JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
31st August 2017, Volume 2, Issue 2, JSEIS, CAOMEI Copyright © 2016-2017
www.jseis.org

Systematic reviews and mapping studies on software requirements: encyclopedia of objectives and issues for newbies

[1]Muhammad Imran Babar, [2]Masitah Ghazali, [3]Imran Ghani, [4]Falak Sher, [5]Muhammad Jehanzeb, [6]Shahid Nazir Bhatti, [7]Dayang N.A. Jawawi, [8]Awad Ali Abdur Rehman, [9]Frukh Zeshan
[1,5]Department of Computer Science, Army Public College of Management & Sciences, Rawalpindi, Pakistan
[2,4,7]Faculty of Computing, Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia
[3]Monash University, Malaysia
[6]Bahria University Islamabad, Pakistan
[8]Department of Computer Science, University of Kassala, Sudan
[9]Department of Computer Science, COMSATS (CIIT), Lahore, Pakistan
Email: [1]imran.babar77@gmail.com, [2]masitah@utm.my, [3]imran.ghani@monash.edu, [4]safalak2@live.utm.my, [5]jehanzeb.utm@gmail.com, [6]snbhattii@bui.edu.pk, [7]dayang@utm.my, [8]awad.ali625@gmail.com, [9]drzfurkh@ciitlahore.edu.pk

ABSTRACT

The domain of requirements engineering (RE) is intercepted by the interest of a wider community in the software industry. The role of RE is inevitable in the success and failure of software projects. RE is a multi-dimensional area and it is comprised of stakeholder analysis, requirements elicitation, and requirements prioritization mainly. In RE, the valuable requirements are explored from a critical set of stakeholders and these requirements are prioritized in order to design a system of high assurance. Several systematic literature reviews (SLR) are written related to RE domain which deliver a comprehensive knowledge in the domain of RE. Hence, currently no SLR exists that is able to provide an aggregate knowledge about RE intricacies. In this research, the different systematic reviews were investigated in order to aggregate the generated knowledge in different sub-domains of RE. This research highlights the objectives, sub-objectives of the existing SLRs and also presents information about RE issues in different perspectives. Initially, 214 studies were identified and only 39 studies were included in this SLR for research and analysis purposes.

Keywords: software requirements engineering; systematic literature review; systematic mapping studies; requirements engineering goals; requirements engineering issues; quality aspects;

1. INTRODUCTION

Requirements engineering (RE) is the most difficult part of software development life cycle. Brooks implicitly states that "the hardest single part of building a software system is deciding precisely what to build... Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements" [1]. In RE domain different key practices are applied in order to gather requirements for a successful development. RE is a sub-domain in software engineering (SE). Different researchers have defined RE differently. As stated by Zave "requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families" [2].

RE cloud is multifaceted and provides opportunities of research in several domains. The three key domains in RE are stakeholder analysis, requirements elicitation, and requirements prioritization along with other mandatory dimensions. The first and foremost domain in RE is stakeholder analysis. Tom Gilb defines a stakeholder, in his '*plangauge*', as 'any person or organisational group with an interest in, or ability to affect, the system or its environment' [3]. Stakeholders are the key entities during REP. Hence, the selection of highly valuable stakeholders has a significant impact on the RE process. A wide range of stakeholder analysis approaches are presented so far. However, there are few with diverse dimensions which focus the software stakeholders like [4-20]. All these stakeholder analysis approaches lack uniformity due to their application in diverse domains with peculiar analysis methods. However, some of the approaches are using existing methods with a little variation. The second key domain of RE is REP. Requirements "elicitation is the process by means of which software analyst gathers information, about the problem domain," [21] from key stakeholders. Several requirements elicitation techniques are presented with a blend of individual identity, hybrid and intermittent applications and methodologies like [22-32]. The third most important dimension of research in RE is requirements prioritization. In requirements prioritization the key needs of the stakeholders or users are prioritized based on some criteria in order to develop a system of high assurance [33], [34]. Near about 50 of the requirements prioritization techniques are proposed based on different measurement scales like ordinal scale, nominal scale, interval scale and ratio scale. Some of the software requirements prioritization approaches are analytical hierarchy process [35], cost-value approach [36], B-Tree prioritize [37], cumulative voting or the 100 Dollar test [38], numerical assignment [39], ranking, top ten requirements [40], planning game [41], theory w or win-win model [42], VIRP [43] and several other hybrid derivatives. In this research, we have adopted a systematic literature review (SLR) as per guidelines given in [44]. Several systematic literature reviews (SLR) are written related to RE domain in order to summarize the research contributions in a particular sub domain of RE. However, the purpose of this research is to aggregate the results of existing SLRs related to the focused goals and highlighted issues in RE process. An initial review protocol of the research is conducted in [45] that summarizes the research studies only. The effective RE practices are applied in order to make the things visible in the dense cloud of RE as shown in Figure 1.

Figure. 1 Requirements cloud

RE has ever been a live research domain in software engineering. In this research, the major focus is on SLRs. Hence, a holistic background of the existing SLRs has been presented in this section. This section however, does not focus on the different phases of SLR due to the enormous information that is available to researchers. An SLR has been presented on software requirements prioritization that summarizes the knowledge in areas like SLR and traditional views, RE and requirements prioritization, phases or design of an SLR, the way to conduct an SLR, and the final focus is on that how to conduct a high quality research in the domain of requirements prioritization [46]. The SLR conducted in [47] highlighted the effectiveness of the different software requirements prioritization approaches in terms of their applicability for smaller to larger datasets. It has been noted that most of the techniques are applied on smaller to medium datasets and few have been reported as suitable for prioritizing large number of requirements. A taxonomy of the errors is reported in an SLR for better visualization of requirements errors [48]. An SLR conducted in [49] highlights the practices of RE in the field of multi-agent systems. "A multi-agent system is a specific system that is composed of multiple interacting intelligent agents"[49].

The remaining paper is divided into 3 major sections. Section 2 gives the details of the research process. Section 3 is about results and discussions and the last Section 4 is about conclusion of the research study.

2. RESEARCH PROCESS

The guidelines proposed by Barbra Kitchenham for systematic literature review are followed here as discussed in [45]. However, some of the modifications are made in research questions that are shown in next sub-section.

a) Research questions

An SLR is written in order to aggregate and summarize the knowledge holistically in a given research domain. The significant contributions are identified based on the empirical evidence. The research papers included in this SLR are analysed based on some key research questions. Hence, in order to explore and aggregate the research contributions in the existing RE SLRs the following research questions are focused in this study:

1. RQ1: What are the objectives and sub-objectives of the existing SLRs about RE?
2. RQ2: What issues have been addressed in existing SLRs about RE?
3. RQ3: Who is going to conduct research on SLRs in the domain of RE?

The research questions are formulated based on the motivation described in Section 1. Moreover, the four key viewpoints of population, intervention, outcome and experimental design are also taken into account in order to comprehend the above-mentioned research questions [44]. The population represents the research papers that focus on SLRs related to the RE domain. The intervention is associated with the issues and research directions in the RE. The outcome is associated with an aggregate or summarized knowledge collected from the SLRs in the domain of RE. However, no specific experimental design is focused in the development and formulation of research questions. The research question RQ1 is associated with the key objectives of the SLRs. This question helps to find out the aims and research directions highlighted in different domains of the RE in the given SLRs. The RQ1 deals at a higher level of abstraction and provides an overview of the conducted SLRs. In order to carry out the nano-exploration (deep research) the RQ2 is formulated. The RQ2 deals with the issues and problems found and tackled in the current RE research. This research questions helps in understanding the different dimensions of the problems which are dealt with by the scientists. The RQ3 is formulated to find out that at who is involved in aggregating the knowledge in different domains of RE in the form of SLRs.

b) Data results

The research studies in the domain of RE were selected and evaluated. The studies that were selected were based on our above-mentioned research questions. Only the final studies were considered and all duplicate studies are not included in this research. Table 1 highlights the selected studies in the domain of RE.

Table. 1 Selected systematic literature reviews and mapping studies

Study Code	Author	Year	Reference	Focused Domain	Article Type	Published In
S[1]	Kashif	2006	[46]	RE Prioritization	SLR	Thesis
S[2]	Qiao Ma	2009	[47]	RE Prioritization	SLR	Thesis
S[3]	Gursimran and Jeffrey	2009	[48]	RE Errors	SLR	Journal
S[4]	Blanes et al.	2009	[49]	RE for Multi-Agent Systems	SLR	Book
S[5]	Ivarsson and Gorschek	2009	[50]	Technology Transfer	SLR	Journal
S[6]	Nicolas and Toval	2009	[51]	Specifications Generation	SLR	Journal
S[7]	Nelly et al.	2009	[52]	Specifications Techniques	SMP	Symposium
S[8]	Richard et al.	2010	[53]	Requirements Quality	SLR	Conference
S[9]	Alves et al.	2010	[54]	RE for Software Product Lines	SLR	Journal

S[10]	Loniewski et al.	2010	[55]	RE in Model- Driven Development	SLR	Book
S[11]	Rabiser et al.	2010	[56]	Product Derivation	SLR	Journal
S[12]	Mellado et al.	2010	[57]	RE for Security	SLR	Journal
S[13]	Jan and Ibrar	2010	[58]	Value-based Software Engineering	SLR	Thesis
S[14]	Iankoulova and Daneva	2011	[59]	RE for Security	SLR	Conference
S[15]	Dieste et al.	2011	[21]	RE Elicitation	SLR	Journal
S[16]	Ghanavati et al.	2011	[60]	RE for Business Process	SLR	Workshop
S[17]	Yue et al.	2011	[61]	Transformation Approaches	SLR	Journal
S[18]	Barmi et al.	2011	[62]	Alignment of RE and Testing	SMP	Conference
S[19]	Saha et al.	2012	[63]	RE Creativity	SLR	Conference
S[20]	Martı´nez-Ruiz et al.	2012	[64]	Software Process	SLR	Journal
S[21]	Li et al.	2012	[65]	Requirements Evolution	SLR	Conference
S[22]	Torkar et al.	2012	[66]	Requirements Traceability	SLR	Journal
S[23]	Tekaet al.	2012	[67]	Service Description	SLR	Book
S[24]	Lai et al.	2012	[68]	Distributed RE	SLR	Journal
S[25]	Lemos et al.	2012	[69]	RE Creativity	SLR	Conference
S[26]	Mohebzada et al.	2012	[69]	RE for Recommendation systems	SLR	Conference
S[27]	Pacheco and Garcia	2012	[70]	RE Stakeholders	SLR	Journal
S[28]	Rinkevicˇs and Torkar	2013	[71]	RE Prioritization	SLR	Journal
S[29]	Pergher and Rossi	2013	[72]	RE Prioritization	SMP	Conference
S[30]	Meth et al.	2013	[73]	RE Elicitation	SLR	Journal
S[31]	Pitangueira et al.	2013	[74]	RE Selection and Prioritization	SLR	Book
S[32]	Achimugu et al.	2014	[75]	RE Prioritization	SLR	Journal
S[33]	Babar et al.	2014	[76]	RE Stakeholders	SLR	Journal
S[34]	Rahul Thakurta	2014	[77]	RE Prioritization	SLR	Journal
S[35]	Yang et al.	2014	[78]	RE for Self-Adaptive Systems	SLR	Book
S[36]	Dermeval et al.	2014	[79]	Use of Ontologies in RE	SLR	Symposium
S[37]	Norman and Joerg	2015	[80]	RE Prioritization	SLR	Book
S[38]	Rahul Thakurta	2016	[81]	RE Prioritization	SMP	Journal
S[39]	Pérez García, et al.	2016	[82]	RE Evolution	SLR	Journal

Table 2 shows the different types of RE research domains. The research studies are divided into different categories, of Prioritization, Elicitation, Stakeholders, Security, Creativity and Miscellaneous, based on domain type. From data given in Table 2 it is evident that requirements prioritization, elicitation, stakeholders' analysis, RE practices for security and RE creativity are hot issues in the domain of RE. The issue of prioritization is the focus of 25% of the studies. Requirements elicitation is the focus of 5.12% studies. Stakeholder analysis is the focus of 5.12% studies, security requirements is the focus of 5.12% research studies, the issue of creativity is focused by 5.12% of the research studies. However, a percentage of 53.84% focuses on miscellaneous aspects or sub-domains of the RE.

