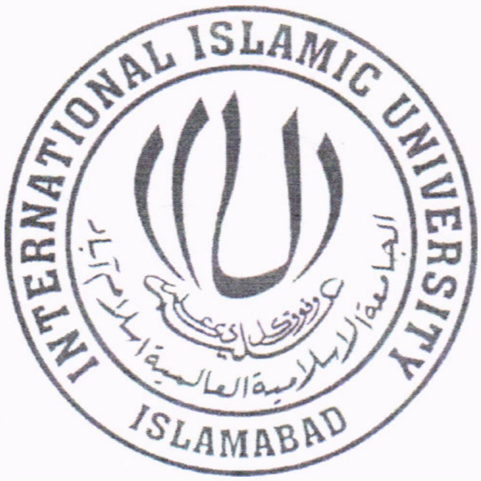
Software Risk Management: A Systematic Review



Submitted By:

Saima Irum

(135-FBAS/MSSE/F06)

Supervised By:

Dr. Naveed Ikram

Department of Computer Science & Software Engineering

Faculty of Basic & Applied Sciences

International Islamic University, Islamabad

(2013)

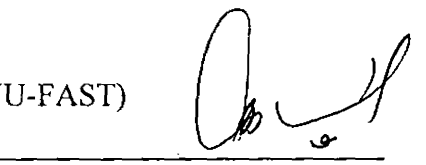
Dated:

FINAL APPROVAL

It is certified that we have read the thesis, entitled **"Software Risk Management: A Systematic Review"**, submitted by Saima Irum Reg. No. 135-FBAS/MSSE/F06. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for MS Degree in Software Engineering.

PROJECT EVALUATION COMMITTEE

External Examiner:
Professor Arshad Ali Shahid
Dean Faculty of Computing
National University of Computing and Emerging Science (NU-FAST)
Islamabad



Internal Examiner:
Muhammad Usman
Assisstant Professor
Department of CS & SE
IIU Islamabad.

Supervisor:
Dr. Naveed Ikram
Associate Professor
Faculty of Computing,
Riphah International University
Islamabad, Pakistan.

Acknowledgement:

I am very thankful to ALLAH who blessed me in the course of my life and especially during this thesis. It was only with Allah's help that I am able to complete my work. I would like to thank Dr. Naveed Ikram for his continued support, guidance and patience for the completion of this work. I express my gratitude to my parents, husband, friends, and peers for their extraordinary support during my research work.

Declaration:

I hereby declare and affirm that this thesis neither as a whole nor as part thereof has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, prepared under the sincere guidance of my supervisors. If any part of this report is proven to be copied out or found to be a reproduction of other's research work, we shall stand by the consequences. No portion of the work presented in this report has been submitted in support of an application for other degree or qualification of this or any other University or Institute of learning.

Saima Irum

135-FBAS/MSSE/F06

Dedication:

I dedicate my work to:

My Parents

&

My Teachers

Table of Contents

List of Tables

List of Figures

Abstract

Software risk management is essential for the successful delivery of software development projects. Risk management is a systematic and continuous process. Numerous studies have shown that risk management techniques get significant attentions because without managing risk properly very good project may fail. Software risk management has become one important topic in software engineering research. To depict the holistic state of the art of empirical work done in software risk management and to find out the strength of empirical literature in software risk management. This paper presents a systematic review on software risk management that was motivated by previous results obtained from piloting study. A total of 68 relevant studies were selected from an initial set of 622 studies in order to extract and synthesize empirical data concerning the state of the art of empirical work done in software risk management. We detected that software risk assessment is the most studied area of software risk management as proper methods, processes, frameworks, tools etc. exists for risk assessment. Our results suggests that there are several strategies for managing risks and in future, more research attention on software risk control techniques is needed.

Chapter 1: Introduction

1 Introduction

Software risk management is essential for the successful delivery of software development projects. During the last ten years, the software industry is paying more and more attention towards Software Risk Management (SRM). According to Barry W. Boehm in [1] "software risk management is an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful software operation or major source software rework".

