR is that it uses a systematic process, which is transparent and repeatable by other researchers. The lack of systematic solutions to searching is problematic because of the potential for inadvertently introducing some type of bias into the SLR searching process, thereby reducing the value gained from using the SLR process and making the resulting SLR non-repeatable (at least in an exact form). In fact, due to the lack of systematic solutions to the searching problem, many SLRs may be semi-systematic rather than truly systematic.

As from the survey point of view, one observation from the solutions and workarounds

reported by SLR authors is that the most effective ones appear to be semi-systematic as well. Only about half of the respondents reported solutions or workarounds, with less than half of those actually evaluating the quality of their solution. For example, the problem of digital libraries each requiring a different syntax leads authors to modify their search strings multiple times until they are satisfied with the results. This *ad hoc* manipulation has the potential to result in slightly different searches and results for each digital library. Another example of a semi-systematic workaround results from the limits to search string length imposed by some of the digital libraries. In this case, authors decompose the search into multiple parts, which once again can result in different results than if the original search string could be use in its entirety. One survey respondent expressed these problems as: *"We spent a lot of time convincing ourselves that the results we got back were sane."* The primary way that authors evaluated these semi-systematic approaches was through comparing with a gold set of papers. As one respondent stated: *"The results were compared with the results obtained from the trial search"*.

When asked about the most effective overall search strategies, many respondents reported that they used only one methodology, e.g. snowballing or searching specific venues. The goal of the SLR is find all the related literature to a topic to answer the research question. Therefore, a combination of methodologies is required to achieve good coverage and precision in the search phase. Even though snowballing is an effective approach it is still does not find all the papers related to a topic [19, 29].

3.7.3 Tool Support is Still Needed

Even the small number of participants who said they did not face problems in the SLR search phase still indicated that there was a need for better tool support. Among the tools respondents used to conduct their SLRs, only four were designed specifically to support the SLR process in SE: StArt, SLuRP, SESRA, and SLR-Tool. Interestingly, the respondents did not report using these tools as often as they reported using other general purpose tools, like spreadsheets. This result is not completely surprising because most of the SLR-specific

tools do not provide much support for the search phase, which was the focus of this survey.

While the authors did use a number of tools to support the SLR, they did indicate that these tools had a lot of limitations. The primary limitation was that, even when using the tool, the author still had to perform a lot of manual work. Some examples from the survey responses include: "*The snowballing effect is difficult and require manual work. The removal of false positives require also lots of manual work*", "*Lots of manual cleaning of references was needed*", and "*We filtered manually the retrieved results*".

Another problem is that most of the tools are general purpose tools which are not well-adapted to the specific needs of the SLR search process. For example as one respondent stated "*excel: it is a general purpose tool that has to be adapted*". Even so, many of these tools were still useful: "*Excel is not really design (sic) to support SLRs but works reasonable well.*"

Based on the responses, we presented a list of the most important requirements for tools that support the SLR search phase. We invite SLR authors, researchers, and software developers to use this list as a reference for building improved tool support. We also provide a list of all the tools currently used to support SLRs along with their benefits and limitations at <http://carver.cs.ua.edu/Data/SLR-Tools/>. This list can be helpful for encouraging integration and collaboration among various tools.

3.8 Threats to Validity

In this section we describe the primary threats to validity of our SLR and survey.

3.8.1 SLR

Related to the primary goal of this study, one of the limitations of our study relates to the search process. As noted throughout this paper, there are a number of limitations with the infrastructure available to support the search process. Therefore, our search string might have missed some relevant papers if authors used terms different from the ones included in our string. We made every effort to be thorough and to use a variety of synonyms for the

key search terms, but there are still limitations.

Another limitation is our choice of digital libraries. We chose the digital libraries most commonly used in published SLRs. However, it is possible that we omitted papers that were not indexed in these digital libraries. All but one of the papers that we knew prior to conducting the SLR were found in the digital libraries we chose, so the chance that we missed important work is low. The paper that we did not find is not indexed by any digital library.

If the search process missed any papers, as indicated by the previous two limitations, then it is possible that there are additional problems or solutions not included in this report. Furthermore, if there are problems or solutions reported in papers that were not in English or not in peer-reviewed venues, they are also omitted from this report.

To help eliminate as much potential bias, we had the methodology reviewed independently. Furthermore, the second author reviewed a random subset of the papers at each phase of the SLR process to ensure that proper selections were being made.

3.8.2 Survey

The survey threats can be categorized in three groups.

Internal Validity

For the results to be valid, it is important that survey respondents have experience with the SLR process. For that reason, we defined our pool of potential respondents to be those who had published at least one SLRs in the software engineering literature. However, it is possible that some of the recipients shared the survey with students or colleagues who did not have the requisite experience. Based on the data about experience, we do not think this threat materialized in our sample. The other threat to internal validity is that we had no way of assessing the severity of each barrier or priority of each requirement. We used frequency of report as a proxy for importance, but future studies should focus specifically on identifying priorities.

External Validity

The main threat to external validity is the representativeness of the sample. It is possible that we missed SLR authors if they published their SLR in a venue other than the ones we searched. Since we had a broad sample, this threat does not appear to be serious in this survey.

Construct Validity

The most critical construct validity threat was that survey respondents may have misunderstood the survey questions and provided incorrect information. To reduce this threat, we pilot tested the survey with experts and with software engineering students (who had performed SLRs themselves) and updated the survey based on their feedback. Also our analysis of the responses indicates that overall the participants understood the questions as we had very few answers that were completely irrelevant.

3.9 Conclusion and Future Work

SLRs are widely used and continually increasing in popularity in SE venues. However, the large number of barriers faced by SLR authors results in a high cost for conducting the SLRs. The search phase is the phase with one the highest number of barriers throughout the whole SLR process. Therefore, in this study we focused on identifying the barriers and solutions specifically in the SLR search phase. We conducted an SLR and surveyed published SLR authors in SE to understand the issues specific to SLR search and to understand their needs for tool support.

We found that the search problems have been reported in different types of studies. There are few studies that specifically focus on problems in the SLR search phase in software engineering. Also, the SLR search phase still suffer from a lot of barriers, resulting in an SLR process that is still costly in time and effort. There is still need for future work, especially in

automating the tasks in the SLR search phase. The difficulty of implementing these solutions varies, but the benefit of automation will reduce the barriers in the SLR search phase and will reduce the cost of conducting SLRs in SE. There is also need for better communication with digital libraries providers, because some of the issues cannot be solved by third-party tools alone and require collaboration with the digital library providers.

The results of the analysis produced a list of requirements for SLR tool support to reduce the problems faced and the manual work required to complete the SLR search phase. The list of requirements can be used a reference for authors, researchers, and developers when developing tools for the SLR community in SE. The results also provide information about the current tools authors use along with the advantages and disadvantages of those tools.

The primary contributions of this work are:

- A list of requirements for tooling to support the SLR search phase; and
- A prioritization of those requirements based on input from the SLR community.
- An analysis of the strengths and weaknesses of the tools used by SLR authors.

3.10 Acknowledgment

We thank the survey respondents for taking the time to complete the survey.

CHAPTER 4

SSRT - A TOOL FOR SYSTEMATIC REVIEWS SEARCHING PHASE

4.1 Introduction

Literature review is the core of the beginning of any research. As it provides a snapshot of the status of the literature and helps motivates new research ideas. The lack of rules for conducting literature review leads to bias and difficulty for replication. Medical researchers have introduced the systematic literature review (SLR) approach to overcome those issues. The SLR process follows specific rules and guidelines resulting in a process that is clear from bias, repeatable, and well documented [35, 38].

Due to the benefits seen in medical field, the SLR approach was adapted for software engineering (SE) in 2004 [31, 38]. The SLR approach has been widely accepted and used in the SE community. The number of publish SE SLRs has increased to over 174 published between 2004 to 2013 [11, 28, 33, 37].

However, the SLR process still suffers from a lot of problems. The process still requires a lot of manual effort, which increases the effort conducting SLRs. These barriers fall in different aspects and stages in the SLR process and are caused by different factors [1, 8, 25, 48]. The top barriers in the SLR process are related to searching and obtaining the literature. Based on a survey and two community workshops, SLR authors in SE identified the SLR search phase as one of the most time consuming phases in need of tool support [8, 24, 25]. Our previous studies identified 29 problems in the SLR search phase (Article 2 - citation for the paper will be added once it is published). Also identified a list of tooling requirements

for the SLR search to overcome those barriers. The problems in the SLR search phase are also faced by other researchers based on multiple studies [1, 7, 12, 16, 56, 57].

The current tool support does not cover the searching process. Based on a study that evaluated SLR tools, only four cover all the stages of the SLR process, SLuRP [6], StArt [27], SLR-Tool [20], SLRTool [4]) and none support automated search [41, 42]. Also based on our previous working of analyzing the status of the current tool support, the tools are lacking the support for the SLR search phase. (Article 1 will be cited once it is published). Therefore, there is a need to automate the searching process to reduce the manual effort required in the SLR searching phase.

This paper presents SSRT (Searching Systematic Reviews Tool), a tool that we have developed to reduce the manual work required for the SLR search phase. SSRT helps automate most of the search tasks from building the search queries, to downloading the results, removing duplicates, and storing them in a database. We designed SSRT based on the findings from the SLR and the survey studies about the SLR search phase barriers that we have conducted (article 2 will be cited once it is published). In addition to describing the tool this paper will describe our evaluation of the tool.

This paper is organized as follow. Section 2 provides background and previous work. Section 3 presents the requirements, design, features, and limitations of SSRT. Section 4 contains the tool evaluation that we have conducted. Section 5 describes threats to validity in our study. Finally, section 6, conclude the paper and discuss future work.

4.2 Background and Previous Work

This section provides information on the SLR process, the barriers and tool support that have been identified in the SLR search phase and our own previous work.

4.2.1 SLR Search Phase Barriers and Tool Support

The SLR search barriers are the most difficult, most time consuming, and most in need for tool support [8, 24, 25]. The SLR search phase barriers are caused by different factors.

In a study comparing the SLR process in medicine to the SLR process in SE, researchers found that the digital libraries and search process are not optimized for SE as they are for medicine [12]. Other studies also support the lack of adequate infrastructure as they discuss the limitations of automation search in the SLR process and the effort needed to search and obtain the literature [1, 7, 12, 16, 33, 56, 57].

Tool support is important to reduce the manual work in the SLR search phase in SE. A mapping study identifies multiple tools [33]. Based on studies that evaluate tool support, only four tools cover all stages of the SLR process. The rest support only a subset of the SLR stages [41]. However, none of the tools focused specifically on the SLR search phase. The four tools that support all stages of the SLR process in SE (SLuRP [6], StArt [27], SLR-Tool [20] have been evaluated, however the outcomes of the evaluation studies stated the tools lack the automated search support in the SLR process [42].

4.2.2 Our Previous Motivating Work

To better understand the search phase barriers, we have conducted two studies. First, we have combined three studies that have been done focusing on the barriers and the requirements in the SLR process (Article 1 will be cited once it is published). The input of the three studies we combined have come directly from the SLR community in SE, through survey and two community workshops. The combination of the study provided a list of functional requirements for all the SLR process. We identified the SLR tools in SE, and evaluated these tools against the list of requirements that we have created. The results of the work showed that the searching part in the SLR process is very problematic. Also the tools that we have evaluated none of them have provided the appropriate tool support for the SLR search phase.

The second study that we have conducted consisted of two parts (Article 2 will be cited once it is published). The first part, we conducted an SLR to identify the search phase barriers reported in the literature. Because many studies identify barriers in the SLR process and because the search phase was one of the top sources of barriers, it was important to

better understand the search phase barriers. We reviewed the literature between 2005 and 2015 and identified 29 problems in the search phase of the SLR process reported in 86 studies. Those barriers were caused by different factors: (A) digital libraries, (B) lack of tool support, and (C) researcher experiences. For all these barriers, the researchers did not report the use of tools to address them. The solutions they followed were semi-systematic, thereby reducing the quality of the SLR and causing the SLR to be more difficult to replicate.

The second part, of the study, we have conducted a survey of SE researchers who publish SLRs to confirm the results of the SLR and ensure that we captured all barriers. 131 researchers have completed the survey. The results generated lists of issues, requirements, and tools being used by the researchers in the SLR search phase. We combined the survey results and the SLR results and produced a final list of tooling requirements specific to the SLR search phase.