Table. 2 SLRs domains

Domain Type	Included Studies	Total Studies	Percentage
Prioritization	S[1], S[2], S[28], S[29],S[31], S[32], S[34], S[37], S[38]	9	25%
Elicitation	S[15], S[30]	2	5.12%

Stakeholders	S[27], S[33]	2	5.12%
Security	S[12], S[14]	2	5.12%
Creativity	S[19], S[25]	2	5.12%
Miscellaneous	S[3], S[4], S[5], S[6], S[7], S[8], S[9], S[10], S[11], S[13], S[16], S[17], S[18], S[20], S[21], S[22], S[23], S[24], S[26], S[35], S[39]	21	53.84%

Figure 2 shows the details of publications research sources and frequency. Majority of the papers they are published in reputed journals, conferences, book chapters, thesis, workshops and symposiums. Figure 4 shows the yearly frequency of the publications and it is found that most of the research in SLRs related to RE is performed in a time bracket of year 2009 to year 2012. However, research in RE in other years is comparatively not enough.

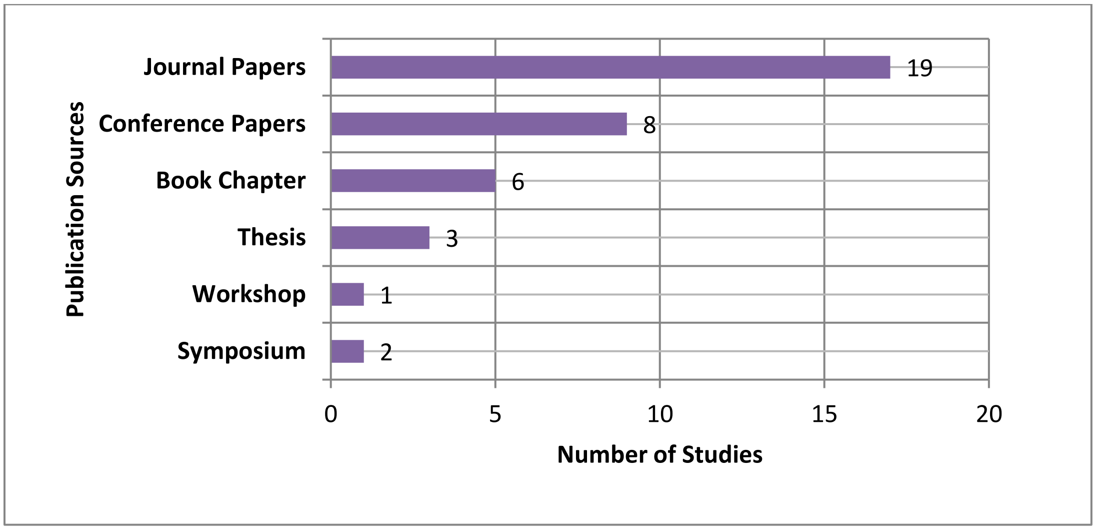

Figure. 2 Research sources and publication frequency

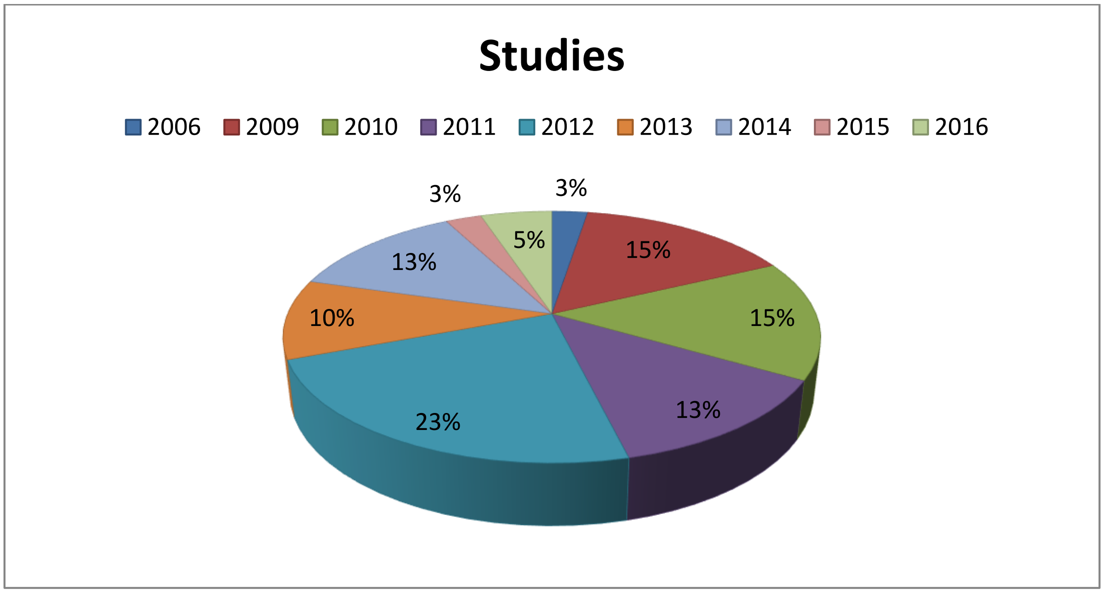

Figure. 3 Year wise percentages of publications

3. RESULTS FROM SECONDARY STUDIES

3.1 RQ1: Objectives and sub-objectives of the SLRs

In RQ1, the main objectives and sub-objectives of the SLRs are explored in order to investigate the key focused dimensions for a given area of the RE. This research question is designed to integrate the knowledge based on the main objectives and associated sub-objectives of the SLRs. Later on, based on this research we can find the key deviations of the reported results from main and sub-objectives of the research. The key objectives of the SLRs will help to further narrow down the research. Table 3 highlights the key objectives of the selected SLRs.

Table. 3 Objectives of the existing SLRs

SLR Code	SLR Objective
S[1]	To analyse the existing requirements prioritization techniques and to develop a research framework in order to align research in the domain of software requirement prioritization. The alignment of any new research is highly vital in order to make it useful for the intended community. The concept may be applied generically in the domain of software RE.
S[2]	To analyse the efficiency of the existing requirements prioritization approaches in terms of scalability. The measurement of efficiency serves as a scale that at which level the new contribution is going to contribute in enhancing the performance as compared to the previous scenarios.
S[3]	To identify and classify types of requirement errors into a taxonomy to support the prevention and detection of errors. Different taxonomies may be defined in order to categories the knowledge in different sub-domains of the software RE.
S[4]	The research investigates the application of requirements engineering techniques in order to develop Multi-Agent Systems. Researchers are developing different techniques in different domains of RE but there is a need to validate these techniques by applying them in real domains in order to check the worth of the proposed technique.
S[5]	The main goal of the research is to investigate that to what extent the industry professionals adopt the reported technologies in the domain of requirements engineering. In the present scenarios people are busy in presenting research in RE but there is a need to look at the point that either industry wants your methods or not.
S[6]	To analyse the research that focuses on software engineering models and textual requirements. Moreover, “to find methods and techniques dealing with the generation, translation, combination, integration, or synchronization of (system or software) models and textual requirements, in this order from (from models to requirements)”[51].
S[7]	To find out the empirical validation of the different aspects of software requirements based on context and research methods.
S[8]	To identify current research related to the management of the quality software requirements in areas of elicitation, dependencies, quality requirements metrics, cost estimation and prioritization.
S[9]	To find out the key implications of the RE practices, in the domain of software product line, to identify latest research trends, issues and the areas which need an improvement.
S[10]	To measure the automation level of existing software requirements engineering techniques and their application in the domain of Model-Driven development.
S[11]	“To identify and validate requirements for tool-supported product derivation”[56].
S[12]	To report the evidence based research in the domain of security requirements and to provide a framework for new research activities.
S[13]	To classify the contributions in the domain of value based software engineering systematically and to investigate the practical usability and usefulness of the proposed solutions.
S[14]	“To provide a comprehensive and structured overview of cloud computing security requirements and solutions” [83].
S[15]	To generate knowledge related to the application of requirements elicitation techniques by using results of the empirical studies on requirements elicitation.

S[16]	"To identify gaps and opportunities related to prioritization to improve compliance, templates for generating law-compliant processes, general links between legal requirements, goal models, and business processes, and semi-automation of legal compliance and analysis"[60].
S[17]	To find out the research studies which focus on transformation of requirements into analysis models, moreover, to highlight the different issues and to give future research directions.
S[18]	To find out the studies which focus on alignment of requirements specifications and testing of functional and non-functional requirements and to provide future research directions.
S[19]	To determine the different roles of creativity and to identify the impact of creativity techniques on requirements.
S[20]	The identification of requirements for process-tailoring notations and the analysis of the process-tailoring mechanisms which are used to support process tailoring.
S[21]	"To depict a holistic state-of-the-art of requirement evolution management"[65].
S[22]	"To examine requirements traceability definitions, challenges, tools and techniques"[66].
S[23]	Focuses on service oriented requirements engineering in order to identify the service description problems and the techniques presented to solve these problems.
S[24]	To provide references of the wiki tool for research in distributed RE and to identify new research directions.
S[25]	The research focuses on identification of the research trends and research opportunities in requirements creativity approaches for RE.
S[26]	To provide an overview of recommendation systems with respect to their characteristics and validation for the RE process.
S[27]	To analyse the stakeholders' identification methods for requirements elicitation.
S[28]	To collect knowledge about cumulative voting in order to support decision-making process and to propose a new method in order to detect the entities with same priority values.
S[29]	To focus on evolution in the domain of software requirements prioritization based on empirical evidence.
S[30]	To investigate the existing automated requirements elicitation techniques and to identify gaps.
S[31]	To investigate the Search-Based Software Engineering (SBSE) approaches for problems of requirements selection and prioritization.
S[32]	To analyse the existing software requirements prioritization techniques based on their limitations, scales and processes.
S[33]	To analyse the stakeholder's quantification techniques for value-based software development and to propose new research directions.
S[34]	To analyse the different considerations and their influence on the software requirements prioritization.
S[35]	To enable research community to understand the research trends in the domain of requirements modelling and analysis for self-adaptive systems.
S[36]	To focus on application of ontologies in RE domain for identification of addressed phases, languages, existing contributions, requirements modelling styles and benefits of the ontologies in RE domain.
S[37]	To find out a customized requirements prioritization criteria.
S[38]	To identify artifacts proposed for requirements prioritization.
S[39]	"To determine the degree of importance of the non-functional requirements" in the case of usability.

The second part of the RQ1 is associated with the sub-objectives of the SLRs. Table 4 highlights the sub-objectives of the selected SLRs.

Table. 4 Sub-objectives of the existing SLRs

SLR Code	Sub-objectives of the SLRs
S[1]	1. To evaluate the systematic reviews for deeper understanding. 2. To investigate the benefits of the systematic reviews in the domain of software engineering. 3. To investigate the systematic reviews in the domain of software engineering and especially in the sub domain of software requirement prioritization.