Risk management is a systematic and continuous process. The risk management paradigm of Software Engineering Institute (SEI) describes this phenomenon more thoroughly. SEI paradigm involves some sequential, concurrent and iterative activities, to identify, plan, track, control and communication of risk [2]. Software Risk Management (SRM) is carried out with the help of different SRM processes, models, frameworks, tools, techniques and methodologies e.g., Software Risk Evaluation (SRE), Team Risk Management (TRM), Analyzer for Reducing Module Operational Risk (ARMOR), Expressing needs and Identifying Security Objectives (EBIOS) Methodology, ProRisk Framework, Riskit Method, SoftRisk, CMMI-RSKM, PMBOK RM Process, GDSP RM Framework, Risk and Performance Model [3][4]. Risk management techniques get significant attentions because without managing risk properly very good project may fail as SRM has become very crucial field at present, further, quality of product, timelines and product cost has become very serious issue. This is the main motivation that software industry is now considering the use of risk management techniques to avoid project failures, budget overruns and time limitation issues.

1.1 Problem Description

There exists lots of literature on diverse areas of software risk management, and each field provides the results of its own areas. Therefore, results of this field in general are very scattered. There is a need to summarize and aggregate the results of this vast field to find out actual trend of the field, identify gaps, scope for further research and quality of the work. This is the main motivation and reason to conduct this systematic literature review.

1.2 Research Objective

Software risk management is a broad field and includes many sub fields. like software risk assessment. identification, analysis, prioritization, planning, software risk monitoring, control and many more. Software industry is high risk business due to ever increasing software complexity and demand for better and faster software. Therefore software risk management is a active research area in this era.

Implementing software risk management is a crucial process because it requires resources, skills at all of the organizational levels and adequate knowledge of stakeholders. That is why literature has various studies and surveys on software risk management practices. However, there is no study in the literature with a focus on empirical evidences, up to our knowledge. Evaluating empirical evidence is equally important for academia and software industry, because gathering and summarizing empirical evidence systematically, can help researchers to :

A. Identify the sub-field of their research based on the recent research trends,
B. This research aims to identify the subject areas and to dig out the exhausted and least investigated research areas in the software risk management.
C. This research will provide a glimpse of the software risk management tools, techniques, processes, frameworks and risk factors which will be helpful for the selection of appropriate research methodology and research model on the particular topic of interest and subject scenario in software risk management.

1.3 Research Methodology

Systematic Literature Review (SLR) is the methodology used to collect empirical evidence reflected in software risk management literature. The SLR is conducted by using the guidelines of (Kitchenham B. 2004). In this SLR a detailed protocol is developed before conducting the review. This protocol provides detail plan of the whole review process. SLR is performed in three phases i.e. planning, conducting and reporting. SLR protocol is developed in planning phase. This protocol is executed in conducting the review and at the end the results are reported. The sub phases of SLR are as follows:

1.3.1 Planning the Review

- Identification of the need for a review
- Development of a review protocol.

1.3.2 Conducting the Review

- Formulating the research question(RQ)
- Search strategy
- Study selection criteria
- Study quality assessment
- Data extraction
- Data synthesis.

1.3.3 Reporting the Review

In this phase results of SLR are reported effectively to produce various research directions so that when the upcoming researchers use this work, they can consider various reflections of the fields and make decisions efficiently.

1.4 Thesis Structure

The structure of this thesis is as follows: *Chapter 2* describes the protocol defined to perform the SLR. Section 2.1 describes Research Questions; section 2.2 describes major search terms and their synonyms whereas section 2.3 and 2.4 describe search string and search sources. After the plan for conducting the SLR is devised, it is executed. *Chapter 3* describes the execution i.e., actual implementation of the protocol devised. Section 3.1 describes Results retrieved by applying search string to data sources, section 3.2 shows selected studies' citation and section 3.3 shows quality assessment score. *Chapter 4* describes the end results of the SLR and analysis of the results. Section 4.1 describes the Results. Section 4.2 describes the findings and section 4.3 and 4.4 describes the analysis. Section 5 concludes the thesis and lists the implications and future directions.