Based on the analysis of the results of those two studies, and the analysis of current tool support, there is still need for better tool support in the SLR search phase. This motivated us to design and build SSRT, a tool to reduce the manual work in the SLR search phase in SE that researchers face.

Table 4.1 shows the top ten requirements from the survey of the researchers and their rank in the survey and the literature review as well. Please note that 1 is the lowest. The ones without SLR ranking mean they link to ones not in the top ten.

4.3 SSRT Tool

This section provides details on how we constructed SSRT. It includes the requirements we considered while building SSRT, the design of SSRT, and the features of SSRT.

To ensure that SSRT contains the most appropriate tool features, we chose the requirements based on input directly from the SLR community in SE. Thus, we conducted two studies to identify the requirements in the SLR search phase. In the first study, we combined three empirical studies on the barriers present in the SLR process to generate a list

Table 4.1: Top Ten Requirements

#	Requirement	# Survey	# Literature
R1	Search multiple databases in standardized query	9	10, 8
R2	Removing duplicates	8	
R3	Filtering capabilities	7	
R4	Merging results from DL	6	New
R5	Synonym recommendation	6	New
R6	Repository for papers	6	
R7	Standardized export	6	New
R8	Download papers	6	7
R9	Automatic determination of relevance	5	New
R10	Support snowballing	5	
R11	Automatic generation of search terms	5	New
R12	Support collaboration	4	
R13	Papers categorization	4	New
R14	Iterative search	3	
R15	Retrieve more relevant results	3	2
R16	Search string definition	3	5
R17	Support visualization and graphical analysis	3	
R18	Papers summarization	2	New
R19	Managing notes	2	New
R20	Import search results to one tool	2	New
R21	Validation of data	2	New
R22	Archiving	2	New
R23	Tagging	2	New
R24	Support for searching process	2	4
R25	Aggregate data	2	New
R26	Extracting tables and figures	1	New
R27	Version control	1	New
R28	Automated analysis	1	
R29	Reference management	1	
R30	Meta data management	1	New
R31	Search papers based on similarity	1	New
R32	Directory for SE venues	1	New
R33	Semantic enrichment	1	New
R34	Import external resources	1	
R35	Suggesting articles based on keywords	1	New
R36	Easy to use	1	New
R37	Tracing studies sources	1	New
R38	Notification of new papers	1	New

of tooling requirements for the entire SLR process (Article 1). The second study consisted of two parts that focused on the SLR search phase. In part one, we conducted an SLR to identify the problems the SE community have reported in the search phase of the SLR process. In part two, we surveyed SLR authors in SE to identify barriers and requirements present in the SLR search phase. We combined the results from these two parts to generate a final list tooling requirements for the SLR search process to remove the most important barriers faced by SLR authors in SE (Article 2).

The goal of SSRT is to help automate the all or part of the SLR searching phase for SE SLRs. The initial version of SSRT runs on Windows. We plan to port it to other operating systems in the future. In building SSRT, we followed the design shown in Figure 4.2. This figure illustrates how the main components of SSRT work together to achieve the overall functionality. We used an object oriented development approach to allow for easier upgrades.

Figure 4.1 shows the initial screen of the SSRT user interface. From this screen, the user can search for literature, download the results, and store them in a separate database for extra screening and processing later. In choosing the specific features to implement in SSRT, we also considered the feasibility of implementation. We plan to add more features in later versions of SSRT.

The following subsections describe each of the main features in SSRT. For each feature we indicate why we chose to implement the feature and provide a table to map the requirements addressed by the feature to those listed in Table 4.1. These requirements come directly from our previous work (Article 1 and Article 2). There are a small number of requirements that do not map back to the table. Those requirements originated in the community workshops we conducted (Article 1). We chose not to implement all of the requirements. Some requirements could not be addressed by a third-party tool as they relate directly to features of the digital libraries. Other requirements we reserved for future versions of SSRT.

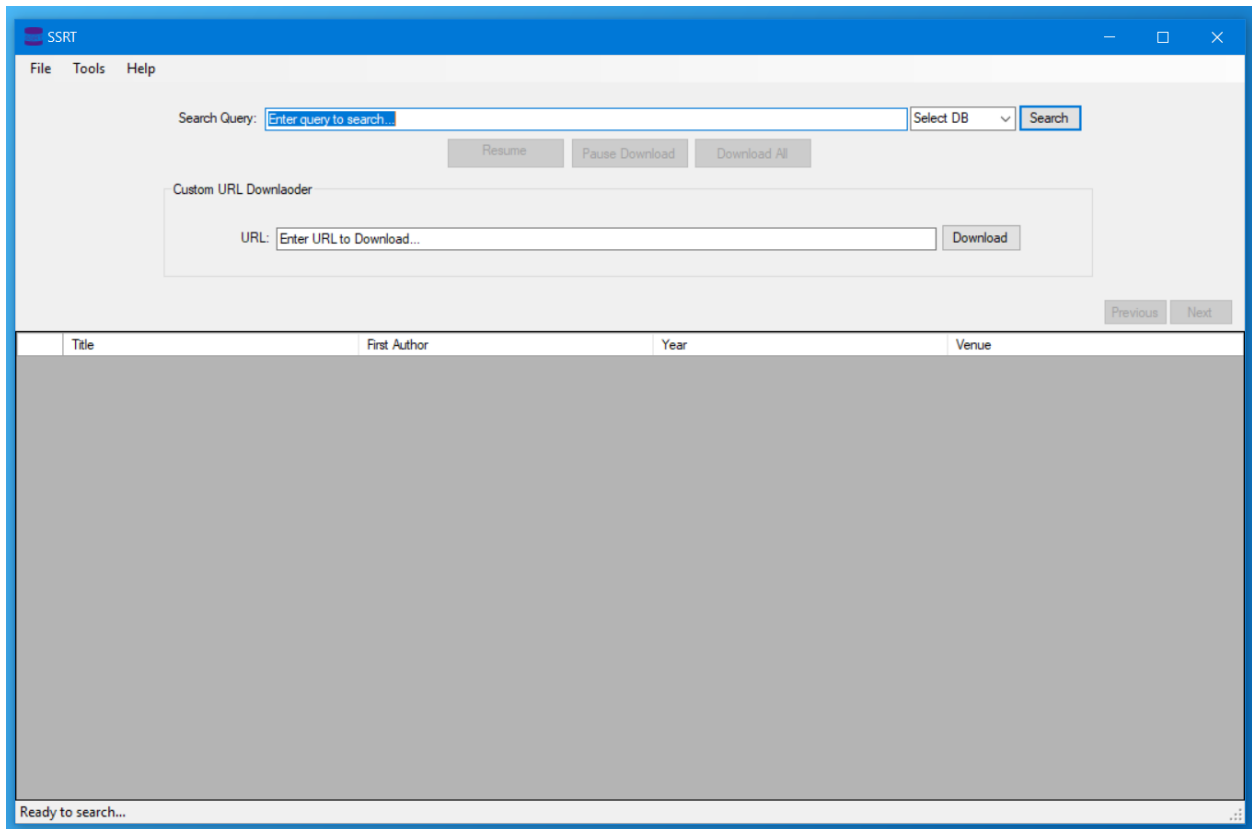


Figure 4.1: SSRT Main Screen

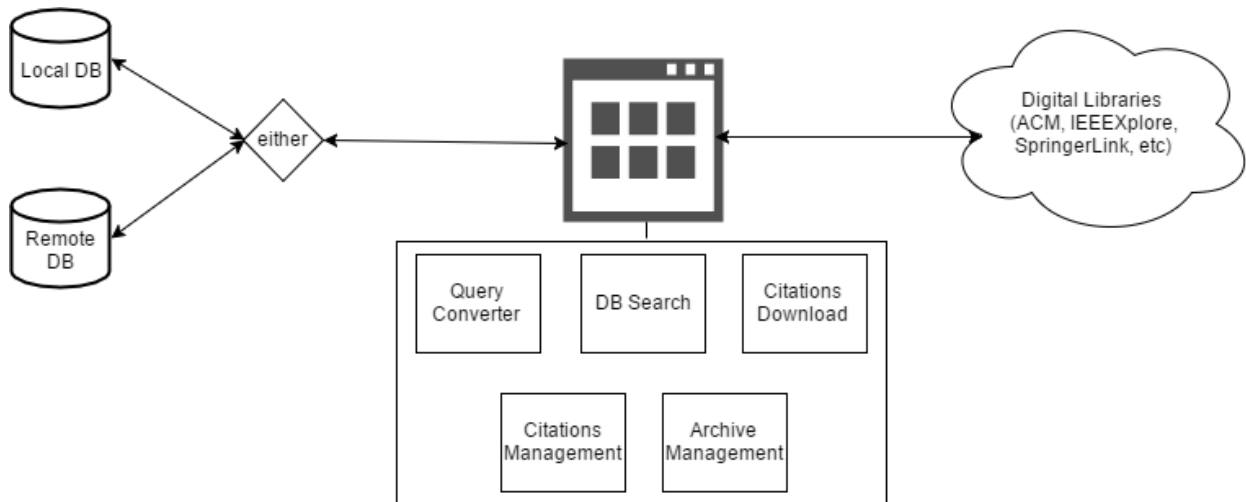


Figure 4.2: SSRT Design

4.3.1 Query Converter (QC)

This feature addresses the highest priority requirements given by the community. This issue has appeared in all of our previous work regarding SLR barriers (Article 1, Article

2). Table 4.2 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.2: QC Addressed Requirements

Requirement	Requirement	Source
R1	Search multiple databases in standardized query	Community Workshop, SLR, Survey
	Same digital library produce different results based on the order of the words	Community Workshop, SLR
	Limitation on the search string	SLR

The query converter helps users develop better search queries. Because databases have different formats and syntax, it is difficult to design a query that can run similarly on all databases. This feature allows the user to enter search query one time. SSRT then automatically adapts the syntax of the query to meet the rules of each target database. SSRT will also help the user construct more accurate queries by alerting the user of illegal characters and search string length limitation in each of the database. SSRT will also suggest how to divide the query into multiple queries when the length of the query exceeds the limit of the digital libraries. In future versions of SSRT, we will implement a “search term suggestion” feature to help users improve their queries.

Some databases provide different results depending upon the order of the search terms. The query converter will generate queries with all permutations of the search terms to ensure it obtains all the results from the digital libraries. Figure 4.3 shows the home screen of the query converter and how it would enhance an example query (Software Engineering and (“Quality” OR “Testing”) Not “Validation”). The output of the query converter is a list of search queries that are ready to be executed to have the maximum coverage of results.

4.3.2 Query Search (QS)

This feature addresses the top rated requirement. In addition, this feature complements the QC feature. This feature addresses multiple requirements. Table 4.3 shows the requirements and the sources of the requirements addressed by this feature.