	4. To investigate the existing requirements prioritization approaches and the studies that are associated with these existing approaches. 5. To conduct an SLR in order to find out the empirical evidences of the existing software requirement approaches. 6. To propose a research framework that may help in facilitation of systematic reviews in future in the domain of software requirement prioritization.
S[2]	1. To analyse the existing requirement prioritization techniques systematically in order to draw the results. 2. The specific aim of the study is associated with the identification of the techniques that are presented for larger requirement sets. 3. To evaluate the evidences presented for the said studies in order to measure the efficacy of the evidences. 4. The evidences are used to draw general conclusions from the related studies.
S[3]	1. To assess that at which level the requirement errors in the existing approaches are useful for an improvement. 2. To identify the types of the errors reported so far in the literature. 3. The investigation of the human errors contribution with respect to the human cognition and psychology. 4. To present an error taxonomy.
S[4]	1. To summarize the knowledge in the domain of RE techniques applied in the development of multi-agent systems. 2. To evaluate the existing studies and find gaps for future research or investigation. 3. To integrate the existing RE techniques with the existing analysis and design methodologies in order to develop more robust multi-agent systems.
S[5]	1. To list all the RE technologies that exist so far. 2. To analyse the sub-process areas that are focused by the RE technologies. 3. To analyse the RE technologies based on their usage with respect to the project timeline. 4. To analyse the current state of the evaluation of the RE technologies. 5. To analyse the research methods that are used in the evaluation of the RE technologies. 6. To analyse the context or setting of the RE technologies in which they are evaluated. 7. To investigate the subjects (professionals) used in the evaluation of the RE technologies and their influence on the findings. 8. To analyse the scale of the evaluation of the RE technologies. 9. To analyse the realism based on the above-mentioned objectives of context, subjects, research method and scale. 10. The evaluation descriptions of the RE technologies are understandable or not and the findings can be mapped to other environments.
S[6]	1. To identify the value that is drawn on the generation of requirements specifications from different SE models. 2. To identify the techniques used to build the initial software engineering models, corresponding textual requirements, documents and the transformation procedures[51].
S[7]	1. To find out the quality aspects of the requirements specification process and product. 2. To identify the study settings and problem domains for the investigated quality aspects. 3. To find out the research method that was used for analysis of the aspects.
S[8]	1. To find out the empirical investigations related to the quality requirements. 2. To identify the empirical research methods used to evaluate quality requirements[53].
S[9]	1. To identify the RE methods/tools in the domain of software product lines. 2. The identification of the evidence for adoption of the proposed methods. 3. To find out the limitations of the current RE methods for software product line.
S[10]	1. To identify the RE techniques used in model-driven development (MDD). 2. To identify the level of automation of RE techniques used in MDD.

S[11]	1. To identify the key requirements for product derivation support in software product line engineering. 2. To identify the relevance and relative importance of the requirements. 3. To identify the analysis and realization of the requirements using existing tools.
S[12]	1. To identify initiatives and experience reports in Software Engineering which consider security requirements from the beginning of the IS development.
S[13]	1. To identify and classify software development process areas and sub-process areas in VBSE. 2. To identify and classify contributions in VBSE. 3. The identification and classification of research type. 4. The identification of software development process areas and sub-process areas which require due consideration based on contributions and research types in the domain of VBSE. 5. The identification of value dimensions, software development process areas and sub-process areas in VBSE.
S[14]	1. To identify the types of the cloud security requirements presented in the literature. 2. The identification of the offered solutions for the cloud security requirements. 3. To identify the type of research conducted in the domain of cloud security requirements.
S[15]	1. To identify the effectiveness of the software requirements elicitation techniques.
S[16]	1. To identify the goal-oriented frameworks that establish and manage the legal compliance of organizations. 2. To identify the goal modeling notations in order to model legal aspects and support compliance. 3. To identify the guidelines for extracting legal requirements. 4. To map the legal requirements to the goal models. 5. To identify the "methods that provide templates for modeling compliant business process"[60]. 6. To identify the "methods that help organizations prioritize instances of non-compliance" [60]. 7. The identification of the tool support for compliance management.
S[17]	1. To find out the approaches to transform requirements into analysis models. 2. The identification of requirements representation ways used in the existing requirements transformation approaches. 3. To find out the level of difficulty faced by the users in documentation of such requirements. 4. The identification of the different tools available for requirements transformation. 5. To identify the analysis models generated by the requirements transformation approaches. 6. The identification of behavioural and structural aspects of a system given by a generated analysis model. 7. The identification of intermediate models that are generated during requirements transformation. 8. The identification of effect of intermediate models on transformation in terms of efficiency. 9. Identify that either the transformation approaches are automated, semi-automated, automatable, or manual. 10. Identify the different types of algorithms used in the requirements transformation approaches. 11. Identification of the different steps used to transform requirements. 12. Identification of the traceability management support in the existing requirements transformation approaches. 13. Identification of empirical validation of the approaches based on case studies. 14. Identification of the results of the case studies. 15. To identify the research methods, other than case studies, used for evaluation purposes. 16. To identify the limitations of the current requirements transformation approaches. 17. The identification of the open issues for future research.
S[18]	1. To identify the studies that focus on linking of specifications and requirements (functional and non-functional) testing. 2. To identify the focused areas and level of their investigation. 3. To identify the studies that focus on NFRs. 4. To identify the perspectives that deals with alignment or traceability of requirements and testing. 5. To identify the solutions presented for alignment of requirements and testing.

	6. To identify the forum where the research is published.
S[19]	1. The identification of influential creativity techniques for RE. 2. To identify the importance of the creativity techniques for RE. 3. To explore the ways to use creativity techniques for RE. 4. To identify the future scope of creativity techniques for RE.
S[20]	1. To identify the elements of the software process model for adaptation. 2. To identify the tailoring operations of the existing software process models. 3. Identify the modeling notations used in the existing software process models. 4. Identification of the ways used to tailor the processes to meet the characteristics of the project or organization.
S[21]	1. To identify the definition of requirements evolution. 2. Identification of the activities of the requirements evolution management process[65]. 3. To identify the ways to measure requirements evolution.
S[22]	1. To identify, clarify and understand the requirements traceability and its techniques. 2. To identify the factors that may hinder the implementation of requirements traceability techniques.
S[23]	1. The identification of the existing service description methods. 2. To identify the problems of the existing service description methods. 3. To identify the pros and cons of the existing service description methods.
S[24]	1. To identify the RE-specific wiki tools. 2. To identify the prototypes used for development of RE-specific wiki tools. 3. To identify the features of RE-specific wiki tools for RE activities. 4. To identify the environments of the RE-specific wiki tools where they are used and developed.
S[25]	1. To identify the existing studies to foster creativity in RE. 2. To identify the implications of these studies in research and RE. 3. To identify the pros and cons of the existing studies on creativity in RE.
S[26]	1. To identify the recommendation systems for RE. 2. To identify the RE activities being focused by the recommendation systems. 3. To identify “the state of validation of recommendation systems”[84] for RE. 4. To identify the different characteristics of the recommender systems.
S[27]	1. To identify the methods or techniques for stakeholder’s identification in RE. 2. To identify the recommended effective practices for stakeholder’s identification in RE. 3. To identify the effects of wrong analysis of the stakeholders on quality of the requirements. 4. To identify the necessary or proposed RE practices for stakeholder analysis.
S[28]	1. To identify “the state of practice in empirical studies that use cumulative voting”[71]. 2. To identify the methods used for analysis of the results of cumulative voting. 3. To identify the equal priority of the requirements using equality in cumulative voting.
S[29]	1. To identify the research areas in the domain of software requirements prioritization. 2. To identify the different categories of the research articles related to the software requirements prioritization. 3. To identify the different types of the empirical studies related to the software requirements prioritization.
S[30]	1. To identify the “tool support for automated requirements elicitation from nature language documents”[73]. 2. To analyse the existing research work in design and evolution perspectives. 3. To categorise the automated requirements elicitation tools based on design and evolution perspectives.
S[31]	1. To identify the modeling approaches for selection and prioritization of requirements with a special emphasis on SBSE. 2. To identify the methods for selection and prioritization of requirements with a special emphasis on SBSE.

	3. To identify the search approaches for selection and prioritization of requirements with a special emphasis on SBSE.
S[32]	1. To identify the existing software requirements prioritization techniques. 2. The identification of the descriptions and limitations of existing software requirements prioritization techniques. 3. To explore the taxonomy of prioritization scales in the existing software requirements prioritization techniques. 4. To explore the different prioritization processes used in the existing software requirements prioritization techniques.
S[33]	1. To identify the stakeholder attributes reported in the existing studies. 2. To identify the usage context of the stakeholder attributes. 3. To identify the different types of the stakeholders. 4. To identify the different metrics reported in the existing studies for stakeholder analysis. 5. To identify the different issues related to the value-based software development.
S[34]	1. To explore the different considerations taken into account for software requirements prioritization. 2. To identify the different developments in the domain of software requirements prioritization artifacts.
S[35]	1. To identify the time of research publications in the domain of modeling and analysis of self-adaptive systems. 2. To identify the venue of research publications in the domain of modeling and analysis of self-adaptive systems. 3. To identify the research group of research publications in the domain of modeling and analysis of self-adaptive systems. 4. To identify the region distribution of research publications in the domain of modeling and analysis of self-adaptive systems. 5. To identify the different modeling methods and RE activities in the domain of modeling and analysis of self-adaptive systems. 6. To identify the different requirements quality attributes and application domains in the modeling and analysis of self-adaptive systems. 7. To identify the different methods that are better applied and rigorously evaluated in the modeling and analysis of self-adaptive systems. 8. To identify the different detailed RE activities in the modeling and analysis of self-adaptive systems.
S[36]	1. Identification of the RE phases that are supported by the use of ontologies. 2. The identification of the languages that are used in ontology-driven RE process. 3. The identification of the existing contributions in the domain of ontology-driven RE process. 4. The identification of the different RE styles supported by the use of ontologies. 5. To identify the evidence of associated benefits with the use of ontologies in RE process.
S[37]	1. "Which prioritization criteria are discussed in the requirements prioritization literature? [80]"
S[38]	1. To identify the objectives of software requirements prioritization. 2. To explore the published software requirements prioritization artifacts. 3. To identify the theoretical foundations of published requirements prioritization artifacts. 4. To identify the design characteristics of published requirements prioritization artifacts. 5. To find out the factors that affect the requirements prioritization factor as a whole.
S[39]	1. To identify the methods used for assessment of non-functional requirements (NFRs) of usability. 2. To identify the procedures used for assessment of NFRs of usability. 3. To identify the supporting tools for assessment of NFRs of usability. 4. To find out the validation of the presented proposal either in an industrial or academic setting.

3.2 RQ2: Issues addressed in the RE SLRs

In software engineering, there are several issues which need to be addressed in order to present acceptable and established paradigms. Hence, it is stated in [85] that "Software engineering does not yet have a widely recognized and widely appreciated set of research paradigms in the way that other parts of computer science do. That is, we don't recognize what our research strategies are and how they establish their results". There are different RE issues that are related to the inception phase, elicitation of the requirements, prioritization of the requirements, requirements creativity and so on. Based on the existing SLRs domains as shown in Table 2 the RQ2 is addressed and the results of the SLRs are given with a broader coverage based on these listed problem domains.