Chapter 2: Protocol Definition

2. Protocol Definition

This chapter describes the highlights of protocol defined to perform the Systematic Literature Review (SLR). SLR is conducted on the basis of predefined plan (protocol). Main sections of protocol are defined in this section and full protocol is provided in appendix A.

2.1 Background and Motivation

The main motive to perform this systematic review is to identify state-of-the-art and gaps in empirical research related to software risk management and summarize the existing empirical evidence to provide basis for future research and practical use. Related work exists in several studies [4-8] where researchers summarized the available literature and pointed out future directions but the focus of such studies was not related to finding the empirical evidence and none of the study reported qualitative and quantitative evaluation of data at a time. In these studies, only simple survey has been conducted to explore specific areas of risk management.

Software risk management is a vast field and all results are scattered regarding the subfields under the umbrella of SRM therefore, being motivated of the fact and to provide aggregate knowledge of this area a systematic review has been conducted. It will also provide directions for the future work and dig out the most and least exhaustive subject areas of software risk management.

Primary research on software risk management focused on defining guidelines for specific tasks [9], [10], [11], [12] but these provide little empirical evidence for the practical usefulness of risk management. Later in 1990's B.W. Boehm, Charette's, Software Engineering Institute (SEI) risk management methods and Hall's risk management principles have increased industry awareness and improved practice. SEI defines risk as the possibility of suffering loss. Richard E. Fairley defined 'risk' as "The probability of incurring a loss or enduring a negative impact" [13]. Software risks are increasing as long as software development industry is increasing [14]. Risks in software should be handled properly because they greatly affect software development process and its outcomes. Software development involves five types of risks: 1) financial risk; 2) technical risk; 3) project risk; 4) functionality risk; 5) political risk. Risk Management consists of

the processes, methodologies and tools that are used to handle with these risks in the Software Development Life Cycle (SDLC) process of a Software Project.

SEI risk program is very useful in order to improve the process of software acquisition and software development [15]. The basic methodological framework introduced for the software acquisition and software development are: Software Acquisition Capability Maturity Model (SA-CMM^{SM}) and Software Capability Maturity Model (SW-CMM^{SM}). These methodologies are supported by Software Risk Evaluation (SRE), Continuous Risk Management (CRM), Team Risk Management (TRM) practices. The SRE practice, developed by the SEI, is used for identifying, analyzing, communicating, and mitigating software technical risk. Its primary functions are Detection, Specification, Assessment, and Consolidation and supporting functions are Planning, Coordination, Verification, Training and Communication. The Continuous Risk Management (CRM) practice is used for managing project risks and opportunities throughout all the activities of the project. Team Risk Management (TRM) is a risk management based on team oriented activities in which both customer and supplier together apply the methodologies. SRE, CRM and TRM are based on three basic constructs of SEI these are Risk Management Paradigm, Risk Taxonomy, and Risk Clinic. While implementing these methodologies and practices SEI experience shows that software risk is the least measured and managed during the lifecycle of software development.

Current perceptions and emerging trends of various software risk management strategies are discussed as followed. Geoffrey G. Roy introduced a ProRisk framework for risk management [16]. ProRisk framework provides a process to analyze and identify the key risk factors, outcomes, reactions and the creation of action plan to mitigate these risks. ProRisk Framework is built on a hierarchical model structure defined in the SEI taxonomy of risk and Karolak [4] and involves following activities. 1) Stakeholder Identification, 2) Risk Factor Identification, 3) Risk Tree Model Construction, 4) Calibrating the Model, 5) Estimating the Risk Event Probabilities, 6) Computing Combined Risk Values, 7) Developing Action Plans, 8) Monitoring the Progress, 9) Operationalizing the Framework. This framework focuses on the business domain and the operational domain of the software development. In business domain level it identifies the economic environment of the organization and the weakness of the organization to expose risks