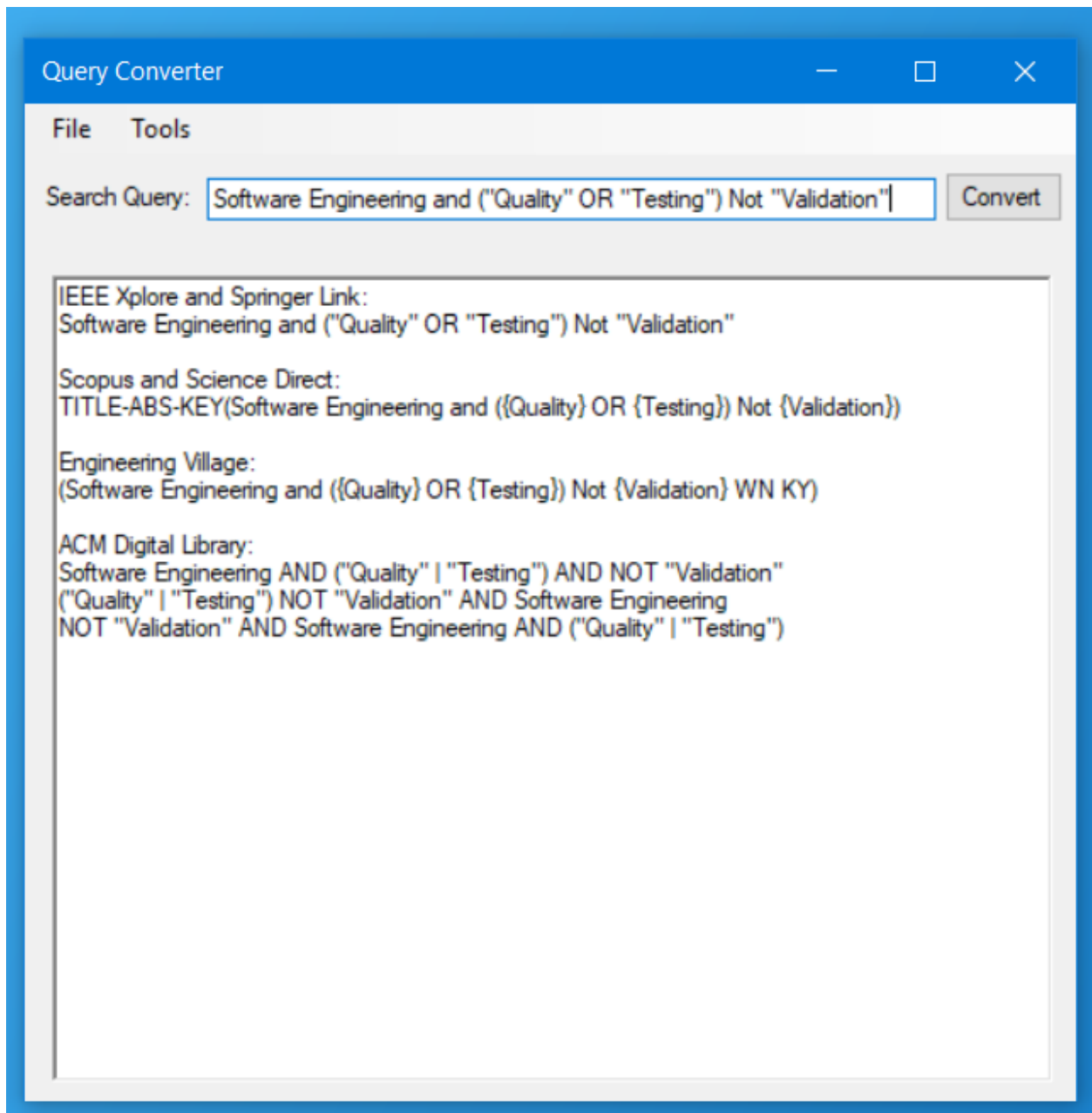


Figure 4.3: Query Converter Home Screen

Query search differs from the QC. While QC helps the user build the query, QS allows the user to execute a query that they have already formulated, that does not need any manipulation, or that was formulated by the QC. QS can be performed on one database or across all the databases. The query search displays the results in a built-in grid inside SSRT. This table or grid shows the results from the digital libraries as they appear. The

Table 4.3: QS Addressed Requirements

Requirement	Requirement	Source
	Support the ability pause/resume the search process	Community Workshop
R1	Search multiple databases in standardized query	Community Workshop, SLR, Survey
	Unified portal to search literature in common DL	Community Workshop, SLR
	Cross search of digital libraries	Community Workshop
	Support for search iterations	Community Workshop
	Support piloting search string	Community Workshop
	Better approaches to display the results from DL	SLR
R24	Support for searching process	Survey

current version supports the following databases: *IEEEExplore*, *ACM Digital Library*, and *SpringerLink*. We will add support for more databases in later versions of SSRT. Figure 4.3.2 illustrates how SSRT displays the results of the query to the user.

4.3.3 Downloading Results (DR)

This feature addresses another of the top requirements and complements the QC and QS features. Without the ability to automatically download the results, the QC and QS features would be less valuable. It also complements the feature of searching the digital libraries. Table 4.4 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.4: DR Addressed Requirements

Requirement	Requirement	Source
R8	Download papers	Community Workshop, SLR, Survey
	Difficult to integrate with digital libraries	Community Workshop, SLR

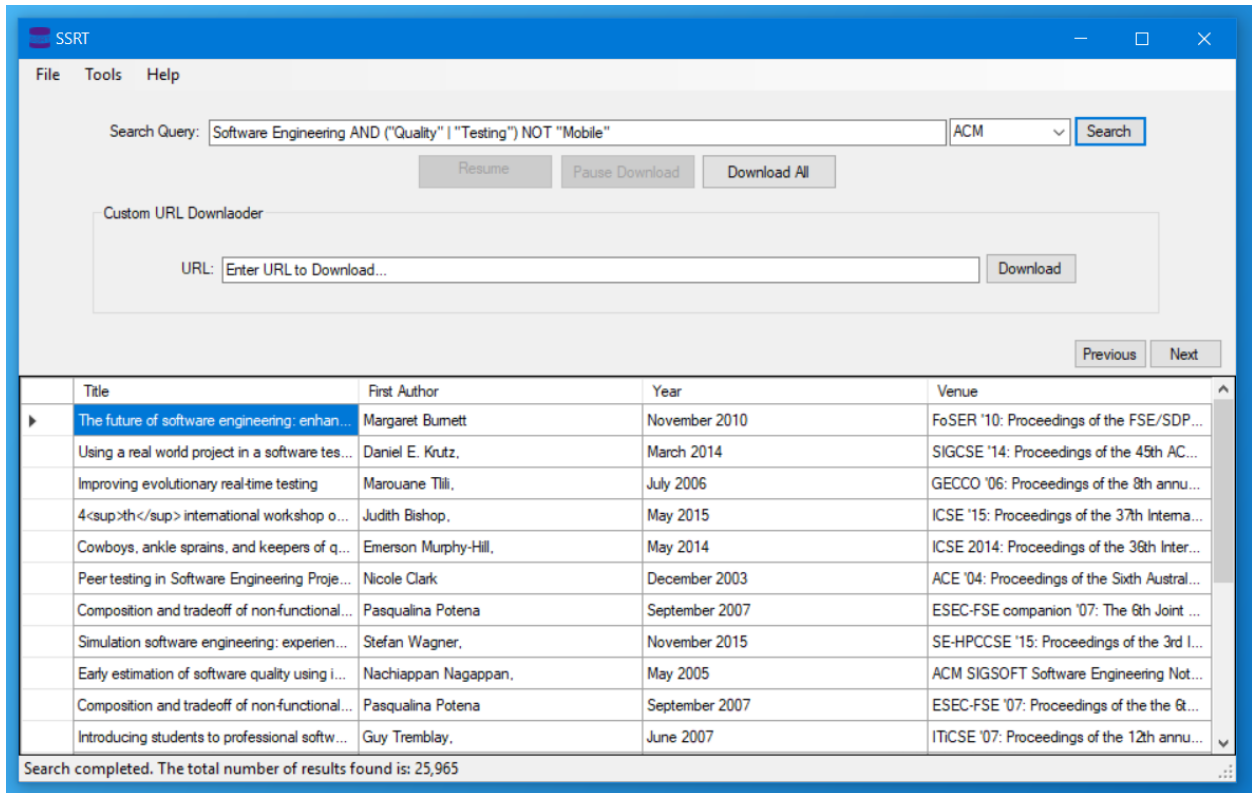


Figure 4.4: Results of Query Search

SSRT provides two options to download the results from the search of the digital libraries. Note that the results include only the citation information and not the actual PDFs, due to copyright issues. Figure 4.3.3 shows the interface to access the options described below:

- **Standard Query:** For this option, the user uses the regular query option in SSRT, searches for the results first, and then downloads all results. This option helps the user edit their search query as necessary before downloading the results from the digital libraries. The user can download from one database at a time or from all the databases together. Note that if the users chooses to download from all databases, they do not have the option to preview the results before downloading.
- **Custom URL:** To use this option, the users must already have a specific URL from one of the supported digital libraries and would like to obtain the results of this exact URL without any manipulation. The user can develop the URL via the interface of

the digital library to design the query, set filters, or any other options then copy/paste the URL in SSRT. This option is useful when the user wants to download the results from a digital library that does not allow batch downloading. The user does not need to go through the whole search process in SSRT. Instead, the user provides only the URL for that page and SSRT automatically downloads the results.

The download process for either option runs in multi-threaded manner which allows the user to search and download at the same time, as well as launching multiple downloads simultaneously.

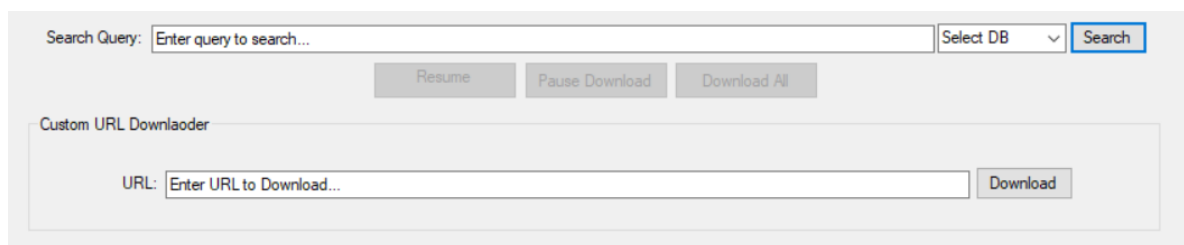


Figure 4.5: Download Options

4.3.4 Back-end Storage (BS)

We did not want to store the the individual .bib files directly on the user's local machine. Instead we store the citations and results in more advance approach to enable integration and standardization We wanted to build a backbone that will support both the current features of SSRT as well as future enhancements. Table 4.5 shows the requirements and the sources of the requirements addressed by this feature.

After downloading the results from the digital libraries, the citations are converted and stored in a MySQL database. SSRT stores results in a database to enable the capability to collaborate on the search between multiple users in different places. This database case reside either locally on the user's machine or on a remote server. SSRT has a connection setting file that provides information about the location of the database, which the user can easily update. The user can choose to use the default database setup on the SSRT server or use their own database. The users' own database can be on their local machine or on

Table 4.5: BS Addressed Requirements

Requirement	Requirement	Source
R12	Support for collaborative search	Community Workshop, SLR, Survey
R22	Archive search queries and search process	Community Workshop, Survey
	Better approaches to display the results from DL	SLR
R4	Merging results from DL	Survey
R6	Repository for papers	Survey
R20	Import search results to one tool	Survey
R37	Tracing studies sources	Survey
R38	Notification of new papers	Survey

a remote server for which the user has proper connection information, i.e. database name, server, username, and password.

Figure 4.6 shows the database design that sits behind the current version of SSRT. We will modify the database as needed when we add new features. The database also has information about the users, to enable collaboration and privacy for each search for each user.

4.3.5 Removing Duplicates (RD)

Beside being the second highest requirement on the list of requirements, this feature also showed up in the limitations of the current tools. Therefore, we chose to address it in this version of SSRT. Table 4.6 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.6: RD Addressed Requirements

Requirement	Requirement	Source
R2	Removing duplicates	Community Workshop, SLR, Survey

Duplicate results from digital libraries is an important problem faced by SLR users. By

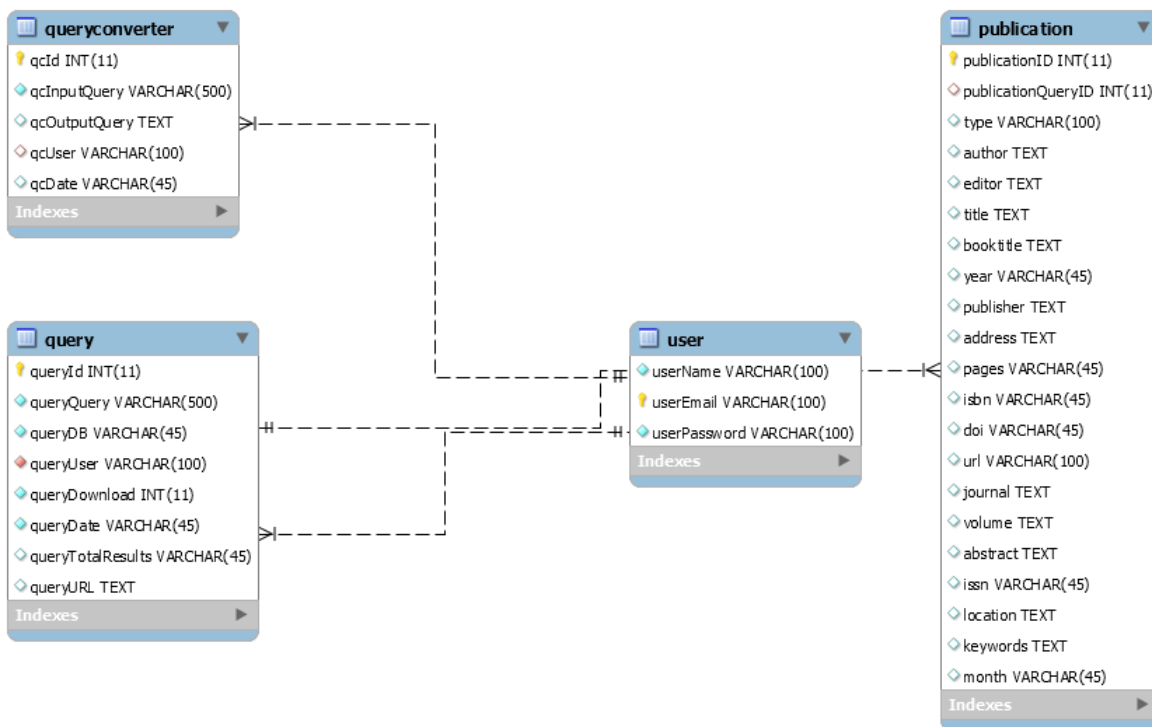


Figure 4.6: Database Design

default, SSRT eliminates duplicate results. After downloading the results from the digital library, SSRT stores the citation information into a MySQL database. The SSRT database is designed to automatically reject any citation that is already stored in the database. This step will reduce the work required for users to manually remove duplicates. SSRT provides the total number of results found and the number of results downloaded after removing duplicates, which makes it easy for the user to determine the percentage of duplicates from each digital library. This information will be very useful to the user while documenting their findings.