3.2.1 Software stakeholder identification and quantification

There are different issues that are stated in the research related to the stakeholders and software requirements. The three key problems of software requirements stated in the literature are comprehension, volatility and gaining[86-88].Stakeholder identification and quantification is the key phase to improve the process of software requirements elicitation[89] and it helps to solve the problems as stated above. The identification of stakeholders is not carried out well and this results in incomplete set of requirements for software projects[90].In [70] a total of 47 studies discuss the stakeholder identification and quantification methods. Majority of the studies are not validated and it is difficult to analyse the impact of these studies on requirements elicitation[70]. Out of 47 papers 40 are grouped into three categories as stated in [91]. Out of 40 studies 6 studies are reported as that merely describes stakeholders without any analysis, identification and quantification process. Out of 40 studies 23 studies focus on interaction between stakeholders. Out of 40 studies 11 studies give an assessment of the stakeholders. Table 5 reports the issues related to stakeholder identification and quantification.

Table. 5 Stakeholder identification and quantification issues

SLR Code	Reported Issues
S[27]	• Few stakeholder identification methods. • The existing methods are unstructured. • Lack of quantitative data analysis for stakeholder identification. • The existing studies on SI are not standardized and consequently the SI methods are also not standardized. • The existing studies on SI do not cover the same aspects. • Are not applicable to the same situations. • Lack of assignment of roles to the stakeholders. • Lack of stakeholders' interaction analysis. • Lack of coverage of the human aspects in order to carry out SI. • Existing studies just characterize the stakeholders. • There are few methods that cover stakeholder assessment. • Missing important aspects. • There is an issue of identification of sufficient stakeholders. • Issue of documentation of the collected information related to the SI. • Lack of guidelines and standards for SI. • Existing studies only facilitates in identification of stakeholders. • The industry practices or standards like CMM, CMMi, BABOK, SWEBOK, ISO/IEC 12207 do not explain SI methods. • Incorrect SI results in incomplete requirements and the relevant information will be omitted.
S[33]	• The existing SI approaches or methods are not systematic. • They are difficult to initiate. • Applied in different domains and are not feasible for all domains. • Cannot be adopted as a standard due to the issue of non-uniformity. • "There is still no SIP framework or uniform description"[92, 93]. • The existing SIQ approaches are complex and lack in low level details.

	• The existing SIQ approaches present stakeholders at higher level of abstraction. • Incorrect results of existing SIQ approaches. • The SIQ approaches are time consuming and costly[5, 94]. • Most of the studies are carried out to explore the new stakeholders instead of starting from the scratch.

3.2.2 Software requirements elicitation

Software requirements elicitation process is used to find out the key requirements of the users for a given software system. The elicitation mechanism is based on interactions between user and system analyst. Different techniques are used to elicit the requirements like, view-points, interviews, prototyping, work groups and many others. The reported issues related to software requirements elicitation are shown in Table 6.

Table. 6 Reported issues of software requirements elicitation

SLR Code	Reported Issues
S[15]	• The existing elicitation techniques are separate studies and their results are not presented in a combined form. • The usage criteria of different techniques[87], [95], [40], [96], is not evidence based rather it is based on personal experiences and theory.
S[30]	• The existing software requirements elicitation process is time-consuming. • This existing elicitation process is error-prone. • The elicitation process is carried out manually. • Unclear and incomplete requirements may lead towards a total failure.

3.2.3 Software requirements prioritization

"Software requirements prioritization is an activity during which the most important requirements for the system (release) should be identified" [90]. Software requirements analysis is a key element in software development process. Hence about requirements F.P. Brooks says that "the hardest single part of building a software system is deciding what to build…. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later" [97]. Different issues related to the software requirements prioritization are reported in the systematic reviews and in order to overcome these issues new and innovative ways are presented. The issues associated with each prioritization technique are also highlighted in the SLRs. The reported issues of software requirements prioritization are highlighted in Table 7.

Table. 7 Reported issues of software requirements prioritization

SLR Code	Reported Issues
S[1]	• There is a lack of empirical work in the domain of software requirements prioritization. • Most of the research studies focus on only two aspects of cost and importance while the other aspects of the requirements, like customer satisfaction, strategic, etc., are neglected. • The suitability of the techniques for different conditions is not highlighted. • Most of the techniques solve the problem of requirements prioritization with a set of requirements less than 20. Hence, it results in the scalability issue. • The existing studies lack in prioritizing detailed refined requirements rather the requirements at higher level or feature level are given considerations. • Most of the existing studies focus on functional requirements. There is a need to give due consideration to non-functional requirements. • It is difficult to generalize the effectiveness of any technique based on the presented results in different studies. • Most of the software requirements prioritization techniques are not applied in real scenarios or industrial settings. • There is no consideration of requirements dependencies in prioritization process.

	• There is a lack of evidence on quality of the research in software requirements prioritization.
S[2]	• The roles and number of the participants in the evaluation of prioritization techniques are not specified. • Most of the techniques are evaluated based on a limited number of software requirements. • It is not specified when the software requirements prioritization process is carried out.
S[28]	• "Few studies present prioritization of more than 30 items and the availability of research data is somewhat limited" [98].
S[29]	• There is a limited research in the domain of tool evaluation for software requirements prioritization. • The case studies do not focus on the ease of use as a key factor in order to evaluate a requirements prioritization method. • The factors of scalability, understandability, learnability and fault tolerance are not given due consideration during evaluation of different requirements prioritization approaches. • Moreover, the aspects of requirements dependencies and stakeholders point of views are not addressed properly. • The granularity level of the requirements is very high in the existing research studies and the low-level details are not given. • There is no research on the relevance of the attributes to evaluate the different requirements prioritization methods.
S[31]	• The next release problem. • Multiple objective next release problem • Problem of release planning. • Issue of prioritization of requirements. • Issue of requirements interaction management.
S[32]	• Analytical Hierarchy Process technique is time consuming when the number of requirements increases and is not scalable [99-101]. • The complexities of the goal graphs, in attribute goal-oriented requirement analysis technique, are not managed efficiently [102]. • The benefit and cost prediction technique does not consider requirements evolution and weighting of requirements using a scale [103]. • Binary Search Tree does not assign any priority value to requirements [100]. Moreover, it is complex and not scalable [104, 105]. • Binary priority list does not consider dependencies of the requirements and the priority criterion is based on just one attribute [106]. It is also not scalable [107]. • Case-Based Ranking technique is not scalable [108]. • Correlation-based priority assessment framework does not focus on requirements priority having negative correlations [109]. • Cost-value ranking is time consuming and non-scalable [101]. • In cognitive driven requirements prioritization, the weights are not aggregated and globalized by using some method during prioritization [110]. • The EVOLVE technique induces computational complexity [111] and its effectiveness needed to be tested in a highly complex industrial setting [112]. • Fuzzy AHP is not scalable and interdependencies are not tackled [113]. • Hierarchy AHP results in judgemental errors [99]. • Interactive requirements prioritization technique is unable to conduct sufficient experimentation [114]. • Lanchester theory is unable to describe relative values of the linguistic terms which may help in finding out the relative weights of the requirements [31]. • Minimal Spanning Tree is more prone to judgemental errors [99].

	• Multi-criteria Preference Analysis Requirements Negotiation technique is unable to detect inconsistencies in the ranking values [21, 115]. • Multi-objective next release problem technique does not provide a complete list of prioritized requirements rather a subset of the requirements, for the next release planning, is selected [22]. • In Numerical Assignment technique the ranking is based on categories like high, medium and low which results in confusion [99, 116]. • Pairwise analysis technique is complex and the results are unreliable [101]. • Planning Game technique is not scalable and is unable to prioritize a set of large number of requirements [100]. • In Priority Groups technique there is no consistency in the judgment of the decision maker [99]. • Quality Functional Deployment technique is not scalable and the inconsistencies are common [21]. Moreover, it is applied to small systems or sub-systems [23]. • The Ranking technique is not scalable. Moreover, it does not show a significant or relative difference among prioritized requirements [24]. • The requirements triage does not recall results and is prone to errors [107, 117]. • Requirement Uncertainty Prioritization Approach is not scalable and is only suitable for small projects with few requirements [118]. • Round the Group Prioritization technique is also not scalable [26, 101]. • Simple Multi-criteria Rating Technique by Swing is also not scalable and is unable to detect inconsistencies [21]. • Software Engineering Risk Understanding and Management technique is unable to cater requirements dependencies [27]. • Hundred Dollar Test or Cumulative Voting technique is not scalable [24, 26]. • Technique for Ordering from Similarity to Ideal Solution is unable to organize requirements in a hierarchical fashion. Moreover, the technique is unable to update the ranks for newly evolved requirements [119, 120]. • The Top Ten Requirements technique is ambiguous as the weights are not assigned in the ranking process [30]. • Value-based Requirements Prioritization technique is unable to organize the requirements in a hierarchical form. Moreover, the report generation and pre-requisite handling is poor [119]. • Value-Oriented Prioritization technique is unable to handle requirements dependencies [111]. Moreover, the technique is not scalable [100]. • Value-based Intelligent Requirements Prioritization technique does not categorise the ranked requirements [43]. • Weighted Critical Analysis technique does not cater for requirements dependencies. • Wieger's Method can be easily manipulated by a particular stakeholder in order to achieve his or her goals [100]. • In WinWin method it is very difficult to reach to a consensus due to the involvement of biased stakeholders and the prioritization results in inconsistent ranking of the requirements [21, 31].
S[34]	• It is required to automate the different approaches in order to use them in the real context in the industry. Hence, there is a lack of the automation of the existing software requirements prioritization techniques.
S[37]	• The determination of a customized requirements prioritization criteria is time consuming [121]. • The criteria identification is based on gut feeling which may result in errors and there is a risk to select wrong criteria for requirements prioritization [121].

3.2.4 Software requirements creativity

To develop a software product the innovation must be given due consideration. For innovation, the requirements are analysed by applying creative techniques. Creative RE techniques help to produce innovative ideas that may yield high business values. Creativity is the part and parcel of RE as the innovative requirements are explored, for the new system, in this phase [122]. The SLRs conducted on creative software requirements techniques have reported different issues that are shown in Table 8.

Table. 8 Reported issues of software requirements creativity

SLR Code	Reported Issues
S[19]	• There is a lack of creative techniques for unclear requirements [123]. • In order to cope with the issues of software complexity and market competitions the innovative RE techniques are required.
S[25]	• There is a need to integrate the tools with RE in order to support creativity. • To investigate about new creativity tools to solve the issue of filtered ideas that, do not contribute towards RE goals or objectives. • There is also a need to categorize the creativity techniques based on the RE processes. • The existing RE creativity techniques are time consuming because they need enough time for preparation.

3.2.5 Security requirements

The internet cloud is connecting millions and trillions of devices for communication and this connectivity may result in severe security issues and may face threats by security hackers in the form of malicious code and terrorism [124]. There is a need to integrate RE process with the security concerns in order to develop a system that may stand against cyber-crimes. The reported issues of security requirements are highlighted in Table 9.

Table. 9 Reported issues of security requirements

SLR Code	Reported Issues
S[12]	• Issue of security breaches for the information systems [125]. • Vulnerability to the threats like malicious hackers, cyber terrorists and code writers [126]. • The security breaches may result in huge financial losses and threats to human beings [126]. • One of the main highlighted issue is the "security is dealt with when the system has already been designed and put into operation" [127]. • There is a lack of understanding of the real security requirements. • Developers design the solutions based on security mechanisms and the declarative propositions are ignored with respect to the required level of security [128]. • For non-functional security requirements there is a dire need of security expertise [61].
S[14]	• The issue of access control in "which the system limits access to its resources only to authorized entities" [129]. • The issue of attack/harm detection is taken into account in order to detect, record and to notify requirements after a successful attack or attempt [128]. • The issue of integrity is highlighted in which the protection of components is taken into account from intentional unauthorized acts that may harm data. • The issue of security auditing is presented in which the use of security mechanism is audited by evaluating different events that are related to the security [128]. • The issue of privacy is highlighted in which the unauthorized users are prevented to access the sensitive data or information. • The issue of non-repudiation is concerned with the requirements to prevent the party "interaction with the cloud to deny the interaction" [129].