factor. It also identifies the knowledge, experience and confidence of the organization to successfully complete the project. In Operational domain it measures the risk values, identify the key risk factors, identify and describes the action plans to reduce the key risk factors, implement action plans and then re-assess the risk key factors. It is continuous cyclic process so it continuously monitor and document risk properties, and provide support for risk mitigation and management on a continuous basis. This framework can be applied to both small scale and complex projects, with controllable levels of data requirements. ProRisk framework covers the complete life cycle of the Project development and provides support to run risk analysis activities in parallel with the project management activities. It is also supported with a ProRisk tool.

Literature has many studies and surveys on software risk management practices. In [17] authors represent a study of various methods that are used in software risk management process from the first step of risk identification up to the last step of risk control. It provides a risk checklist for risk identification gives step by step procedure for the risk assessment and provides complete risk management plan. Another survey presented in [4] gives suggestions for the selection of best tools and techniques for software risk management. It provides the limitations and beneficial qualities of Software Risk Evaluation (SRE), Team Risk Management (TRM), Softrisk tool, ARMOR (Analyzer for Reducing Module Operational Risk), Riskit technique and CMM based risk control optimization model.

Implementing risk management means to insert the risk management principles and practices into software development life cycle. Best implementation strategy is to use incremental model because in this way risk management practices can easily be adjusted in organizational culture. [18]. Software Risk Management Practices in a Small Scale Project were accessed in [19]. They have selected a small scale project, purpose of the study was to access the strengths and weakness of the risk practices in that project. They have selected Capability Maturity Model Integration (CMMI) model to measure the gap between current software risk management practices in selected project. After assessment findings they also used Standard CMMI Appraisal Method for Process Improvement (SCAMPI) for the formal characterization. The characterization indicator indicates that the software risk management practices are Partially Implemented in that project.

The practical implementation of the ongoing processes of software risk management is challenging task. Organizations that implement effective tools and techniques in software development project are successful by changing organization culture [20]. Implementing software risk management is a crucial process because it requires resources, skills at all organizational levels and adequate knowledge of stakeholders. That's why literature has many studies and surveys on software risk management practices. But there is no study in the literature with a focus on empirical evidence. Evaluating empirical evidence is equally important for academia and software industry, as gathering and summarizing empirical evidence systematically will help researchers for future research and practitioners will get quantitative knowledgebase to make informed decisions.

2.2 SLR Protocol

Protocol is a detailed plan of the whole process of SLR which is refined by different people. At first a protocol was developed, this first draft was refined by the supervisor and by the external reviewer's comments. This chapter describes the phases of protocol development of the SLR defined in the final version of the protocol.

2.3 Research Questions

The two investigated questions are:

RQ.1: What is the state-of-the-art in empirical studies of software risk management?

RQ.2:.What is the strength of empirical evidence reflected in empirical software risk management literature?

2.4 Search Strategy

For our search strategy we have taken inspirations from the protocol of Sarah Beecham [21] and modified according to our requirements. Major search terms will be identified from the RQs and their synonyms and alternate spellings will be used to form the search strings.

2.4.1 Major Search Terms and Synonyms

The steps used for extracting search terms are as follows:

1. We will derive major search strings from PICO;
2. Identify alternative spellings and synonyms for major terms; also alternative terms used in literature will be considered
3. When database allows, use the Boolean OR to incorporate alternative spellings and synonyms.
4. When database allows, use the Boolean AND to link the major terms from population, intervention and outcome.

In order to answer the stated research questions, search strategy need to be defined before conducting the review. Research studies based on empirical evidence, with either professional software developers or students as participants, were the main focus of this literature review. Studies focusing software risk management were considered. The final search string was selected on the basis of experience from the pilot search.