4.3.6 Import/Export (IE)

In order to enable SSRT to interact with existing tools that implement portions of the SLR process, SSRT needs to provide a way to import and export data. This feature addresses the issue of being locked-in to using particular tools that cannot interact with others. Table 4.7 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.7: IE Addressed Requirements

Requirement	Requirement	Source
	Export search results	Community Workshop
R7	Standardized Export	Community Workshop, Survey
	Ability to import external studies not from digital libraries	Community Workshop
	Importing search results from different databases	Community Workshop
R4	Merging results from DL	Survey
R20	Import search results to one tool	Survey
R34	Import external resources	Survey

Integration with other tool was a major aspect that we wanted to consider while designing SSRT. Thus, SSRT provides the ability to export and import results from other tools or digital libraries. For example, the user can import any citation in Bibtex format and still take advantage of the other features in SSRT, like removing duplicates automatically.

The results can be exported in a plain text format or in Bibtex format. Since SSRT stores all results in a MySQL database, it is very easy to add more export formats in future versions. Figure 4.7 shows the options for export results.

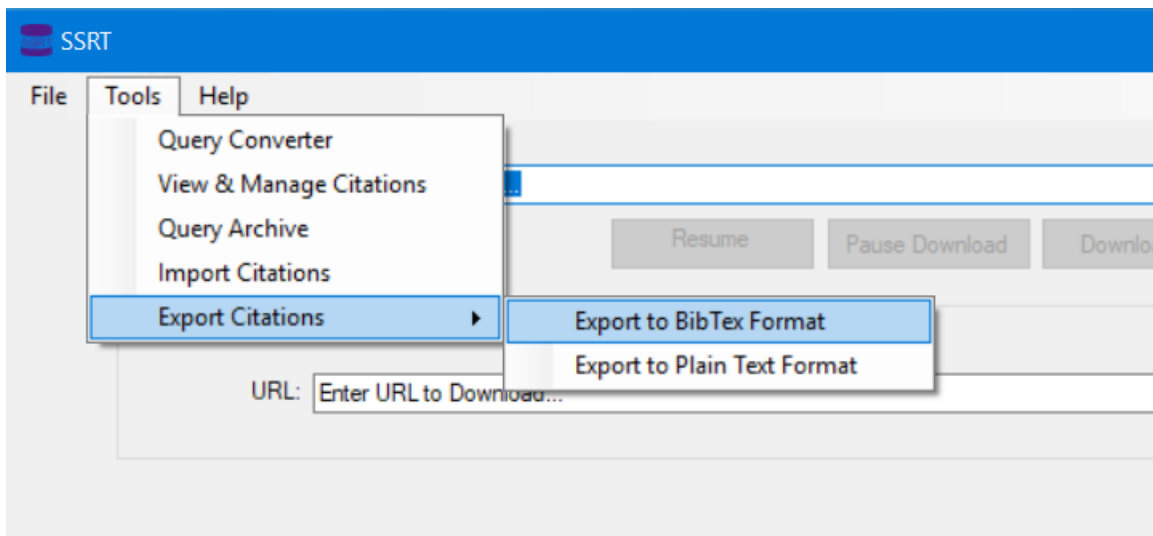


Figure 4.7: Export Options

4.3.7 Archive (AR)

Because the information is stored in a MySQL database, it makes developing an archive very easy. In addition, the SE community desired this capability. Therefore, we choose to implement this feature. Table 4.8 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.8: AR Addressed Requirements

Requirement	Requirement	Source
R22	Archive search queries and search process	Community Workshop, Survey

SSRT provides archive features across all the tool components. It tracks the queries that have been executed, the queries that have been downloaded, the number of results, and more information that can be useful for users while conducting SLRs. The query archive logs (1) the queries inserted and converted using QC, (2) the queries searched using SSRT, and (3) the external URL downloads. This feature makes it easier to identify the source of each citation, not only the digital library but also the specific query. The archive also keeps a log of the imported citations to enable the users to differentiate which citations originated from the search and which were imported from external sources. Figure 4.3.7 shows the screen shot of the archive component.

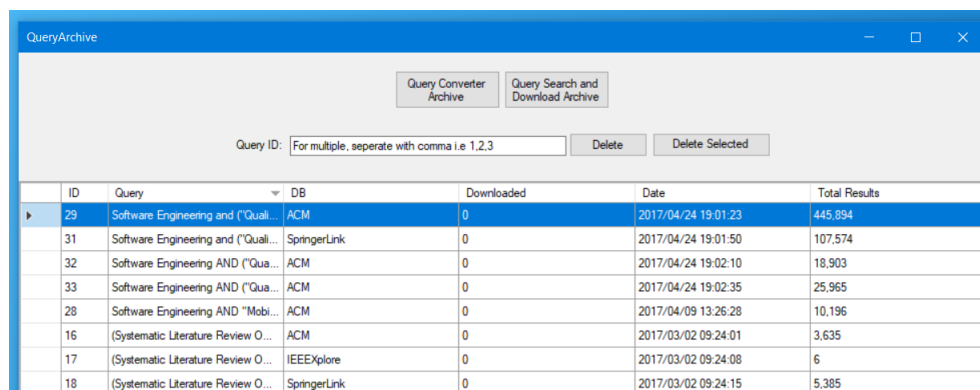
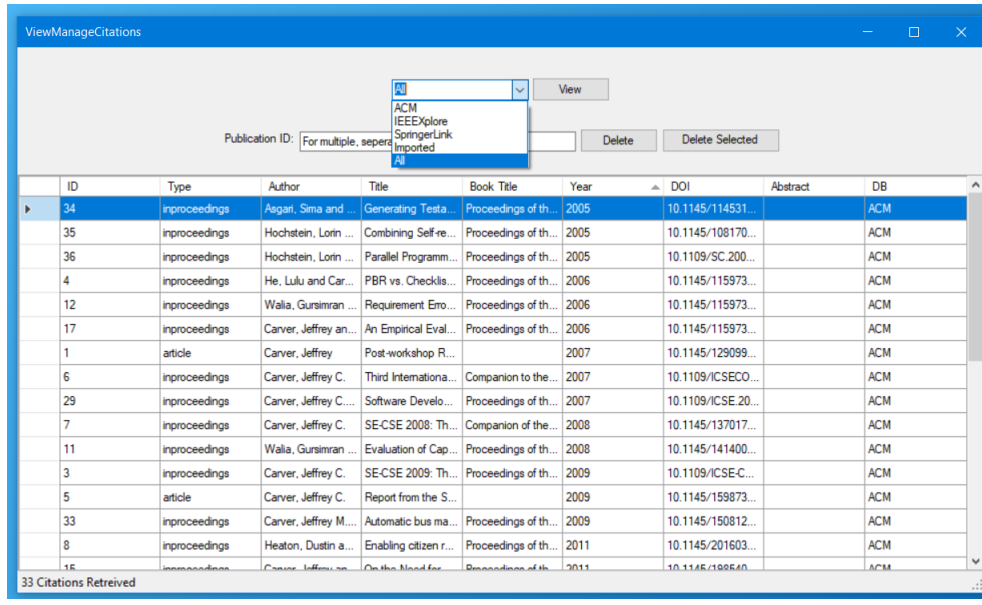


Figure 4.8: Archive Home Screen



1htb]

Figure 4.9: Citations Management Home Screen

4.3.8 Citations Management (CM)

Even though citation management tools exist, there were still requirements from the community for this feature. It is helpful for SSRT to allow the user to view and manage the citations before exporting them to other tools. Table 4.9 shows the requirements and the sources of the requirements addressed by this feature.

Table 4.9: CM Addressed Requirements

Requirement	Requirement	Source
	Better approaches to display the results from DL	SLR
R6	Repository for papers	Survey
R29	Reference management	Survey
R37	Tracing studies sources	Survey

To be able to use the information stored by SSRT, users must have a way to view that information. Therefore, we added a simple citation management module. This feature enables the user to view and/or the citations they downloaded or imported. The user can filter the view by the source of the citations, e.g. from one digital library or from all sources. Figure 4.3.8 shows the screen of citations management.

4.4 Evaluation

This section describes the three internal evaluations of SSRT. For each evaluation, I explain why I chose that evaluation, what benefits the evaluation provides, and the results of the evaluation.

4.4.1 Feature Analysis

Evaluating the tool using feature analysis provides an objective evaluation and reduce bias in the evaluation. It ensures that the tool is evaluated against concrete requirements which ensures the scope of the evaluation is fixed. Since it follows defined steps and process, it makes it easier to replicate as well.

The goal of this analysis is to determine whether our tool provides complete coverage of the SLR tool requirements provided by the SLR community in SE. The requirements that we are evaluating the tool against are the ones show in table 4.1 from our study on identifying requirements for the SLR search phase (article 2 will be cited once it is completed).

To perform such analysis, we chose the *feature analysis* method [39], which is one of the evaluation methods defined in the DESMET methodology used for evaluating tools and methods [36]. We chose DESMET because its goal is to report information about the suitability of a tool within a specific context rather than trying to rank the tool. DESMET identifies nine methods of evaluation and a set of criteria in order to help researchers select the appropriate method for their evaluation. The DESMET method provides detailed guidelines for each evaluation. In addition, DESMET is a well-established evaluation method in software engineering [23, 26, 42].

We attempted to perform the tasks described by each requirement for SSRT. In each case, we determined whether SSRT supported the requirement fully, partially, or not all. Then we generated a *total coverage score* for SSRT as follows:

- We weighted each requirements (from table 4.1) from 1-9 (based on the ordering). Requirements that are not tool requirements (e.g. requirements for the digital libraries)

have a weight of 0, because we did not evaluate them.

- We rated the level of coverage for each requirement as *fully*, *partially*, or *none*.
- Finally, for each rating we gave a score (fully - 100%, partially - 50%, none - 0%). We then computed total of the scores and divided by the total points available, and we gave a final score for the tool.

Table 4.10 shows the results of the analysis. Overall SSRT covers 48% of the requirements desired by the community. Among these requirements, SSRT addresses most of the highly rated requirements. And, of those requirements it addresses, SSRT fully addresses most of them. As we add more functionality to SSRT in later versions, we will continue to address additional requirements.

As a point of comparison, we evaluated all of the other SLR tools that have support for the SLR search phase using the same requirements. Table 4.11 shows the scores for all the other tools. Focusing only on those requirements for the SLR search phase, SSRT has a higher score than any of the other tools. This result is expected because these tools were built prior to the publication of our list of requirements. Some of the tools got zero score in visualization even though it is supported, however that is because the visualization support they have is not in the search phase it is in the data extraction part.

The results of this evaluation are encouraging in the fact that SSRT is the most comprehensive of the existing tools. Conversely, SSRT only covers about half of the requirements. In the first version of the tool, we focused on the requirements that were the most feasible for implementation. Based on the results of the feature analysis, we will increase the coverage of requirements by focusing on collaboration, visualization, and reporting in future versions of the tool.