3.2.6 Miscellaneous

Under the umbrella of this section different issues have been covered related to the RE domains like requirements traceability, information errors, requirements evolution or change and so on. Table 10 highlights the issues related to the sub-domains like traceability, information errors, requirements evolution and requirements change.

Table. 10 Issues related to miscellaneous sub-domains of RE

SLR Code	Reported Issues
S[3]	• Different software error information methods are reported which "indicates that knowledge of the source of faults is useful for process improvement" [130]. The presented methods are unable to "provide a formal process to assist developers in finding and fixing errors" [130]. • Another drawback of these methods is that they depend on a fault sample for identification of errors that is cost intensive, people-intensive and useful however, some errors are overlooked instead of the detailed list of errors [98, 130-135]. • Nine error information methods are reported that resolve different limitations however, there other limitations that are associated with them. Existing methods do not provide sufficient documentation related to the problem reports and inspected results; moreover, the dependence on historical data is very high [98, 132, 135-137]. • The cost of causal analysis implemented through actions is about 0.5% to 1.5% of the budget, hence, a start-up investment is highly desirable [98, 132, 133, 135]. • Experienced manpower is required to find out the main causes of failure [134].
S[5]	• The research related issue is highlighted. It is shown "that new technologies are presented once and that replication by other authors or additional evaluations are not common". • It is difficult to find out the requirements before initiation of the project and the selection of a relevant analysis tool becomes useless [138].
S[6]	• There is a lack of research on interconnecting different requirements models and more research is required to find out the ways to integrate different requirements engineering techniques [139].
S[8]	• "The handling and balance of quality requirements are important and difficult part of requirements engineering process" [111].
S[9]	• Studies related to RE for software product lines are not clearly reported hence it is difficult to reach to a specific conclusion about SPLs. • There is a lack of evidence of effectiveness of RE approaches for SPLs. • Difficult to build a confidence of the researchers, in terms of adoption, about a proposed method for RE in the domain of SPLs.
S[10]	• Different methods and techniques have been proposed in the domain of model driven development (MDD) but only few discussed the role of RE in MDD.
S[11]	• In the domain of SPLs only few approaches and tools exist for product derivation. • No focus on how to utilize the SPLs effectively rather the major focus is on scope, definition and development of SPLs. • The product derivation process is "slow and error prone even if no new development is involved".
S[13]	• Most of the practices in the domain of software engineering are carried out in a value neutral settings which result in project failures.
S[17]	• In model driven development the transformation from requirements to analysis model is not taken into account where as in MDD the analysis model is a starting point and code is an ending point [140].
S[18]	• Software errors may result in irreparable losses hence there is a need to align RE and testing.
S[20]	• The existing process reference models are generic and have mechanism limitations in terms of adaptation of the processes with respect to the needs of organizational units and project goals and environments.

S[21]	• Requirements evolution may result in higher costs, schedule slips,
S[22]	• There is no immediate effect of requirements traceability on developmental process [141]. • There is a lack of resources, cost, time for requirements traceability and coordination too. The developmental methods are informal and there is no standardization [142]. • Lack of team coordination, results in more costs. Moreover, it is difficult to manage too many links due to excessive traceability [143]. • The manual requirements traceability is very costly [144, 145]. • Requirements change results in inappropriate information to make decisions [146]. • The problems of tracing back to the sources [147]. • Issue of requirements traceability adaptation [148]. • The non-functional requirements cannot be traced [149]. • The factors related to organization, environment and technicality [150]. • Challenges associated with the requirements management [151]. • The issue of responsibility to identify requirements and to start with and the requirements change. The knowledge loss if the main employer vanishes from the scene [152].
S[23]	• There is a lack of description of the behaviours [153, 154]. • Requirements specifications are not precise [155]. • The service description methods are producer only centric [156, 157]. • The dynamic adaptability lacks in existing service description methods and do not support situation awareness [158-161]. • There exists a gap in IT and business oriented service realizations [162-165]. • In syntactic description there is a lack of semantics and the methods fail in description of QoS [158-160, 166-174].
S[24]	• To narrow down communication gap due to global or distributed scenarios. • How to manage the distributed phases of the requirements engineering [139, 175]?
S[26]	• In case of recommendation systems for RE most of the techniques are validated based on academic experiments and the rest are not validated. There is a dire need that the validations must be carried out in industrial settings.
S[35]	• The RE for self-adapting systems (SASs) still lacks research. • There must be research on model and context uncertainty. • There is a need to focus on "reasoning with context uncertainty". • The research must be carried out in the domain of "requirements-driven adaptation with context uncertainty". • The other research issue is to focus on "requirements-driven architecture adaptation and requirements-driven evolution". • There is need to present quantitative models and representations for RE of SASs. • It is also required to explore that how one can improve "the quality of modeling methods and RE activities" for SASs.
S[36]	• There is a lack of knowledge about the use of ontologies in the domain of RE.

3.3 RQ3: Affiliations of the SLRs

RQ3 shows the detailed affiliations of different universities, research institutes and countries involved in conducting SLRs. Moreover, the publication forum and sources are also shown in Table 11.

Table. 11 Research Studies per Publication Source

Study Code	Publication Forum	Total Studies	Publication Source
S[1], S[13]	Blekinge Institute of Technology Sweden	2	Blekinge Institute of Technology

S[2]	Auckland University of Technology New Zealand	1	Auckland University of Technology
S[3], S[6], S[9], S[11], S[28], S[30], S[32],	Information and Software Technology	7	Elsevier
S[5], S[17], S[38]	Requirements Engineering, Springer	3	Springer
S[7]	Third International Symposium on Empirical Software Engineering and Measurement	1	IEEE
S[8]	36th EUROMICRO Conference on Software Engineering and Advanced Applications	1	IEEE
S[10]	Model driven engineering languages and systems, LNCS	1	Springer
S[12]	Computer Standards & Interfaces	1	Elsevier
S[14]	Sixth International Conference on Research Challenges in Information Science (RCIS)	1	IEEE
S[15]	IEEE Transactions on Software Engineering	1	IEEE Computer Society
S[33]	IET Software	1	IET
S[16]	Fourth International Workshop on Requirements Engineering and Law (RELAW 2011)	1	IEEE
S[18]	Fourth International Conference on Software Testing, Verification and Validation Workshops	1	IEEE Computer Society
S[19]	IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision	1	IEEE
S[20]	Software Quality Journal	1	Springer
S[21]	16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)	1	IET
S[22]	International Journal of Software Engineering and Knowledge Engineering	1	World Scientific
S[4], S[23], S[35], S[37],	Requirements Engineering: Foundation for Software Quality, LNCS	4	Springer
S[24]	Wuhan University, Wuhan, China, Tech. Rep	1	Wuhan University
S[25]	Proceedings of the 27th Annual ACM Symposium on Applied Computing	1	ACM
S[26]	International Conference on Software and System Process (ICSSP)	1	IEEE
S[27]	The Journal of Systems and Software	1	Elsevier
S[29]	Third International Workshop on Empirical Requirements Engineering (EmpiRE)	1	IEEE
S[31]	5th International Symposium, SSBSE 2013	1	Springer
S[34]	International Journal of Software Engineering and its Applications	1	SERSC
S[36]	Brazilian Symposium on Software Engineering	1	IEEE
S[39]	International Journal of Advanced Research in Computer Science	1	-

Table 12 highlights the detailed description of the universities, research centres and countries who have written SLRs in the domain of RE. The name and the country of the research institute is selected based on the corresponding author in case if there is a corresponding author otherwise, the affiliations are shown based on the first author.

Table. 12 Active research institutes in SLR on software requirements engineering

Affiliations	Research Studies	Total
Blekinge Institute of Technology, Sweden	S[1], S[2], S[13], S[19], S[22], S[28]	6
University of Alabama, Computer Science, USA	S[3]	1
Universidad Politécnica de Valencia, Spain	S[4], S[7], S[10]	3
Chalmers University of Technology, Sweden	S[5]	1
Universidad de Murcia, Spain	S[6]	1
Lund University, Lund, Sweden	S[8]	1
Universidad de Brasília, Brazil	S[9]	1
Johannes Kepler University, Austria	S[11]	1
University of Castilla-La Mancha, Spain	S[12], S[20]	2
University of Twente, Netherlands	S[14], S[23]	2
Universidad Politécnica de Madrid, Spain	S[15]	1
University of Ottawa, Ottawa, Canada	S[16]	1
Simula Research Laboratory, University of Oslo, Norway	S[17]	1
Chalmers and Gothenburg University, Sweden	S[18]	1
Institute of Software, Chinese Academy of Sciences, China	S[21]	1
Wuhan University, Wuhan, China	S[24]	1
Federal University of Pernambuco Recife, Brazil	S[25]	1
University of Calgary, Canada	S[26]	1
Technological University of the Mixtec Region, Mexico	S[27]	1
Free University of Bozen-Bolzano, Italy	S[29]	1
University of Mannheim, Germany	S[30]	1
Universiti Teknologi Malaysia, Johor, Malaysia	S[32], S[33]	2
Federal University of Bahia Brazil	S[31]	1
Xavier Institute of Management Bhubaneswar, India	S[34], S[38]	2
Institute of Mathematics, Academy of Maths and Syst. Sci., Chinese Academy of Sciences, China	S[35]	1
Universidade Federal de Campina Grande. Campina Grande, PB, Brasil	S[36]	1
Fraunhofer IESE, Fraunhofer Platz 1, 67663 Kaiserslautern, Germany	S[37]	1
Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados, CITMATEL, Cuba	S[39]	1

Table 13 highlights the research in the domain of RE SLRs based on countries and shows the leading countries at the top. Sweden, Spain and Brazil are the leading countries that are conducting research in the domain of RE SLRs. However, the countries considered as main hub for research like China and USA have published a small number of SLRs in the domain of RE.

Table. 13 Country wise SLR research

Country		Total
Sweden		9
Spain		7
Brazil		4
China		3
Germany		2
Malaysia	-	2
Netherlands		2

Canada	2
India	2
USA	1
Austria	1
Norway	1
Mexico	1
Italy	1

4. CONCLUSION

This research paper serves as a mini encyclopaedia for fresh or inexpert researchers in the domain of RE. Section 4 discusses the different research directions that may be considered by newbies to carry out research in RE. Based on the data, described above in Section 3 especially in RQ1 we may consider different research terminologies that may help to inexpert researchers to understand these research terms. After getting a knowledge of existing objectives and sub-objectives mentioned in the RQ1 can be reconsidered in order to improve RE research in the form of enhancements. The purpose of this research is to present a holistic view of the goals, objectives and issues related to the software requirements engineering in different perspectives. Research has been carried out in all spheres of RE and different articles, conference papers, reviews and systematic reviews have been written in this important domain. However, the aggregation of systematic reviews about RE has not been presented previously at any forum. RE is considered as vital for the success and failure of a software system. The research studies are carried out with a number of objectives and sub-objectives to present knowledge to the intended community. It is too laborious to go through all the SLRs and find out the focus objectives and sub-objectives of different SLRs. Moreover, with these objectives and sub-objectives of the research dimension different issues related to RE are also highlighted. There was a dire need to aggregate the presented knowledge in the field of RE with respect to the SLRs. A total of 39 key SLRs were found and the knowledge about objectives, sub-objectives and issues is presented in the form of a new SLR in this research study. This research study will serve as a mini encyclopaedia for new researchers to easily find out the objectives and issues focused in the previous SLRs and may choose research directions for their future research.