Major search terms for the research questions are as follows:

RQ1:

- Software Risk Management
- SRM

RQ2:

- Software Risk Management
- SRM
- Empirical

2.4.2 Search String

Synonyms of major terms and search string are:

RQ1

((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management plan OR software risk management

case study OR software risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment)

RQ2

((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management plan OR software risk management case study OR software risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment) **AND** (Empirical OR Industrial OR Experiment OR Case Study OR Survey))

Customized search string for each data base is given in chapter 3 execution.

The search strategy has the following decisions that were adopted according to SLR guidelines [22].

Items: Journal articles, workshop and conference papers.

Apply search on: Abstract

Language: The papers written in English.

Publication Period: Since 1989 till April 2011.

This review considered journal articles, workshop papers and conference papers written in English language and published since 1989 till April 2011.

2.4.3 Search Sources:

To ensure the quality of this review, most authenticated databases have been selected the for search. Although, there are many other platforms available across the world at international and national level, however, based on the Computer Science research community, we have chosen only those platforms which are most widely used and acknowledged at international level. This is also a plus point of this study because the research work published on these forum is double-blind peer-reviewed, and so already verified for the authenticity, originality and importance of the contents of the papers. This selection of strong suite is also the base of authenticated quantitative knowledgebase, which we have produced based on the selected papers. The selected platforms are:

- IEEE Explore
- ACM Digital Library
- Science Direct
- Springerlink

2.5 Study Selection Criteria:

A study selection criterion was:

- The initial selection will be on the basis of the TITLE and ABSTRACT of the paper.
- All obtained data from the search process will be archived in database according to the journal from which it is retrieved.
- From data base the duplicates will be removed after initial scan of results.
- Inclusion and exclusion criteria will be applied on the results to sort out the accepted papers.
- On accepted papers detail inclusion criteria which is a Quality Instrument for studies. will be applied to see whether they qualify to be included or not.
- Full papers of all studies that are not clearly ineligible will then be obtained.
- The excluded papers and reasons for exclusion will be recorded in a file, and the included papers and study type will be recorded in another file.

2.5.1 Study Inclusion/Exclusion Criteria

The criteria were intended to identify those studies that provide direct evidence for the research questions. Following were the inclusion and exclusion criteria for our research questions:

a. ***Inclusion criteria:***

In order to answer the stated research questions, we have searched for research articles by reading the abstracts that contain information about software risk management (SRM) and some empirical work done as well. So the paper can be a case study, an experiment, survey, experience report etc. when it was confirmed after reading abstract that the article is relevant to our research, and then we studied the whole paper. The objective of the selection process was to identify the articles relevant for the objectives of the systematic review. The search strings, were quite broad and hence it was expected that not all studies identified would make it to the final phase in the selection process. Only those studies were included that were empirical based i.e. experiment, case study, survey or industrial experience report and where the main focus was Software Risk Management.

Exclusion criteria:

- Those studies will be excluded that were based on personal expert opinion.
- Literature surveys and books were excluded.
- Only one inclusion for studies with the same results reported multiple times.
- Multiple studies could be reported in one paper; if any of them did not stand to our inclusion criteria then only that study was excluded.

2.6 Search Process Documentation

Search strings were applied to the selected databases and obtained studies were saved in different folders according to different databases. After apply inclusion/ exclusion criteria selected studies were copied into another folder. The categorization was implemented by making folders and saving files in these folders.

2.7 Quality Instrument for Quality Assessment

After initial selection of studies, a more detail criteria is required to judge the quality of study to see whether it is worth considering as evidence to answer our research question or not. Quality Instrument was designed for assigning numerical values for factors in the checklist to be evaluated for each study. Main focus was on the study design. The research questions and our inclusion/exclusion criteria suggest us that we are going to have evidence in form of empirical studies like case studies, industrial experience reports etc. So firstly we created a check list for assessing the quality of study and assigning numerical values to the questions so we can rank the obtained papers. If any paper was considered to be very poor in quality it was excluded at this stage and was recorded in the file of excluded papers with reasons. One paper can report multiple studies, in that case those studies will be evaluated individually for their criteria to be included or excluded.