4.4.2 Replicating an SLR

For this evaluation, we replicated the search phase of a previously conducted SLR. We chose to replicate our own SLR because we were familiar with the effort required to perform

Table 4.10: SSRT Feature Analysis Score

#	Requirement	Weight	Supported	Score
R1	Search multiple databases in standardized query	9	P	4.5
R2	Removing duplicates	8	F	8
R3	Filtering capabilities	7		
R4	Merging results from DL	6	F	6
R5	Synonym recommendation	6		
R6	Repository for papers	6	F	6
R7	Standardized export	6	F	6
R8	Download papers	6	F	6
R9	Automatic determination of relevance	5		
R10	Support snowballing	5		
R11	Automatic generation of search terms	5		
R12	Support collaboration	4	P	2
R13	Papers categorization	4		
R14	Iterative search	3	F	3
R15	Retrieve more relevant results	0	NA	
R16	Search string definition	0	NA	
R17	Support visualization and graphical analysis	3		
R18	Papers summarization	2		
R19	Managing notes	2		
R20	Import search results to one tool	2	F	2
R21	Validation of data	0	NA	
R22	Archiving	2	F	2
R23	Tagging	2		
R24	Support for searching process	2	P	1
R25	Aggregate data	2	F	2
R26	Extracting tables and figures	1		
R27	Version control	1	P	0.5
R28	Automated analysis	1		
R29	Reference management	1	F	1
R30	Meta data management	1		
R31	Search papers based on similarity	1		
R32	Directory for SE venues	1		
R33	Semantic enrichment	1		
R34	Import external resources	1	F	1
R35	Suggesting articles based on keywords	1		
R36	Easy to use	1	F	1
R37	Tracing studies sources	1	F	1
R38	Notification of new papers	1	F	1
R39	Other	0	NA	
	Total	110		54 - 50%

Table 4.11: SLR Tools Feature Analysis Scores

#	Weight	SESRA	Parsifal	StArt	SLuRP	SLRTOOL	SLR-Tool
R1	9	4.5	4.5				
R2	8						
R3	7						
R4	6	6	6	6	6	6	6
R5	6						
R6	6	6	6	6	6	6	6
R7	6						
R8	6	6	6				
R9	5	2.5	2.5	5	2.5	2.5	2.5
R10	5			5			
R11	5						
R12	4	4	4	2	4	4	
R13	4						
R14	3	3	3	3			
R15	0						
R16	0						
R17	3						
R18	2						
R19	2			2	2		
R20	2						
R21	0						
R22	2	2	2				
R23	2			2	2		
R24	2	1	1	1	1		
R25	2	2	2	2	2	2	
R26	1						
R27	1						
R28	1						
R29	1	1	1	1	1	1	1
R30	1						
R31	1						
R32	1						
R33	1						
R34	1	1	1	1	1	1	1
R35	1						
R36	1	1	1	1			
R37	1	1	1				
R38	1						
R39	0						
Total	110	41	41	37	28	23	17
Percentage	100	38	38	33	25	20	15

the search manually. This knowledge gave us a basis for comparing SSRT's performance. This evaluation is more informal in that it helps us understand how SSRT performs in a real SLR setting. It should reveal any deficiencies apparent from running the tool in a practical scenario that may not have been evident during the development process.

We evaluated SSRT's performance on the following dimensions:

- **Time:** The amount of time required to search, download, and remove duplicates from results.
- **Ease of Use:** The number of steps taken or the amount of manual work.
- **Coverage:** The ratio of relevant results found using SSRT to those found using the manual process.
- **Duplicates:** The number of duplicates in the results after download.

We chose the SLR to identify the SLR search barriers (Article 2 in this dissertation) as the SLR for replication. The search process for that article included the following steps:

- We constructed the two search strings shown in Table 4.12. The goal of the first search string was to identify papers that describe problems in the SLR search phase (either in SLR papers or in SLR meta-papers). The goal of the second search string is to identify papers related to SLR tools.
- We executed these queries on the set of digital libraries which are commonly used across SLRs, including: ACM Digital Library, IEEEXplore, ScienceDirect, Scopus, and Engineering Village. We chose this set of digital libraries, because they are the ones most commonly used in SLRs and they index high quality journals and conferences. Also each of these digital libraries has the advanced search capabilities to properly execute the queries.

We had to manually covert each search string into the proper format for each digital library, a time-consuming process that required reading the documentation for each library.

Table 4.12: Search Strings

SQ#	Search String	Motivation
SQ1	(Systematic Literature Review OR Systematic Review OR SLR OR Systematic Mapping) AND (Software Engineering) AND (Searching OR Automated Search OR Search OR Digital Library) AND (Problem OR Barrier OR Issue OR Answer OR Solution)	Searching problems in published SLR papers or SLR meta-papers
SQ2	(Systematic Literature Review OR Systematic Review OR SLR OR Systematic Mapping) AND (Software Engineering) AND (Tool OR Tooling OR Tools)	SLR tool papers

The we manually executed each search string on each digital library. The process of retrieving the citations required a number of manual steps because some digital libraries do not allow for batch downloading or the use of tools like Zotero. We also noticed that not all digital libraries exported the citations in BibTeX format. Finally, we manually removed duplicate results by using Excel and EndNotes.

We replicated this process using SSRT. Using the same search strings (Table 4.12), we repeated the process of converting the string for each library, searching the libraries, downloading the results, and removing duplicate entries. The current version of SSRT only supports three digital libraries (ACM, IEEEExplore, and SpringerLink). Therefore, we only executed the search on those libraries.

Our comparison of SSRT and the manual process showed the following results for each of the evaluation factors defined above:

- *Time*: The manual search process took two days, including reading the library documentations and converting the queries for each digital library. Conversely, the Query Converter feature of SSRT required only a few seconds to perform this task. Note that Query Converter does not convert the “near” operation because it is not supported by all digital libraries. In addition, the download time was faster using SSRT. Using the manual process, we had to download the results either page-by-page or one-by-one,

which took approximately 3 days to get the list of citations. Using SSRT required just one click. We provided the link for the results and the results were downloaded within 10 minutes automatically.

- *Ease of Use*: While interacting directly with the digital libraries is not difficult, it does require a large amount of manual work. It is much easier to use SSRT's automatic features, which require only logging in and providing the URL to download the references.
- *Coverage*: Because SSRT uses the same digital library interface as the manual process, the number of results return for each query is identical. However, when SSRT generates the search strings, it generates additional permutations of the search terms. Because ACM gives different results based on the order of the search terms, SSRT provided more results in ACM digital library. While, manually entering all of the queries generated by SSRT would produce the same results, the fact that SSRT performed this task manually means that it may result in additional papers that could be missed if the author does not perform all permutations manually.
- *Duplicates*: The number of results returned using SSRT is the same number of results the digital libraries return. However, when we downloaded the results from SSRT, the results were already cleaned from duplicates. When we downloaded 220 citations, there were only 4 duplicates that SSRT missed. SSRT eliminates duplicates that matches on "authors", "year", and "title". If there is any variation, then SSRT does not eliminate the duplicate. While SSRT could enforce stronger rules for removing duplicates, we think it is safer to err towards leaving a small number of duplicates compared with accidentally excluding a relevant paper that is not a duplicate. SSRT cut down the manual work and the time required to find the duplicates and remove them, as the tool automatically reject duplicates from digital libraries when they are downloaded.

This evaluation reveled some issues with SSRT that we will address in future versions.

First, because SSRT is multi-threaded, it does not always clearly report overall progress of the task. The user really only knows when the task is finished. In addition, in some cases SSRT does not kill all threads when the user closes the tool. Second, because SSRT missed some duplicates, we will further investigate how to improve this process without accidentally removing non-duplicates. Finally, there is some delay in deletion citations from the database when it is in remote mode and not local database, and we plan to tackle this issue in the future.

4.4.3 Evaluation by Other Users

The third evaluation was to have people other than the tool developers who were familiar with the SLR process use the tool and provide feedback. By having additional people evaluate the tool, we hoped to eliminate any bias that was present from our own evaluations. This external perspective was useful for identifying deficiencies in SSRT as well as suggestions for improvement. In addition, we gathered information about the time they took, the number of results found, the number of duplicates, and the number of errors encountered.

Based on our previous evaluations, we pose the following hypotheses about the use of SSRT:

- Authors will spend less time to perform the SLR search phase activities.
- SSRT will reduce the manual effort required to perform the SLR search phase activities.
- The results provided by SSRT will have fewer duplicates.

The participants provided their informal feedback after they tested out the tool in simulating an SLR search phase. We had one PhD student and two researchers who were all conducting SLRs at the time. The feedback primarily highlighted desired enhancements for SSRT and typos in the user interface. One of the participants indicated that the application froze and stopped working while he was working on exporting citations but then he indicated there was a message box that he did not notice it required his input.

The participants indicated that SSRT helped them download multiple batches of results together and combine results from different databases easily. One participant who downloaded 39000 citations from the ACM Digital Library indicated that it took between an hour to and hour and half to download that amount of results. However the export feature was very slow for the 39000 especially since he was using a remote database and not a local one.

Based on this evaluation, the users encountered a few issues that we did not consider. But overall, we received positive feedback from the participants especially for enabling multiple downloads from different digital libraries. We plan to fix these issues in the next version of SSRT. For example, due to the physical location of messages, the users might have missed them, resulting in the user running another operation and causing the tool to crash for missing user input. The users also desired to have progress bars during the download process. Finally, the users would like to have support for more digital libraries.

4.5 Threats to Validity

There are limitations and threats in the design of SSRT. We might have interpreted some of the requirements differently from how the community intended them. However, because two researchers worked together to analyze and extract the meanings of the requirements, the risk of misinterpretation is reduced. In addition, while we tested all of the features of SSRT, we have not done a 100% coverage test suite yet. Thus, even though I tested out all the functionalities, the coverage of line of codes might have not been 100%. There could be additional defects that we have yet to uncover. We depend on the feedback of the users from using the tool, and if there are things that did not look right, the users have reported these issues to the developers, and they have been logged, and fixed.

There are also threats to validity of the SSRT evaluation. We performed all evaluations internally. The lack of external validation introduces the following threats: A) Because we both developed and evaluated SSRT, there is the potential for bias in the evaluation. To reduce the bias, we used the feature analysis methodology to provide some level of objectivity.

B) We have not yet performed a formal, external validation of SSRT. We plan to do this validation in our future work.

4.6 Conclusion and Future Work

SLRs are beneficial for the SE community. They help researchers produce higher quality results and enable readers to get information on issues and topics faster and with more confidence. However, the SLR process is a difficult and time-consuming task. Specifically, the searching phase of the SLR process is one of the most difficult and time-consuming. Proper tool support can address most of the issues that arise the SLR search phase.

In our previous work, we developed a set of requirements for SLR tool support. Based on those requirements, we built SSRT, a tool to help SE researchers in the SLR search phase. SSRT automates multiple tasks in the SLR search phase and reduces the manual work for the author. Some of those tasks includes converting the query to match each digital library's requirements, searching different digital libraries, downloading results, and eliminating duplicates in the results.

We have also conducted informal evaluations of SSRT. Based on our evaluations, we can conclude that SSRT solves some of the problems the SE community reported. It also reduced the time and manual work in the search phase of conducting SLRs.

In our future work, we plan to enhance SSRT to address more of the requirements. Based on the evaluations, there are a number of features still needed. We built SSRT so that it would be easy to add new features. We will specifically focus on features related to collaboration and graphic visualization. We also plan to conduct a more formal evaluation of SSRT, by having external authors use it to perform their SLRs.

Finally, we plan to release SSRT as an open-source tool. This release will allow other developers to enhance it based either upon the requirements that we have defined in our studies or upon other requirements specifically interesting to the developers. However, the main goal of the tool will stay the same that is enhancing the SLR search phase.

CHAPTER 5

DISSERTATION CONCLUSION AND PUBLICATIONS

SLRs are very useful to the SE community as evidenced by the increasing number published in SE venues. Even with this increase, the SLR process is difficult and time-consuming due to a number of barriers and to the large amount of manual work required. One of the most difficulty and costly stages in the SLR process is the search phase in which authors identify and retrieve the papers required to conduct the review.

The issues faced by authors during the SLR search phase result from multiple factors. Some of these factors could be improved by tool support, while others are the result of problems with the digital libraries or with the authors themselves. Therefore, the goal of this dissertation is to **understand and document the barriers related to the SLR search phase and build and evaluate a solution to overcome those barriers**. The dissertation consists of three articles that achieve the dissertation goal.