ACKNOWLEDGEMENT

Special thanks to Army Public College of Management & Sciences, Rawalpindi, Pakistan for providing resources to carry out this research. Moreover, special thanks to Prof. Barbra Kitchenham for her kind support in completion of this research.

REFERENCES

1. Brooks, F.P., *No silver bullet: Essence and accidents of software engineering.* IEEE computer, 1987. **20**(4): p. 10-19.
2. Zave, P., *Classification of research efforts in requirements engineering.* ACM Computing Surveys (CSUR), 1997. **29**(4): p. 315-321.
3. Gilb, T., *Competitive engineering: a handbook for systems engineering, requirements engineering, and software engineering using Planguage*. 2005: Butterworth-Heinemann.
4. Ballejos, L.C., S.M. Gonnet, and J.M. Montagna, *A Stakeholder Model for Interorganizational Information Systems.*, in *Requirements Engineering: Foundation for Software Quality*. 2008b, Springer: Berlin / Heidelberg. p. 73-87.
5. Ballejos, L.C. and J.M. Montagna, *Method for stakeholder identification in interorganizational environments.* Requirements engineering, 2008a. **13**(4): p. 281-297.
6. Ballejos, L.C. and J.M. Montagna, *Modeling stakeholders for information systems design processes.* Requirements engineering, 2011. **16**(4): p. 281-296.
7. Ballejos, L.C. and J.M. Montagna, *Stakeholders Selection for Interorganizational Systems: A Systematic Approach*, in *The Past and Future of Information Systems: 1976–2006 and Beyond*, S.E. David Avison, John Krogstie, Jan Pries-Heje, Editor. 2006, Springer. p. 39-50.
8. Mitchell, R.K., B.R. Agle, and D.J. Wood, *Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts.* Academy of management review, 1997: p. 853-886.

9. Fuentes-Fernández, R., J.J. Gómez-Sanz, and J. Pavón, *Understanding the human context in requirements elicitation.* Requirements engineering, 2010. **15**(3): p. 267-283.
10. Preiss, O. and A. Wegmann. *Stakeholder discovery and classification based on systems science principles*. in *Second Asia-Pacific Conference on Quality Software*. 2001. Hong Kong: IEEE.
11. Boonstra, A., *Interpreting an ERP-implementation project from a stakeholder perspective.* International Journal of Project Management, 2006. **24**(1): p. 38-52.
12. Coakes, E. and T. Elliman, *Focus issue on legacy information systems and business process engineering: the role of stakeholders in managing change.* Communications of the AIS, 1999. **2**(1es): p. 4.
13. Pan, G.S. and D. Flynn, *Information systems project abandonment: a case of political influence by the stakeholders.* Technology Analysis & Strategic Management, 2003. **15**(4): p. 457-466.
14. Pouloudi, A., *Stakeholder analysis as a front-end to knowledge elicitation.* AI & Society, 1997. **11**(1): p. 122-137.
15. Whitley, E.A., A. Poulymenakou, and T. Cornford, *The spring model for knowledge-based systems analysis: a case study.* ACM Sigmis Database, 1992. **23**(2): p. 1-5.
16. McManus, J. *A stakeholder perspective within software engineering projects*. in *IEEE International Engineering Management Conference*. 2004: IEEE.
17. Glinz, M. and R.J. Wieringa, *Guest Editor's Introduction: Stakeholders in Requirements Engineering.* IEEE Software, 2007. **24**(2): p. 18-20.
18. Razali, R. and F. Anwar, *Selecting the Right Stakeholders for Requirements Elicitation: A Systematic Approach.* Journal of Theoretical and Applied Information Technology, 2011. **33**(2).
19. Power, K. *Stakeholder Identification in Agile Software Product Development Organizations: A Model for Understanding Who and What Really Counts*. in *Agile Conference (AGILE)*. 2010: IEEE.
20. Lim, S.L., D. Quercia, and A. Finkelstein. *StakeNet: using social networks to analyse the stakeholders of large-scale software projects.* in *32nd ACM/IEEE International Conference on Software Engineering.* 2010: ACM.
21. Dieste, O. and N. Juristo, *Systematic review and aggregation of empirical studies on elicitation techniques.* IEEE Transactions on Software Engineering, 2011. **37**(2): p. 283-304.
22. Fowlkes, J.E., et al., *The utility of event-based knowledge elicitation.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 2000. **42**(1): p. 24-35.
23. Hudlicka, E. *Requirements elicitation with indirect knowledge elicitation techniques: comparison of three methods*. in *Requirements Engineering, 1996., Proceedings of the Second International Conference on*. 1996: IEEE.
24. Johnson, L. and N.E. Johnson, *Knowledge elicitation involving teachback interviewing*, in *Knowledge acquisition for expert systems*. 1987, Springer. p. 91-108.
25. Ericsson, K.A. and H.A. Simon, *Protocol analysis*. 1985: MIT press.
26. LaFrance, M., *The knowledge acquisition grid: A method for training knowledge engineers.* International Journal of Man-Machine Studies, 1987. **26**(2): p. 245-255.
27. Wood, J. and D. Silver, *Joint application development*. 1995: John Wiley & Sons, Inc.
28. Korhonen, L., *Nominal group technique.* Adult learning methods, 1990: p. 247-259.
29. Massey, A.P. and W.A. Wallace, *Focus groups as a knowledge elicitation technique: an exploratory study.* Knowledge and Data Engineering, IEEE Transactions on, 1991. **3**(2): p. 193-200.
30. Adelman, L., P.J. Sticha, and M.L. Donnell, *An experimental investigation of the relative effectiveness of two techniques for structuring multiattributed hierarchies.* Organizational Behavior and Human Decision Processes, 1986. **37**(2): p. 188-196.
31. Frankel, S., *NGT+ MDS: An adaptation of the nominal group technique for ill-structured problems.* Journal of Applied Behavioral Science, 1987.
32. Sutcliffe, A. *A technique combination approach to requirements engineering*. in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. 1997: IEEE.
33. Ruhe, G., *Software engineering decision support–a new paradigm for learning software organizations*, in *Advances in Learning Software Organizations*. 2003, Springer. p. 104-113.
34. Ruhe, G., A. Eberlein, and D. Pfahl, *Trade-off analysis for requirements selection.* International Journal of Software Engineering and Knowledge Engineering, 2003. **13**(04): p. 345-366.

35. Saaty, T.L., *The analytic hierarchy process: planning, priority setting, resources allocation.* M cGraw-Hill, 1980.
36. Karlsson, J. and K. Ryan, *A cost-value approach for prioritizing requirements.* IEEE software, 1997. **14**(5): p. 67-74.
37. Beg, R., Q. Abbas, and R.P. Verma. *An approach for requirement prioritization using b-tree*. in *First International Conference on Emerging Trends in Engineering and Technology.* 2008. Nagpur, Maharashtra: IEEE.
38. Leffingwell, D. and D. Widrig, *Managing software requirements: a unified approach*. 2000: Addison-Wesley Professional.
39. Aurum, A. and C. Wohlin, *Engineering and managing software requirements*. Vol. 1. 2005: Springer.
40. Lauesen, S., *Software requirements: styles and techniques*. 2002: Pearson Education.
41. Bergin, J., *Learning the Planning Game: an extreme exercise*. 2001.
42. Boehm, B.W. and R. Ross, *Theory-W software project management principles and examples.* IEEE Transactions on Software Engineering, 1989. **15**(7): p. 902-916.
43. Ramzan, M., M.A. Jaffar, and A.A. Shahid, *Value based Intelligent Requirement Prioritization (VIRP): Expert Driven Fuzzy Logic based Prioritization Technique.* International Journal of Innovative Computing, Information and Control (IJICIC).(Accepted), 2010.
44. Kitchenham, B., *Procedures for performing systematic reviews.* Keele, UK, Keele University, 2004. **33**: p. 2004.
45. Babar, M.I., M. Ghazali, and D.N. Jawawi, *Systematic Reviews in Requirements Engineering: A Systematic Review*, in *Mylasian Software Engineering Conference MySEC*. 2014: Langkwawi. p. 6.
46. Khan, K.A., *A systematic review of software requirements prioritization.*, in *Blekinge Institute of Technology, Ronneby, Sweden.* 2006.
47. Ma, Q., *The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review*. 2009, Auckland University of Technology.
48. Walia, G.S. and J.C. Carver, *A systematic literature review to identify and classify software requirement errors.* Information and Software Technology, 2009. **51**(7): p. 1087-1109.
49. Blanes, D., E. Insfran, and S. Abrahão, *Requirements engineering in the development of multi-agent systems: a systematic review*, in *Intelligent Data Engineering and Automated Learning-IDEAL 2009*. 2009, Springer. p. 510-517.
50. Ivarsson, M. and T. Gorschek, *Technology transfer decision support in requirements engineering research: a systematic review of REj.* Requirements engineering, 2009. **14**(3): p. 155-175.
51. Nicolás, J. and A. Toval, *On the generation of requirements specifications from software engineering models: A systematic literature review.* Information and Software Technology, 2009. **51**(9): p. 1291-1307.
52. Condori-Fernandez, N., et al. *A systematic mapping study on empirical evaluation of software requirements specifications techniques*. in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. 2009: IEEE Computer Society.
53. Svensson, R.B., M. Host, and B. Regnell. *Managing quality requirements: A systematic review*. in *36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA).* 2010: IEEE.
54. Alves, V., et al., *Requirements engineering for software product lines: A systematic literature review.* Information and Software Technology, 2010. **52**(8): p. 806-820.
55. Loniewski, G., E. Insfran, and S. Abrahão, *A systematic review of the use of requirements engineering techniques in model-driven development*, in *Model driven engineering languages and systems*. 2010, Springer. p. 213-227.
56. Rabiser, R., P. Grünbacher, and D. Dhungana, *Requirements for product derivation support: Results from a systematic literature review and an expert survey.* Information and Software Technology, 2010. **52**(3): p. 324-346.
57. Mellado, D., et al., *A systematic review of security requirements engineering.* Computer Standards & Interfaces, 2010. **32**(4): p. 153-165.
58. Jan, N. and M. Ibrar, *Systematic mapping of value-based software engineering-a systematic review of value-based requirements engineering*. 2010, Technical report, Master Thesis no: MSE-2010: 40, Karlskrona, Sweden.
59. Wohlin, C., *Engineering and managing software requirements*. 2005: Springer Science & Business Media.