Table 1: Quality Assessment Checklist

Quality Assessment Checklist	
Generic	
Does the author clearly state study objectives?	Yes/No
Is the study context adequately described?	Yes/Partial/No
Is a clear Chain of evidence established from observations to conclusions?	Yes/Partial/No
Do the researchers explain future implications?	Yes/No
Survey	
Was the denominator (i.e. the population size) reported?	Yes/No
Did the author justified sample size?	Yes/No
Is the sample representative of the population to which the results will generalize?	Yes/Partial/No
Are the statistical methods justified by the author?	Yes/Partial/No
Experiment	
Were treatments randomly allocated?	Yes/No
Are the variables used in the study adequately measured (i.e. are the	Yes/No

Quality Assessment Checklist	
variables likely to be valid and reliable)?	
Case Study/Action Research	
Does the case study describe multiple cases?	Yes/No
Is the case study based on theory and linked to existing literature?	Yes/No
Experience Report	
Is the focus of study reported?	Yes/No
Does the author report personal observation?	Yes/No
Is there a link between data, interpretation and conclusion?	Yes/No/Partial
Does the study report multiple experiences?	Yes/No

The answers to each of these questions could be yes, no and partially. The quality score used for 'yes' is 2, No is 0 and for partially is 1. The total sum of the scores will be used for the quality assessment of studies. Quality checklist adopted from [22][23][24][25][26][27] and modified for this SLR. The checklist items of case study and experience report sections were not adopted from any Quality instruments. These checklist items were designed based upon experience report and case study reporting guidelines [26][27].

2.8 Data Extraction

Data Extraction is performed after quality assessment. Data is extracted according to data extraction form. Data extraction form is designed by following [28] and we modified it for this SLR study. All the extracted data were relevant to the objective of the study and useful to answer the research questions. The review is undertaken by a single primary researcher, who is responsible for the data extraction. A secondary reviewer is approached for guidance in case of an issue regarding the data extraction. The inter-rated reliability test is performed after the data extraction process accomplished by the primary reviewer. The secondary reviewer selects few publications randomly from the list of publications already chosen by the primary reviewer. The secondary reviewer extracts the data independently from the randomly selected publication. The results of secondary reviewer are then compared with the results produced by the primary reviewer.

Primary Reviewer: Saima Irum.

Secondary Reviewer: Research supervisor (Dr. Naveed Ikram)

The Data extraction form contains following data fields about the SLR study:

i. Study ID
ii. Title
iii. Year
iv. Author
v. Search Database
vi. Quality Checklist ranking
vii. Empirical Background
viii. Software risk management Areas and research output.
ix. Application domain
x. Focus of the study
xi. Qualitative Evaluation

In data extraction process data was stored in a MS-Excel sheet. In addition, a qualitative evaluation of the papers was also performed. The qualitative evaluation was useful to cross-check the extracted data.

Table 2: Fields of Data Extraction Form

DATA EXTRACTION FORM	
Study ID	
Title	
Year	
Database	
Quality Assessment Ranking	
Relevance	
Is this article relevant to SE field?	○ Highly relevant ○ Relevant ○ Irrelevant

DATA EXTRACTION FORM	
Is this article relevant to software risk management field?	○ Highly relevant ○ Relevant ○ Irrelevant
Is this an empirical study?	○ Yes ○ No
Does this article repeat already reviewed article(s)?	○ Yes ○ No
Empirical Background	
Main Method	☐ Survey ☐ Case Study ☐ Interviews ☐ Controlled Experiment ☐ Survey
Sub-Method	☐ Survey ☐ Case Study ☐ Interviews ☐ Archive Analysis ☐ Controlled Experiment ☐ Other...
Subjects of investigation	☐ Students ☑ Industry/Real world
Empirical focus	☐ Empirically based ☐ Empirically evaluated
Software Risk management Background	
SRM Processes	
Types of Risks	
SRM Practices	
SRM methodologies	
SRM frameworks	
SRM Models	
SRM techniques	
Software tool support	
Risk Identification	
Risk Assessment	
Risk Prioritization Tech.	
Risk Control	
Risk Control Technique	
Study	
Focus of the Study	○ SRM in General ○ Single Practice(s) ◉ Development Phase(s) ○ Others...
Application Domain	☐ Telecom ☐ Automative ☐ Web ☐ Finance ☐ Automation ☐ Unclear ☐ Other...