Article 1 describes the results of three empirical studies focused on identifying SLR process barriers and tool requirements. This article provided a prioritized list of functional requirements for SLR tools based on input from SLR authors within the SE community. This list of requirements covers all phases of the SLR process, not just the search phase. Using this list of requirements, the article also provides a snapshot of the current status of tool support for the SLR process. This list of requirements and the results of the tool evaluations can be used as reference for SLR authors, researchers, and developers who wish to build SLR tools.

Based on the gaps between the current tools and the requirements for SLR searching,

Article 2 sought to better understand the specific needs for SLR search. This article describes two studies: an SLR to identify the search phase barriers in the SLR and a survey of SLR authors to identify barriers faced in practice. The results of these studies provide: (1) an understanding of the barriers in the SLR search phase, (2) a list of tooling requirements for SLR search based on the community input, and (3) a summary of the tools authors currently use for searching, along with the advantages and disadvantages of each tool.

Finally, Article 3 describes the construction and evaluation of SSRT, a tool to support SLR search, based upon the output of the first and the second articles. SSRT enables authors to use one interface to search multiple databases using standardized queries, download the results in batch mode, and automatically eliminate duplicates. This article also describes an informal evaluation of SSRT. The results of the evaluation show that SSRT does reduce the manual work required in the SLR search phase. The tool is still in its first version and we will make enhancements in the future versions.

Once again, the overall and main contributions of this dissertation are:

- Provides empirical evidence about the existence of and the importance of problems in the SLR search phase.
- A list of requirements to be used as a reference for SLR authors, researchers, and developers who want to build tools.
- Enables SLR authors to be more efficient by automating some of the manual work required during the search phase.
- Improves the quality of SLR by improving the quality of the search process through automation.

REFERENCES

- [1] Muhammad Ali Babar and He Zhang. Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *3rd International Symposium on Empirical Software Engineering and Measurement.*, pages 346–355. IEEE Computer Society, 2009.
- [2] Deepika Badampudi, Claes Wohlin, and Kai Petersen. Experiences from using snowballing and database searches in systematic literature studies. In *19th International Conference on Evaluation and Assessment in Software Engineering*, page 17. ACM, 2015.
- [3] Souvik Barat, Tony Clark, Balbir Barn, and Vinay Kulkarni. A model-based approach to systematic review of research literature. In *Proceedings of the 10th Innovations in Software Engineering Conference*, pages 15–25. ACM, 2017.
- [4] Balbir S. Barn, Franco Raimondi, Lalith Athappian, and Tony Clark. Slrtool: A tool to support collaborative systematic literature reviews. In *Proceedings of the 16th International Conference on Enterprise Information Systems - Volume 2: ICEIS.*, pages 440–447, 2014.
- [5] Jorge Biolchini, P Gomes Mian, Ana Candida Cruz Natali, and Guilherme Horta Travassos. Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, 679(05):45, 2005.
- [6] David Bowes, Tracy Hall, and Sarah Beecham. Slurp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In *Proceedings of the 2nd International Workshop on Evidential Assessment of software technologies*, pages 33–36. ACM, 2012.
- [7] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571–583, 2007.

- [8] Jeffrey C Carver, Edgar Hassler, Elis Hernandez, and Nicholas A Kraft. Identifying barriers to the systematic literature review process. In *7th International Symposium on Empirical Software Engineering and Measurement*, pages 203–212. IEEE, 2013.
- [9] Archibald Leman Cochrane and Rock Carling Fellowship. *Effectiveness and efficiency: random reflections on health services*, volume 900574178. Nuffield Provincial Hospitals Trust London, 1972.
- [10] Juliet Corbin, Anselm Strauss, et al. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks CA: SAGE, 2008.
- [11] Fabio Da Silva, A. Santos, Sérgio Soares, A França, Cleiton Monteiro, and Felipe Maciel. Six years of systematic literature reviews in software engineering: An updated tertiary study. *Journal of Information and Software Technology*, 53(9):899–913, 2011.
- [12] Jorge Calmon de Almeida Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, Tayana Uchôa Conte, and Guilherme Horta Travassos. Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, 21(2):133–151, 2007.
- [13] A.L. Delbecq, A.H. Van de Ven, and D.H. Gustafson. *Group techniques for program planning: a guide to nominal group and Delphi processes*. Management applications series. Scott, Foresman, 1975.
- [14] Oscar Dieste, Anna Grimán, and Natalia Juristo. Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, 14(5):513–539, 2009.
- [15] Oscar Dieste and Anna Griman Padua. Developing search strategies for detecting relevant experiments for systematic reviews. In *First International Symposium on Empirical Software Engineering and Measurement*, pages 215–224. IEEE, 2007.
- [16] Tore Dyba, Torgeir Dingsoyr, and Geir K Hanssen. Applying systematic reviews to diverse study types: An experience report. In *1st International Symposium on Empirical Software Engineering and Measurement*, pages 225–234. IEEE, 2007.
- [17] Emelie Engström, Mats Skoglund, and Per Runeson. Empirical evaluations of regression test selection techniques: a systematic review. In *2nd International Symposium on Empirical Software Engineering and Measurement*, pages 22–31. ACM, 2008.

- [18] Sandra Fabbri, Cleiton Silva, Elis Hernandez, Fábio Octaviano, André Di Thommazo, and Anderson Belgamo. Improvements in the start tool to better support the systematic review process. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, page 21. ACM, 2016.
- [19] Katia Romero Felizardo, Emilia Mendes, Marcos Kalinowski, Érica Ferreira Souza, and Nandamudi L Vijaykumar. Using forward snowballing to update systematic reviews in software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 53. ACM, 2016.
- [20] Ana M Fernández-Sáez, Marcela Genero Bocco, and Francisco P Romero. Slr-tool: A tool for performing systematic literature reviews. In *International Conference on Software Technology (2)*, pages 157–166, 2010.
- [21] Vitor Freitas. Parsifal systematic literature review tool. <https://parsif.al/>. [Online; accessed 1-December-2016].
- [22] Barney G Glaser, Anselm L Strauss, and Elizabeth Strutzel. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, 17(4):364, 1968.
- [23] Anna Grimán, Maria Pérez, L Mendoza, and Francisca Losavio. Feature analysis for architectural evaluation methods. *Journal of Systems and Software*, 79(6):871–888, 2006.
- [24] Edgar Hassler, Jeffrey C Carver, David Hale, and Ahmed Al-Zubidy. Identification of SLR tool needs—results of a community workshop. *Journal of Information and Software Technology*, 70:122–129, 2016.
- [25] Edgar Hassler, Jeffrey C Carver, Nicholas A Kraft, and David Hale. Outcomes of a community workshop to identify and rank barriers to the systematic literature review process. In *18th International Conference on Evaluation & Assessment in Software Engineering*, page 31. ACM, 2014.
- [26] Henrik Hedberg and Jouni Lappalainen. A preliminary evaluation of software inspection tools, with the desmet method. In *Fifth International Conference on Quality Software*, pages 45–52. IEEE, 2005.
- [27] Elis Hernandez, Augusto Zamboni, Sandra Fabbri, and André Di Thommazo. Using gqm and tam to evaluate start-a tool that supports systematic review. *CLEI Electronic Journal*, 15(1):3–3, 2012.

- [28] Salma Imtiaz, Muneera Bano, Naveed Ikram, and Mahmood Niazi. A tertiary study: experiences of conducting systematic literature reviews in software engineering. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, pages 177–182. ACM, 2013.
- [29] Samireh Jalali and Claes Wohlin. Systematic literature studies: database searches vs. backward snowballing. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 29–38. ACM, 2012.
- [30] Biren B Kamdar, Pooja A Shah, Sruthi Sakamuri, Bharat S Kamdar, and Jiwon Oh. A novel search builder to expedite search strategies for systematic reviews. *International Journal of Technology Assessment in Health Care*, 31(1-2):51–53, 2015.
- [31] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele University*, 33(2004):1–26, 2004.
- [32] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and Software Technology*, 55(12):2049–2075, December 2013.
- [33] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and Software Technology*, 55(12):2049–2075, 2013.
- [34] Barbara Kitchenham, Pearl Brereton, Mark Turner, Mahmood Niazi, Stephen Linkman, Rialette Pretorius, and David Budgen. The impact of limited search procedures for systematic literature reviews a participant-observer case study. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 336–345. IEEE Computer Society, 2009.
- [35] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, Keel University, 2007.
- [36] Barbara Kitchenham, Stephen Linkman, and David Law. Desmet: a methodology for evaluating software engineering methods and tools. *Computing & Control Engineering Journal*, 8(3):120–126, 1997.

- [37] Barbara Kitchenham, Rialette Pretorius, David Budgen, O Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. Systematic literature reviews in software engineering—a tertiary study. *Information and Software Technology*, 52(8):792–805, 2010.
- [38] Barbara A Kitchenham, Tore Dyba, and Magne Jorgensen. Evidence-based software engineering. In *26th International Conference on Software Engineering*, pages 273–281. IEEE Computer Society, 2004.
- [39] Barbara Ann Kitchenham. Evaluating software engineering methods and tools, part 7: planning feature analysis evaluation. *ACM SIGSOFT Software Engineering Notes*, 22(4):21–24, 1997.
- [40] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. *Evidence-Based Software Engineering and Systematic Reviews*, volume 4. CRC Press, 2015.
- [41] Christopher Marshall and Pearl Brereton. Tools to support systematic literature reviews in software engineering: A mapping study. In *7th International Symposium on Empirical Software Engineering and Measurement*, pages 296–299. IEEE, 2013.
- [42] Christopher Marshall, Pearl Brereton, and Barbara Kitchenham. 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*, page 13. ACM, 2014.
- [43] Jefferson Seide Molléri and Fabiane Barreto Vavassori Benitti. Sesra: a web-based automated tool to support the systematic literature review process. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, page 24. ACM, 2015.
- [44] Jefferson Seide Molléri, Luiz Eduardo da Silva, and Fabiane Barreto Vavassori Benitti. Proposal of an automated approach to support the systematic review of literature process (s). In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, pages 488–493, 2013.
- [45] Birgit Penzenstadler, Ankita Raturi, Debra Richardson, Coral Calero, Henning Femmer, and Xavier Franch. Systematic mapping study on software engineering for sustainability (se4s). In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 14. ACM, 2014.

- [46] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *International Conference on Evaluation & Assessment in Software Engineering*, volume 8, pages 68–77, 2008.
- [47] Heri Ramampiaro, Daniela Cruzes, Reidar Conradi, and Manoel Mendona. Supporting evidence-based software engineering with collaborative information retrieval. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*, pages 1–5. IEEE, 2010.
- [48] Mehwish Riaz, Muhammad Sulayman, Norsaremah Salleh, and Emilia Mendes. Experiences conducting systematic reviews from novices’ perspective. In *14th International Conference on Evaluation and Assessment in Software Engineering*, 2010.
- [49] David L Sackett, William M C Rosenberg, J A Muir Gray, R Brian Haynes, and W Scott Richardson. Evidence based medicine: what it is and what it isn’t. *BMJ*, 312(7023):71–72, 1996.
- [50] Dag IK Sjoberg, Tore Dyba, and Magne Jorgensen. The future of empirical methods in software engineering research. In *Future of Software Engineering, 2007. FOSE’07*, pages 358–378. IEEE, 2007.
- [51] Mats Skoglund and Per Runeson. Reference-based search strategies in systematic reviews. In *International Conference on Evaluation & Assessment in Software Engineering*, 2009.
- [52] Irena Spasic, Sophia Ananiadou, John McNaught, and Anand Kumar. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in bioinformatics*, 6(3):239–251, 2005.
- [53] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *18th International Conference on Evaluation and Assessment in Software Engineering*, page 38. ACM, 2014.
- [54] Claes Wohlin. Writing for synthesis of evidence in empirical software engineering. In *8th International Symposium on Empirical Software Engineering and Measurement*, page 46. ACM, 2014.
- [55] Claes Wohlin. Second-generation systematic literature studies using snowballing. In *20th International Conference on Evaluation and Assessment in Software Engineering*, page 15. ACM, 2016.

- [56] He Zhang, Muhammad Ali Babar, Xu Bai, Juan Li, and Liguo Huang. An empirical assessment of a systematic search process for systematic reviews. In *15th International Conference on Evaluation and Assessment in Software Engineering*, pages 56–65. IET, 2011.
- [57] He Zhang, Muhammad Ali Babar, and Paolo Tell. Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637, 2011.

APPENDIX A

INCLUDED PAPERS IN THE SLR

List of papers included in the SLR shown in Table A.1.