60. Ghanavati, S., D. Amyot, and L. Peyton. *A systematic review of goal-oriented requirements management frameworks for business process compliance*. in *Fourth International Workshop on Requirements Engineering and Law (RELAW).* 2011: IEEE.
61. Yue, T., L.C. Briand, and Y. Labiche, *A systematic review of transformation approaches between user requirements and analysis models.* Requirements engineering, 2011. **16**(2): p. 75-99.
62. Barmi, Z.A., A.H. Ebrahimi, and R. Feldt. *Alignment of requirements specification and testing: A systematic mapping study*. in *IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (ICSTW).* 2011: IEEE.
63. Saha, S.K., et al. *A systematic review on creativity techniques for requirements engineering*. in *International Conference on Informatics, Electronics & Vision (ICIEV).* 2012: IEEE.
64. Martínez-Ruiz, T., et al., *Requirements and constructors for tailoring software processes: a systematic literature review.* Software Quality Journal, 2012. **20**(1): p. 229-260.
65. Li, J., et al. *Preliminary results of a systematic review on requirements evolution*. in *16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)*. 2012: IET.
66. Torkar, R., et al., *Requirements traceability: a systematic review and industry case study.* International Journal of Software Engineering and Knowledge Engineering, 2012. **22**(03): p. 385-433.
67. Teka, A.Y., N. Condori-Fernandez, and B. Sapkota, *A systematic literature review on service description methods*, in *Requirements Engineering: Foundation for Software Quality*. 2012, Springer. p. 239-255.
68. Lai, H., et al., *A systematic review of Re-specific wikis for distributed requirements engineering.* Wuhan University, Wuhan, China, Tech. Rep, 2012. **2012530**.
69. Lemos, J., et al. *A systematic mapping study on creativity in requirements engineering*. in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. 2012: ACM.
70. Pacheco, C. and I. Garcia, *A systematic literature review of stakeholder identification methods in requirements elicitation.* Journal of Systems and Software, 2012. **85**(9): p. 2171-2181.
71. Riņķevičs, K. and R. Torkar, *Equality in cumulative voting: A systematic review with an improvement proposal.* Information and Software Technology, 2013. **55**(2): p. 267-287.
72. Pergher, M. and B. Rossi. *Requirements prioritization in software engineering: A systematic mapping study*. in *Third International Workshop on Empirical Requirements Engineering (EmpiRE), 2013 IEEE* 2013: IEEE.
73. Meth, H., M. Brhel, and A. Maedche, *The state of the art in automated requirements elicitation.* Information and Software Technology, 2013. **55**(10): p. 1695-1709.
74. Pitangueira, A.M., et al., *A Systematic Review of Software Requirements Selection and Prioritization Using SBSE Approaches*, in *Search Based Software Engineering*. 2013, Springer. p. 188-208.
75. Achimugu, P., et al., *A systematic literature review of software requirements prioritization research.* Information and Software Technology, 2014. **56**(6): p. 568-585.
76. Babar, M.I., et al., *Stakeholder management in value-based software development: systematic review.* IET Software, 2014.
77. Thakurta, R., *Research Trends on Software Requirement Prioritization.* International Journal of Software Engineering and Its Applications, 2014. **8**(6): p. 287-298.
78. Yang, Z., et al., *A Systematic Literature Review of Requirements Modeling and Analysis for Self-adaptive Systems*, in *Requirements Engineering: Foundation for Software Quality*. 2014, Springer. p. 55-71.
79. Dermeval, D., et al. *A systematic review on the use of ontologies in requirements engineering*. in *Software Engineering (SBES), 2014 Brazilian Symposium on*. 2014: IEEE.
80. Riegel, N. and J. Doerr. *A systematic literature review of requirements prioritization criteria*. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2015: Springer.
81. Thakurta, R., *Understanding requirement prioritization artifacts: a systematic mapping study.* Requirements engineering, 2016: p. 1-36.
82. Ing Marelis V., P.G., V.R. Ing. Yordan, and M.A. Ahmad, *Evaluation of the Non-Functional Requirements of Usability: A Systematic Study.* International Journal of Advanced Research in Computer Science, 2016. **7**(3): p. 29-35.
83. Iankoulova, I. and M. Daneva. *Cloud computing security requirements: A systematic review*. in *Research Challenges in Information Science (RCIS), 2012 sixth international conference on*. 2012: IEEE.

84. Mohebzada, J.G., G. Ruhe, and A. Eberlein. *Systematic mapping of recommendation systems for requirements engineering*. in *Software and System Process (ICSSP), 2012 International Conference on*. 2012: IEEE.
85. Shaw, M. *The coming-of-age of software architecture research*. in *Proceedings of the 23rd international conference on Software engineering*. 2001: IEEE Computer Society.
86. Sommerville and P. Sawyer, *Requirements Engineering -A Good Practice Guide*. 1997, Chichester, UK.: John Wiley and Sons.
87. Sommerville and G. Kotonya, *Requirements engineering: processes and techniques*. 1998: John Wiley & Sons, Inc.
88. Chung, L. and J.C.S. do Prado Leite, *On non-functional requirements in software engineering*, in *Conceptual modeling: Foundations and applications*. 2009, Springer. p. 363-379.
89. Nuseibeh, B. and S. Easterbrook. *Requirements engineering: a roadmap*. in *Proceedings of the Conference on the Future of Software Engineering*. 2000: ACM.
90. Sommerville, I., *Software Engineering*. 6th ed. 2002, Wokingham, England: Addison-Wesley.
91. Pacheco, C. and E. Tovar. *Stakeholder identification as an issue in the improvement of software requirements quality*. in *Advanced Information Systems Engineering*. 2007: Springer.
92. Pacheco, C. and I. Garcia. *Stakeholder Identification Methods in Software Requirements: Empirical Findings Derived from a Systematic Review*. 2008: IEEE.
93. Pacheco, C. and I. Garcia. *Effectiveness of Stakeholder Identification Methods in Requirements Elicitation: Experimental Results Derived from a Methodical Review*. 2009: Ieee.
94. Ballejos, L.C., S.M. Gonnet, and J.M. Montagna, *A Stakeholder Model for Interorganizational Information Systems*, in *Requirements Engineering: Foundation for Software Quality*, C.R. Barbara Paech, Editor. 2008b, Springer: Berlin / Heidelberg. p. 73-87.
95. Macaulay, L.A., *Requirements engineering*. 1996: Springer-Verlag.
96. Robertson, S., *Requirements trawling: techniques for discovering requirements.* International Journal of Human-Computer Studies, 2001. **55**(4): p. 405-421.
97. Brooks, F.P., *The mythical man-month: essays on software engineering*. 1995: Addison-Wesley.
98. Masuck, C., *Incorporating a fault categorization and analysis process in the software build cycle.* Journal of Computing Sciences in Colleges, 2005. **20**(5): p. 239-248.
99. Karlsson, J., C. Wohlin, and B. Regnell, *An evaluation of methods for prioritizing software requirements.* Information and Software Technology, 1998. **39**(14-15): p. 939-947.
100. Laurent, P., J. Cleland-Huang, and C. Duan. *Towards automated requirements triage*. in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*. 2007: IEEE.
101. Karlsson, J. and K. Ryan, *A cost-value approach for prioritizing requirements.* Software, IEEE, 1997. **14**(5): p. 67-74.
102. Viega, J. *Building security requirements with CLASP*. in *ACM SIGSOFT Software Engineering Notes*. 2005: ACM.
103. Boehm, B.W., R.K. Mcclean, and D. Urfrig, *Some experience with automated aids to the design of large-scale reliable software.* IEEE transactions on software engineering, 1975(1): p. 125-133.
104. Tavolato, P. and K. Vincena, *A prototyping methodology and its tool*, in *Approaches to prototyping*. 1984, Springer. p. 434-446.
105. Cysneiros, L.M. and J.C.S. do Prado Leite, *Nonfunctional requirements: From elicitation to conceptual models.* IEEE transactions on software engineering, 2004. **30**(5): p. 328-350.
106. Breitman, K.K., J.C.S. Leite, and A. Finkelstein, *The world sa stage: a survey on requirements engineering using a real-life case study.* Journal of the Brazilian Computer Society, 1999. **6**(1): p. 13-37.
107. Babar, M.I., M. Ramzan, and S. Ghayyur. *Challenges and future trends in software requirements prioritization*. in *Computer Networks and Information Technology (ICCNIT), 2011 International Conference on*. 2011: IEEE.
108. Finkelstein, A. and J. Dowell. *A comedy of errors: the London Ambulance Service case study*. in *Proceedings of the 8th International Workshop on Software Specification and Design*. 1996: IEEE Computer Society.
109. Ning, A., et al. *Requirements engineering processes improvement: a systematic view*. in *Software Process Workshop*. 2005: Springer.

110. Chung, L., et al., *Non-functional requirements in software engineering*. Vol. 5. 2012: Springer Science & Business Media.
111. Jacobs, S. *Introducing measurable quality requirements: a case study*. in *Requirements Engineering, 1999. Proceedings. IEEE International Symposium on*. 1999: IEEE.
112. Olsson, T., R. Berntsson Svensson, and B. Regnell. *Non-functional requirements metrics in practice-an empirical document analysis*. in *Workshop on Measuring Requirements for Project and Product Success*. 2007.
113. Bell, T.E. and T.A. Thayer. *Software requirements: Are they really a problem?* in *Proceedings of the 2nd international conference on Software engineering*. 1976: IEEE Computer Society Press.
114. Valusek, J.R. and D.G. Fryback. *Information requirements determination: obstacles within, among and between participants*. in *Proceedings of the twenty-first annual conference on Computer personnel research*. 1985: ACM.
115. Juristo, N. and O. Dieste, *Systematic Review and Aggregation of Empirical Studies on Elicitation Techniques.* 2011.
116. Wiegers, K., *First things first: prioritizing requirements.* Software Development, 1999. **7**(9): p. 48-53.
117. Karlsson, L., et al., *Requirements prioritisation: an experiment on exhaustive pair-wise comparisons versus planning game partitioning.* 2004.
118. Ericsson, K.A. and H.A. Simon, *Protocol analysis*. 1993: MIT press Cambridge, MA.
119. Korhonen, L., *Nominal group technique.* Adult Learning Methods, ed. MW Galbraith, 1990: p. 247-259.
120. Massey, A.P. and W.A. Wallace, *Focus groups as a knowledge elicitation technique: an exploratory study.* IEEE Transactions on Knowledge and Data Engineering, 1991. **3**(2): p. 193-200.
121. Hotz, L., et al., *Configuration in industrial product families*. 2006: Ios Press.
122. Maiden, N. and A. Gizikis, *Where do requirements come from?* IEEE software, 2001. **18**(5): p. 10-12.
123. Grube, P.P. and K. Schmid. *Selecting creativity techniques for innovative requirements engineering*. in *2008 Third International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games*. 2008: IEEE.
124. Walton, J.P. *Developing an enterprise information security policy*. in *Proceedings of the 30th annual ACM SIGUCCS conference on User services*. 2002: ACM.
125. Jan, N. and M. Ibrar, *Systematic Mapping of Value-based Software Engineering: A Systematic Review of Value-based Requirements Engineering.* 2010.
126. Iankoulova, I. and M. Daneva. *Cloud computing security requirements: A systematic review*. in *2012 Sixth International Conference on Research Challenges in Information Science (RCIS)*. 2012: IEEE.
127. Ebert, C. and R.J. Wieringa, *Requirements engineering: Solutions and trends*, in *Engineering and managing software requirements*. 2005, Springer. p. 453-476.
128. Ghanavati, S., D. Amyot, and L. Peyton. *A systematic review of goal-oriented requirements management frameworks for business process compliance*. in *Requirements Engineering and Law (RELAW), 2011 Fourth International Workshop on*. 2011: IEEE.
129. Hogarth, R., *Judgement and Choice*. 2nd Edition ed. 1987, New York: Wiley.
130. Nakashima, T., et al., *Analysis of software bug causes and its prevention.* Information and Software Technology, 1999. **41**(15): p. 1059-1068.
131. Basili, V.R. and H.D. Rombach. *Tailoring the software process to project goals and environments*. in *Proceedings of the 9th international conference on Software Engineering*. 1987: IEEE Computer Society Press.
132. Card, D.N., *Learning from our mistakes with defect causal analysis.* IEEE software, 1998. **15**(1): p. 56-63.
133. Grady, R.B., *Software failure analysis for high-return process improvement decisions.* Hewlett Packard Journal, 1996. **47**: p. 15-24.
134. Jacobs, J., et al., *Exploring defect causes in products developed by virtual teams.* Information and Software Technology, 2005. **47**(6): p. 399-410.
135. Mays, R.G., et al., *Experiences with defect prevention.* IBM Systems Journal, 1990. **29**(1): p. 4-32.
136. Kan, S., V.R. Basili, and L.N. Shapiro, *Software quality: an overview from the perspective of total quality management.* IBM Systems Journal, 1994. **33**(1): p. 4-19.
137. Leszak, M., D.E. Perry, and D. Stoll. *A case study in root cause defect analysis*. in *Proceedings of the 22nd international conference on Software engineering*. 2000: ACM.

138. Hsia, P., A. Davis, and D. Kung, *Status report: requirements engineering.* IEEE software, 1993. **10**(6): p. 75-79.
139. Cheng, B.H. and J.M. Atlee. *Research directions in requirements engineering*. in *2007 Future of Software Engineering*. 2007: IEEE Computer Society.
140. Kleppe, A.G., J.B. Warmer, and W. Bast, *MDA explained: the model driven architecture: practice and promise*. 2003: Addison-Wesley Professional.
141. Arkley, P. and S. Riddle. *Overcoming the traceability benefit problem*. in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. 2005: IEEE.
142. Cleland-Huang, J., C.K. Chang, and M. Christensen, *Event-based traceability for managing evolutionary change.* IEEE transactions on software engineering, 2003. **29**(9): p. 796-810.
143. Cleland-Huang, J., G. Zemont, and W. Lukasik. *A heterogeneous solution for improving the return on investment of requirements traceability*. in *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*. 2004: IEEE.
144. Cleland-Huang, J., et al. *Utilizing supporting evidence to improve dynamic requirements traceability*. in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. 2005: IEEE.
145. Heindl, M. and S. Biffl. *A case study on value-based requirements tracing*. in *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*. 2005: ACM.
146. Gotel, O. and S.J. Morris. *Crafting the requirements record with the informed use of media*. in *Multimedia Requirements Engineering, 2006. MERE'06. First International Workshop on*. 2006: IEEE.
147. Ravichandar, R., J.D. Arthur, and M. Pérez-Quinones, *Pre-requirement specification traceability: Bridging the complexity gap through capabilities.* arXiv preprint cs/0703012, 2007.
148. Blaauboer, F., K. Sikkel, and M.N. Aydin. *Deciding to adopt requirements traceability in practice*. in *International Conference on Advanced Information Systems Engineering*. 2007: Springer.
149. Cleland-Huang, J. *Toward improved traceability of non-functional requirements*. in *Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering*. 2005: ACM.
150. Ramesh, B., *Factors influencing requirements traceability practice.* Communications of the ACM, 1998. **41**(12): p. 37-44.
151. Tvete, B. *Introducing efficient requirements management*. in *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on*. 1999: IEEE.
152. Gotel, O. and A. Finkelstein. *Extended requirements traceability: Results of an industrial case study*. in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*. 1997: IEEE.
153. Klein, M., B. König-Ries, and P. Obreiter. *Stepwise refinable service descriptions: Adapting DAML-S to staged service trading*. in *International Conference on Service-Oriented Computing*. 2003: Springer.
154. Yun, B., J. Yan, and M. Liu. *Behavior-based web services matchmaking*. in *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. 2008: IEEE.
155. Hartmann, J., et al. *Formal incremental requirements specification of service-oriented automotive software systems*. in *Service-Oriented System Engineering, 2006. SOSE'06. Second IEEE International Workshop*. 2006: IEEE.
156. Tsai, W.-T., et al. *Global software enterprise: A new software constructing architecture*. in *E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on*. 2006: IEEE.
157. Tsai, W.-T., et al. *Consumer-centric service-oriented architecture: A new approach*. in *Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on*. 2006: IEEE.
158. Yau, S.S. and J. Liu. *Incorporating situation awareness in service specifications*. in *Object and Component-Oriented Real-Time Distributed Computing, 2006. ISORC 2006. Ninth IEEE International Symposium on*. 2006: IEEE.
159. Martin, D., et al. *Bringing semantics to web services: The OWL-S approach*. in *International Workshop on Semantic Web Services and Web Process Composition*. 2004: Springer.
160. Sirin, E., J. Hendler, and B. Parsia. *Semi-automatic composition of web services using semantic descriptions*. in *1st Workshop on Web Services: Modeling, Architecture and Infrastructure*. 2003.

161. Di Modica, G., et al. *Enabling re-negotiations of SLA by extending the WS-Agreement specification*. in *Services Computing, 2007. SCC 2007. IEEE International Conference on*. 2007: IEEE.
162. Cardoso, J., et al. *Towards a unified service description language for the internet of services: Requirements and first developments*. in *Services Computing (SCC), 2010 IEEE International Conference on*. 2010: IEEE.
163. Rychlý, M. and P. Weiss. *Modeling of Service Oriented Architecture-From Business Process to Service Realisation*. in *ENASE*. 2008.
164. Shishkov, B., J.L. Dietz, and M. van Sinderen, *Closing the business-application gap in SOA.* 2007.
165. Rolland, C., M. Kirsch-Pinheiro, and C. Souveyet, *An intentional approach to service engineering.* IEEE Transactions on Services Computing, 2010. **3**(4): p. 292-305.
166. Papazoglou, M., *Web services: principles and technology*. 2008: Pearson Education.
167. Slimane, A.A.A., M.K. Pinheiro, and C. Souveyet. *Goal reasoning for quality elicitation in the ISOA approach*. in *Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on*. 2009: IEEE.
168. Stefanovic, M., et al., *Method of design and specification of web services based on quality system documentation.* Information Systems Frontiers, 2009. **11**(1): p. 75-86.
169. Qiu, Q. and Q. Xiong. *An ontology for semantic web services*. in *International Conference on High Performance Computing and Communications*. 2007: Springer.
170. Kritikos, K. and D. Plexousakis, *Requirements for QoS-based web service description and discovery.* IEEE Transactions on Services Computing, 2009. **2**(4): p. 320-337.
171. Pfeffer, H., et al. *Towards light-weight semantic descriptions for decentralized service-oriented systems*. in *Semantic Computing, 2007. ICSC 2007. International Conference on*. 2007: IEEE.
172. Bocciarelli, P. and A. D'Ambrogio, *A model-driven method for describing and predicting the reliability of composite services.* Software & Systems Modeling, 2011. **10**(2): p. 265-280.
173. Di Marco, A. and A. Sabetta. *Model-based dynamic QoS-driven service composition*. in *Proceedings of the 2nd International Workshop on the Quality of Service-Oriented Software Systems*. 2010: ACM.
174. Fornasier, P., J. Webber, and I. Gorton. *Soya: a programming model and runtime environment for component composition using SSDL*. in *International Symposium on Component-Based Software Engineering*. 2007: Springer.
175. Jiménez, M., M. Piattini, and A. Vizcaíno, *Challenges and improvements in distributed software development: A systematic review.* Advances in Software Engineering, 2009. **2009**: p. 3.

AUTHORS PROFILE

Dr. Muhammad Imran Babar received his MCS from Al-Khair University AJK Pakistan. He received his MS degree in Software Engineering from International Islamic University Islamabad Pakistan. He received his Ph.D. degree in Computer Science from Universiti Teknologi Malaysia (UTM) in 2015. Currently, he is serving as an Assistant Professor at Army Public College of Management & Sciences (APCOMS) Rawalpindi, Pakistan. APCOMS is an affiliate institute of University of Engineering and Technology Taxila, Pakistan. Moreover, he is Head of Department in Department of Computer Science at APCOMS. His research interests are in software engineering, decision support systems, artificial intelligence, neural networks, predictive analytics, fuzzy logic and big data. He is editor in chief of Journal of Software Engineering and Intelligent Systems and Journal of Theoretical and Applied Information Technology.

Dr. Masitah Ghazali received her BE degree in software engineering from University of Manchester Institute of Science and Technology (UMIST), UK. She received her Master degree in Distributed Interactive Systems from Lancaster University, Lancaster UK. She earned her Ph.D. degree in Computer Science from Lancaster University, Lancaster UK. She is a Senior Lecturer at the Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia (UTM). Her research interests include human-computer interaction, software engineering, usability, human factors, interaction design, physicality, and user interfaces.

Dr. Imran Ghani is a Senior Lecturer at Monash University Malaysia Campus Malaysia. He received his Master of Information Technology degree from UAAR Pakistan, M.Sc. Computer Science from UTM (Malaysia) and Ph.D. from Kookmin University, South Korea. His research interests are in software architecture and design, software engineering, secure software development, software application development using agile software development methods.

Falak Sher received his MCS from Al-Khair University AJK Pakistan. He received his MS degree in Software Engineering from International Islamic University Islamabad Pakistan. Currently, he is pursuing his Ph.D. in Computer Science from Universiti Teknologi Malaysia. His research interests are in software engineering, decision support systems, artificial intelligence, software requirements engineering, and software quality assurance. He is an experienced teacher of computing at Beaconhouse School System Islamabad, Pakistan and has a teaching experience of more than 15 years.

Dr. Muhammed Jehanzeb received his B.Sc. degree in Computer Science from Arid Agriculture University, Pakistan in 2005, and M.S. in Computer Science from Iqra University, Pakistan, in 2007. He earned Ph.D. degree in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia in 2014. Presently, he is working as an Assistant Professor at the Department of Computer Science in Army Public College of Management & Sciences (APCOMS), Pakistan. His research interests include data analysis, document analysis and recognition, pattern recognition, image analysis and classification.

Dr. Shahid Nazir Bhatti is currently working as Senior Assistant Professor in the Software Engineering Department at Bahria University Islamabad, Pakistan. He has obtained his PhD from Johannes Kepler University Linz, Austria in year 2007 in the area of Software Engineering. Dr. Bhatti has more than 14 years of teaching, research & industrial experience. Dr. Bhatti's current research activities are in the field of System Engineering, Requirement Engineering, Software Metrics & Applications, Software Quality Engineering, Information Systems & Software Reengineering.

Dr. Dayang N.A. Jawawi received her B.Sc. Degree in Software Engineering from Sheffield Hallam University, UK and M.Sc. and Ph.D. by research in software engineering from Universiti Teknologi Malaysia. She is an Associate Professor at the Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia (UTM). Her research areas are software reuse, component-based software engineering and embedded real-time software. In her Ph.D. works she proposed a practical approach for component-based software engineering of embedded software in resource constrained embedded systems such as autonomous mobile robots and intelligent instruments systems.

Dr. Awad Ali received the Ph.D. in Computer Science from University Teknologi Malaysia in 2016. He is now an Assistant Professor/Head of Computer Science Department at University of Kassala, Sudan. His research interest includes software quality, software modeling and component-based systems, software reliability and software engineering.

Dr. Furkh Zeshan holds a Ph.D. degree in Computer Science from University Teknologi Malaysia (UTM). He has six years of teaching experience and have authored twelve research papers. He is a review member of Aloy Journal of Soft Computing & Applications (AJSCA) and editorial board member of Journal of Software Engineering & Intelligent Systems (JSEIS). His research interests include service-oriented computing, embedded and real-time computing, agent-oriented software engineering, software project management and self-organizing systems. Currently, he is working as an Assistant Professor in the Department of Computer Science of COMSATS, Institute of Information Technology (CIIT), Lahore, Pakistan.