Table A.1: List of papers included in the SLR

Study	Reference
S1	Li, Z., Liang, P., & Avgeriou, P. (2013). Application of knowledge-based approaches in software architecture: A systematic mapping study. <i>Information and Software Technology</i> , 55(5), 777-794.
S2	Manuel D. D. D. ., Walquiria C. B. L., Eric B. P. M. (2014). An Approach and a Tool for Systematic Review Research. ICSEA 2014: The Ninth International Conference on Software Engineering Advances.
S3	Mehmood, A., & Jawawi, D. N. (2013). Aspect-oriented model-driven code generation: A systematic mapping study. <i>Information and Software Technology</i> , 55(2), 395-411.

Table A.1: List of papers included in the SLR

Study	Reference
S4	Khan, S. U., Niazi, M., & Ahmad, R. (2011). Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. <i>Information and Software Technology</i> , 53(7), 693-706.
S5	Cavalcanti, Y. C., MotaSilveira Neto, P. A., Machado, I. D. C., Vale, T. F., Almeida, E. S., & Meira, S. R. D. L. (2014). Challenges and opportunities for software change request repositories: a systematic mapping study. <i>Journal of Software: Evolution and Process</i> , 26(7), 620-653.
S6	da Silva, F. Q., Costa, C., Franca, A. C. C., & Prikladnicki, R. (2010, August). Challenges and solutions in distributed software development project management: a systematic literature review. In <i>Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on</i> (pp. 87-96). IEEE.
S7	Dieste, O., Grimm, A., & Juristo, N. (2009). Developing search strategies for detecting relevant experiments. <i>Empirical Software Engineering</i> , 14(5), 513-539.

Table A.1: List of papers included in the SLR

Study	Reference
S8	Turner, M., Kitchenham, B., Brereton, P., Charters, S., & Budgen, D. (2010). Does the technology acceptance model predict actual use? A systematic literature review. <i>Information and Software Technology</i> , 52(5), 463-479.
S9	Kitchenham, B., Brereton, P., & Budgen, D. (2010, May). The educational value of mapping studies of software engineering literature. In <i>Software Engineering, 2010 ACM/IEEE 32nd International Conference on</i> (Vol. 1, pp. 589-598). IEEE.
S10	Fernandez-Sez, A. M., Genero, M., & Chaudron, M. R. (2013). Empirical studies concerning the maintenance of UML diagrams and their use in the maintenance of code: A systematic mapping study. <i>Information and Software Technology</i> , 55(7), 1119-1142.
S11	Rievis, K., & Torkar, R. (2013). Equality in cumulative voting: A systematic review with an improvement proposal. <i>Information and Software Technology</i> , 55(2), 267-287.
S12	Niazi, M., Ikram, N., Bano, M., Imtiaz, S., & Khan, S. U. (2013). Establishing trust in offshore software outsourcing relationships: an exploratory study using a systematic literature review. <i>Software, IET</i> , 7(5), 283-293.

Table A.1: List of papers included in the SLR

Study	Reference
S13	Silva, F. Q., Prikladnicki, R., Frana, A. C. C., Monteiro, C. V., Costa, C., & Rocha, R. (2012). An evidence-based model of distributed software development project management: results from a systematic mapping study. <i>Journal of software: Evolution and Process</i> , 24(6), 625-642.
S14	Staples, M., & Niazi, M. (2007). Experiences using systematic review guidelines. <i>Journal of Systems and Software</i> , 80(9), 1425-1437.
S15	Bakar, N. H., Kasirun, Z.M., & Salleh, N. (2015). Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review. <i>Journal of Systems and Software</i> , 106, 132-149.
S16	Gonzalez, C. A., & Cabot, J. (2014). Formal verification of static software models in MDE: A systematic review. <i>Information and Software Technology</i> , 56(8), 821-838.
S17	Hassler, E., Carver, J.C., Hale, D., & Al-Zubidy, A. (2016). Identification of SLR tool needs results of a community workshop. <i>Information and Software Technology</i> , 70, 122-129.

Table A.1: List of papers included in the SLR

Study	Reference
S18	Carver, J. C., Hassler,E., Hernandes, E., & Kraft, N. A. (2013, October). Identifying barriersto the systematic literature review process. In EmpiricalSoftware Engineering and Measurement, 2013 ACM/IEEE International Symposiumon (pp. 203-212). IEEE.
S19	Al Dallal, J. (2015).Identifying refactoring opportunities in object-oriented code: A systematicliterature review. Information and softwareTechnology,58, 231-249.
S20	Zhang, H., Babar, M. A.,& Tell, P. (2011). Identifying relevant studies in softwareengineering. Information and Software Technology, 53(6),625-637.
S21	Kitchenham, B., Brereton,P., Turner, M., Niazi, M., Linkman, S., Pretorius, R., & Budgen, D.(2009, October). The impact of limited search procedures for systematicliterature reviews A participant-observer case study. InProceedingsof the 2009 3rd International Symposium on Empirical Software Engineering andMeasurement (pp. 336-345). IEEE ComputerSociety.
S22	Zarour, M., Abran, A.,Desharnais, J. M., & Alarifi, A. (2015). An investigation into the bestpractices for the successful design and implementation of lightweightsoftware process assessment methods: A systematic literature review. Journal of Systems and Software, 101, 180-192.

Table A.1: List of papers included in the SLR

Study	Reference
S23	Nidhra, S., Yanamadala,M., Afzal, W., & Torkar, R. (2013). Knowledge transfer challenges and mitigation strategies in global software development A systematic literature review and industrial validation. International journal of information management, 33(2), 333-355.
S24	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. Journal of systems and software, 80(4),571-583.
S25	Petersen, K. (2011).Measuring and predicting software productivity: A systematic map and review. Information and Software Technology, 53(4),317-343.
S26	Petersen, K. (2011).Measuring and predicting software productivity: A systematic map and review. Information and Software Technology, 53(4),317-343.
S27	Santiago, I., Jimnez,A., Vara, J. M., De Castro, V., Bollati, V. A., & Marcos, E. (2012).Model-Driven Engineering as a new landscape for traceability management: A systematic literature review. Information and Software Technology, 54(12), 1340-1356.

Table A.1: List of papers included in the SLR

Study	Reference
S28	Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review. <i>Information and software technology</i> , 50(9), 860-878.
S29	Hassler, E., Carver, J.C., Kraft, N. A., & Hale, D. (2014, May). Outcomes of a community workshop to identify and rank barriers to the systematic literature review process. In <i>Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering</i> (p. 31). ACM.
S30	Lavalle, M., Robillard, P. N., & Mirsalari, R. (2014). Performing systematic literature reviews with novices: An iterative approach. <i>Education, IEEE Transactions on</i> , 57(3), 175-181.
S31	Lane, S., & Richardson, I. (2011). Process models for service-based applications: A systematic literature review. <i>Information and Software Technology</i> , 53(5), 424-439.
S32	Prikladnicki, R., & Audy, J. L. N. (2010). Process models in the practice of distributed software development: A systematic review of the literature. <i>Information and Software Technology</i> , 52(8), 779-791.

Table A.1: List of papers included in the SLR

Study	Reference
S33	Kitchenham, B. A., Brereton, P., Turner, M., Niaz, M. K., Linkman, S., Pretorius, R., & Budgen, D. (2010). Refining the systematic literature review process through participant-observer case studies. <i>Empirical Software Engineering</i> , 15(6), 618-653.
S34	Kitchenham, B. A., Brereton, P., Turner, M., Niaz, M. K., Linkman, S., Pretorius, R., & Budgen, D. (2010). Refining the systematic literature review process through participant-observer case studies. <i>Empirical Software Engineering</i> , 15(6), 618-653.
S35	Da Silva, F. Q., Suassuna, M., Lopes, R. F., Gouveia, T. B., Frana, A. C. A., de Oliveira, J.P. N., ... & Santos, A. L. (2011, September). Replication of empirical studies in software engineering: Preliminary findings from a systematic mapping study. In <i>Replication in Empirical Software Engineering Research (RESER)</i> , 2011 Second International Workshop on (pp. 61-70). IEEE.
S36	Martnez-Ruiz, T., Mnch, J., Garca, F., & Piattini, M. (2012). Requirements and constructors for tailoring software processes: a systematic literature review. <i>Software Quality Journal</i> , 20(1), 229-260.

Table A.1: List of papers included in the SLR

Study	Reference
S37	Alves, V., Niu, N., Alves, C., & Valena, G. (2010). Requirements engineering for software product lines: A systematic literature review. <i>Information and Software Technology</i> , 52(8), 806-820.
S38	Rabiser, R., Grnbacher, P., & Dhungana, D. (2010). Requirements for product derivation support: Results from a systematic literature review and an expert survey. <i>Information and Software Technology</i> , 52(3), 324-346.
S39	Rabiser, R., Grnbacher, P., & Dhungana, D. (2010). Requirements for product derivation support: Results from a systematic literature review and an expert survey. <i>Information and Software Technology</i> , 52(3), 324-346.
S40	Liu, D., Wang, Q., & Xiao, J. (2009, October). The role of software process simulation modeling in software risk management: A systematic review. In <i>Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on</i> (pp. 302-311). IEEE.
S41	de Almeida Biolchini, J.C., Mian, P. G., Natali, A. C. C., Conte, T. U., & Travassos, G. H. (2007). Scientific research ontology to support systematic review in software engineering. <i>Advanced Engineering Informatics</i> , 21(2), 133-151.

Table A.1: List of papers included in the SLR

Study	Reference
S42	Da Silva, F. Q., Santos, A. L., Soares, S., Frana, A. C. C., Monteiro, C. V., & Maciel, F. F. (2011). Six years of systematic literature reviews in software engineering: An updated tertiary study. <i>Information and Software Technology</i> , 53(9), 899-913.
S43	Fernandez-Sez, A. M., Bocco, M. G., & Romero, F. P. (2010, July). SLR-Tool: A Tool for Performing Systematic Literature Reviews. In <i>ICSOFT(2)</i> (pp. 157-166).
S44	Bowes, D., Hall, T., & Beecham, S. (2012, September). SLuRp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In <i>Proceedings of the 2nd international workshop on Evidential assessment of software technologies</i> (pp. 33-36). ACM.
S45	Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. <i>International Journal of Project Management</i> , 30(4), 458-469.
S46	Radjenovi, D., Heriko, M., Torkar, R., & Ivkovi, A. (2013). Software fault prediction metrics: A systematic literature review. <i>Information and Software Technology</i> , 55(8), 1397-1418.

Table A.1: List of papers included in the SLR

Study	Reference
S47	Zhang, H., Kitchenham, B., & Pfahl, D. (2010). Software process simulation modeling: an extended systematic review. In <i>New Modeling Concepts for Today's Software Processes</i> (pp. 309-320). Springer Berlin Heidelberg.
S48	Barney, S., Petersen, K., Svahnberg, M., Aurum, A., & Barney, H. (2012). Software quality trade-offs: A systematic map. <i>Information and Software Technology</i> , 54(7), 651-662.
S49	Pitangueira, A. M., Maciel, R. S. P., & Barros, M. (2015). Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. <i>Journal of Systems and Software</i> , 103, 267-280.
S50	Luna, A. J. D. O., Kruchten, P., Pedrosa, M. L. D. E., Neto, H. R., & de Moura, H. P. (2014). State of the Art of Agile Governance: A Systematic Review. arXiv preprint arXiv:1411.1922.
S51	Tosi, D., & Morasca, S. (2015). Supporting the semi-automatic semantic annotation of web services: A systematic literature review. <i>Information and Software Technology</i> , 61, 16-32.

Table A.1: List of papers included in the SLR

Study	Reference
S52	Wen, J., Li, S., Lin, Z.,Hu, Y., & Huang, C. (2012). Systematic literature review of machinelearning based software development effort estimation models. Information and Software Technology, 54(1), 41-59.
S53	Inayat, I., Salim, S. S.,Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematicliterature re- view on agile requirements engineering practices and- challenges. Computers in human behavior, 51,915-929.
S54	Caldern, A., & Ruiz,M. (2015). A systematic literature review on serious games evaluation: Anapplication to software project management.Computers& Education, 87, 396-422.
S55	Ali, N. B., Petersen, K.,& Wohlin, C. (2014). A system- atic literature review on the industrial useof software process simulation. Journal of Systemsand Software, 97, 65-85.
S56	Sharafi, Z., Soh, Z.,& Guhneuc, Y. G. (2015). A sys- tematic literature review on the usage ofeye-tracking in software engineering. Informationand Software Technol- ogy, 67, 79-107.

Table A.1: List of papers included in the SLR

Study	Reference
S57	Kitchenham, B.,Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., &Linkman, S. (2010). Systematic literature reviews in software engineering tertiary study. Information and Software Technology, 52(8),792-805.
S58	Jalali, S., & Wohlin,C. (2012, September). Systematic literature studies: database searches vs.backward snowballing. In Proceedings of theACM-IEEE international symposium on Empirical software engineering andmeasurement (pp. 29-38). ACM.
S59	Lopez-Herrejon, R. E.,Linsbauer, L., & Egyed, A. (2015). A systematic mapping study ofsearch-based software engineering for software product lines. Information and Software Technology, 61, 33-51.
S60	Holl, G., Grnbacher, P.,& Rabiser, R. (2012). A systematic review and an expert survey oncapabilities supporting multi product lines. Informationand Software Technology, 54(8), 828-852.
S61	Dom, E., Prez, B.,& Rubio, . L. (2012). A systematic review of code generation proposalsfrom state machine specifications. Information andSoftware Technology, 54(10), 1045-1066.

Table A.1: List of papers included in the SLR

Study	Reference
S62	Ali, M. S., Babar, M. A., Chen, L., & Stol, K. J. (2010). A systematic review of comparative evidence of aspect-oriented programming. <i>Information and software Technology</i> , 52(9), 871-887.
S63	Chen, L., & Babar, M.A. (2011). A systematic review of evaluation of variability management approaches in software product lines. <i>Information and Software Technology</i> , 53(4), 344-362.
S64	Singh, K., Singh, P., & Kumar, K. (2016). A systematic review of IP traceback schemes for denial of service attacks. <i>Computers & Security</i> , 56, 111-139.
S65	Staples, M., & Niazi, M. (2008). Systematic review of organizational motivations for adopting CMM-based SPI. <i>Information and software technology</i> , 50(7), 605-620.
S66	Shahin, M., Liang, P., & Babar, M. A. (2014). A systematic review of software architecture visualization techniques. <i>Journal of Systems and Software</i> , 94, 161-185.
S67	Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. <i>Information and Software Technology</i> , 55(12), 2049-2075.

Table A.1: List of papers included in the SLR

Study	Reference
S68	Ali, S., Briand, L. C., Hemmati, H., & Panesar-Walawege, R. K. (2010). A systematic review of the application and empirical investigation of search-based test case generation. <i>Software Engineering, IEEE Transactions on</i> , 36(6), 742-762.
S69	Tahir, A., Tosi, D., & Morasca, S. (2013). A systematic review on the functional testing of semantic web services. <i>Journal of Systems and Software</i> , 86(11), 2877-2889.
S70	Bano, M., & Zowghi, D. (2015). A systematic review on the relationship between user involvement and system success. <i>Information and Software Technology</i> , 58, 148-169.
S71	Zhang, H., & Babar, M. A. (2013). Systematic reviews in software engineering: An empirical investigation. <i>Information and Software Technology</i> , 55(7), 1341-1354.
S72	Marshall, C., & Brereton, P. (2013, October). Tools to support systematic literature reviews in software engineering: A mapping study. In <i>Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on</i> (pp. 296-299). IEEE.

Table A.1: List of papers included in the SLR

Study	Reference
S73	Marshall, C., Brereton,P., & Kitchenham, B. (2014, May). 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 (p. 13). ACM.
S74	Portillo-Rodríguez, J., Vizcano, A., Piattini, M., & Beecham, S. (2012). Tools used in Global Software Engineering: A systematic mapping review. Information and Software Technology, 54(7), 663-685.
S75	Fernandez, A., Insfran,E., & Abraho, S. (2011). Usability evaluation methods for the web: A systematic mapping study. Information and Software Technology, 53(8), 789-817.
S76	dos Santos Rocha, R., & Fantinato, M. (2013). The use of software product lines for business process management: A systematic literature review. Information and Software Technology, 55(8), 1355-1373.
S77	Seriai, A., Benomar, O., Cerat, B., & Sahraoui, H. (2014, September). Validation of Software Visualization Tools: A Systematic Mapping Study. In Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on (pp. 60-69). IEEE.

Table A.1: List of papers included in the SLR

Study	Reference
S78	Mahdavi-Hezavehi, S., Galster, M., & Avgeriou, P. (2013). Variability in quality attributes of service-based software systems: A systematic literature review. <i>Information and Software Technology</i> , 55(2), 320-343.
S79	Malheiros, V., Hohn, E., Pinho, R., & Mendonca, M. (2007, September). A visual text mining approach for systematic reviews. In <i>Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on</i> (pp. 245-254). IEEE.
S80	Ramampiaro, H., Cruzes, D., Conradi, R., & Mendonca, M. (2010, October). Supporting evidence-based Software Engineering with collaborative information retrieval. In <i>Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on</i> (pp. 1-5). IEEE.
S81	Yang, Z., Li, Z., Jin, Z., & Chen, Y. (2014). A systematic literature review of requirements modeling and analysis for self-adaptive systems. In <i>Requirements Engineering: Foundation for Software Quality</i> (pp. 55-71). Springer International Publishing.

Table A.1: List of papers included in the SLR

Study	Reference
S82	Chagas, L. F., deCarvalho, D. D., Lima, A. M., & Reis, C. A. L. (2014). Systematic literature review on the characteristics of Agile project management in the context of maturity models. In <i>Software Process Improvement and Capability Determination</i> (pp.177-189). Springer International Publishing.
S83	Pons, C., Giandini, R., & Arvalo, G. (2012). A systematic review of applying modern software engineering techniques to developing robotic systems. <i>Ingeniera e Investigacin</i> , 32(1), 58-63.
S84	Raza, B., MacDonell, S.G., & Clear, T. (2013). Topics and treatments in global software engineering research-A systematic snapshot.
S85	RuizRube, I., Doderó, J. M., PalomoDuarte, M., Ruiz, M., & Gawn, D. (2013). Uses and applications of Software & Systems Process Engineering MetaModel process models. A systematic mapping study. <i>Journal of Software: Evolution and Process</i> , 25(9), 999-1025.
S86	Hernandes, E., Zamboni, A., Fabbri, S., & Thommazo, A. D. (2012). Using GQM and TAM to evaluate StArt-a tool that supports Systematic Review. <i>CLEI Electronic Journal</i> , 15(1), 3-3.

APPENDIX B

TOOLS USED IN SLR SEARCH PHASE

List of tools that were used in the SLR search phase along with their advantages and disadvantages of each tool as reported by the researchers in the survey.

B.1 Spreadsheet(Excel)

Table B.1: Spreadsheet(Excel) - Purpose

Item	Reported By
Organize studies	7
Group data	3
Removing Duplicates	3
Data Extraction	2
Analysis	2
Track process	2
Visualization	1
Collaboration	1
Inclusion and Exclusion Support	1
Classify data	1
Paper identification	1

Table B.2: Spreadsheet(Excel) - Limitations

Item	Reported By
Not adapted for SLRs	4
Manual analysis	2
Cannot Store PDF	1
Data extraction	1
Printing Graphs	1
Integration with other tools	1
Snowballing not supported	1

Table B.3: Digital Libraries and Search Engines - Purpose

Item	Reported By
Retrieve studies	1
Other	1
Remove Duplicates	1
Searching	1

Table B.4: Digital Libraries and Search Engines - Limitations

Item	Reported By
Search string differences across tools	2
Other	2
Manual analysis	1
Difficult Snowballing	1
Too Many False Positive	1

Table B.5: Jabref - Purpose

Item	Reported By
Organize studies	2
Availability	2
Filtering Capabilities	1
Removing Duplicates	1
Tailorable	1
Handle differently formatted references	1
Commonly used	1
Manage references	1
Annotation	1

Table B.6: Jabref - Limitations

Item	Reported By
Difficult to Customize	1
Defining Search String	1
Search feature does not work	1

Table B.7: Mendeley - Purpose

Item	Reported By
Organize studies	5
Organize notes	2
Generate bib file	1
Unknown	1
Coding	1
Study Classification	1
Storage of results	1
Export references	1

Table B.8: Mendeley - Limitations

Item	Reported By
No Customizable UI To Store Information	2
Not adapted for SLRs	2
No Space To Store Paper	1
Removing Duplicates is Difficult	1
PDF Annotation is Independent	1
Tracing Papers throughout SLR Phases	1
Sync problems	1
Paper organization	1

Table B.9: Custom tool - Purpose

Item	Reported By
Data extraction	1
Summarization	1
Studies Classification	1

Table B.10: Custom tool - Limitations

Item	Reported By
Other	1
Manual Work	1
Obscure	1

B.2 Digital Libraries and Search Engines

B.3 Jabref

B.4 Mendeley

B.5 Custom tool

B.6 StArt

Table B.11: StArt - Purpose

Item	Reported By
Organizing Studies	2
Protocol Guidelines	1
Removing Duplicates	1
Importing external resources	1
Step-by-step guide	1
Track search process	1

Table B.12: StArt - Limitations

Item	Reported By
Defining Search String	2
Downloading References is Not Automated	1
Snowballing not supported	1

B.7 End Note

Table B.13: End Note - Purpose

Item	Reported By
Organize bibliographic data	3
Download references	2
Collaboration	1
Extract bibliographic data	1
Validate search process	1

Table B.14: End Note - Limitations

Item	Reported By
Difficult to Collaborate	2
Incomplete Records	1
Lack of set-based operations	1
Manual Work	1
Difficult to Customize	1
Difficult to Import and Export	1

Table B.15: RefWorks - Purpose

Item	Reported By
Exporting References	1
Aggregate papers in common format	1
Storage of results	1

Table B.16: RefWorks - Limitations

Item	Reported By
Removing Duplicates is Difficult	1
No Customizable UI To Store Information	1
PDF Annotation is Independent	1
Tracing Papers throughout SLR Phases	1

B.8 RefWorks

B.9 Text Mining

Table B.17: RefWorks - Purpose

Item	Reported By
Identify keywords	1

Table B.18: RefWorks - Limitations

Item	Reported By
Results,made no sense	1

B.10 Google scholar

Purpose - None Reported

Limitations - None Reported

B.11 REviewER

Table B.19: RefWorks - Purpose

Item	Reported By
Downloading Papers Automatically	1
Automated and Aggregated Search	1

Table B.20: REviewER - Limitations

Item	Reported By
It does not download from all sources	1

B.12 Zotero

Table B.21: Zotero - Purpose

Item	Reported By
Cross Platform	1
Retrieve References	1
Detect duplicates	1

Table B.22: Zotero - Limitations

Item	Reported By
Slow	1
Managing duplicates	1

B.13 SLR Tool

Purpose - None Reported

Limitations - None Reported

B.14 Sharelatex

Limitations - None Reported

Table B.23: Sharelatex - Purpose

Item	Reported By
Writing Paper	1

B.15 Unknown Tool

Purpose - None Reported

Limitations - None Reported

B.16 Google drive

Table B.24: Google drive - Purpose

Item	Reported By
Organize studies	1
Version control	1
Storage of information	1

Table B.25: Google drive - Limitations

Item	Reported By
Disorganized	1
Desktop synchronization	1
No batch copy	1
Shared repositories confusing	1

B.17 Dropbox

Purpose - None Reported

Limitations - None Reported

B.18 MS Word

Purpose - None Reported

Limitations - None Reported

B.19 Publish or perish

Table B.26: Publish or perish - Purpose

Item	Reported By
Organize notes	1

Limitations - None Reported

B.20 Reseachr

Purpose - None Reported

Table B.27: Reseachr - Limitations

Item	Reported By
No Space To Store Paper	1
No Customizable UI To Store Information	1

B.21 SESRA

Table B.28: SESRA - Purpose

Item	Reported By
Process Management	1
Follow-up	1

Table B.29: SESRA - Limitations

Item	Reported By
Needs additional support	1

B.22 NVivo

Limitations - None Reported

Table B.30: NVivo - Purpose

Item	Reported By
Study Storage	1
Paper identification	1

Table B.31: GSResearch - Purpose

Item	Reported By
Customizability	1
Automation	1

B.23 GSResearch

Limitations - None Reported

B.24 SLuRP

Table B.32: SLuRP - Purpose

Item	Reported By
Automatic Download of Papers	1
Inclusion and Exclusion Support	1

Table B.33: SLuRP - Limitations

Item	Reported By
Difficult to use without training	1