DATA EXTRACTION FORM	
Definitions in the introduction-like sections?	☐ No ☐ Software Risk Management ☐ Other related definitions
Qualitative Evaluation	
Claims	Narrative
Personal reflection	Narrative
Recommendations	Narrative

2.9 Piloting:

Initially piloting was done before conducting this SLR to check whether we have enough number of empirical studies in the field of software risk management. Many empirical studies were found from different databases. After this initial piloting to increase the consistency in the data extraction phase a designed data extraction form was evaluated in a pilot extraction of 15 papers randomly chosen from the primary studies database. Based on the results of the pilot review the data extraction form was modified.

2.10 Data Analysis and Synthesis

Results gained from the extracted data are analyzed and synthesized. The research areas in the field of software risk management were identified along with gaps and future directions. Relationships among various categories of data were pointed out with multiple perspectives. Quantitative and qualitative analysis of the data was performed to evaluate the strength of the literature. Results were presented in the form of systematic maps like Bar graphs, Bubble charts etc. Outcomes of this research are information like what's the most widely used empirical method applied by the researchers and practitioners in software risk management? Who are most involved in software risk management research? What are most widely used tools, methodologies, processes, techniques etc used for software risk management?

2.11 Validation of the Review Process

This review process is evaluated by an internal reviewer and one external reviewer reviewed the SLR protocol before execution of the protocol. The protocol was initially evaluated by the

research supervisor Dr. Naveed Ikram. Then it was reviewed by external reviewer and protocol was updated according to the comments of the external reviewer Dr. Saad Zafar.

Chapter 3: Protocol Execution

3. Protocol Execution

Systematic Literature Review is performed to find out the state-of-the-art in empirical studies of software risk management and the strength of empirical evidence reflected in empirical software risk management literature. A plan was defined to conduct the SLR in the last chapter. This chapter describes the steps performed during realization of the plan execution.

3.1 Search String Application to Databases

General search string is provided in section 2.4. An initial scoping study helped in identifying search terms and search sources. Google scholar was included in search sources but later removed as it gave different search results of the same search string at different times. Some databases did not allow the complete search string. Different search sources have different search string format so search string was modified and then applied on such search sources.

Customized search string applied on each database and citations are downloaded in a master library in endnote. Customized search string for each database and results retrieved are shown below:

3.1.1 IEEE Search Query

IEEE accepted the complete search string. The executed string is shown in table 3:

Table 3: Search string for IEEE

IEEE
String Query
((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management case study OR software

IEEE
risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment) **AND** (Empirical OR Industrial OR Experiment OR Case Study OR Survey))

3.1.2 Science Direct Search Query

Science Direct accepted the complete search string. The search string is shown table 4:

Table 4: Search String for Science Direct

Science Direct
String Query
((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management case study OR software risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment) AND (Empirical OR Industrial OR Experiment OR Case Study OR Survey))[All Sources(Computer Science)]

3.1.3 SpringerLink Search Query

Due to Springer Link's ability to search limited number of terms provided in query, the search string is broken down into thirty one sub strings. Each string is executed separately. Number of results retrieved is shown in table 3.3 after executing all sub search strings. SpringerLink provide many filter options to get the accurate results. Two filters, "computer science" and "software engineering" were applied on these sub strings. Before applying filters obtained number of studies were very large which brought many irrelevant studies.

Table 5: Search String for Springer Link

Springer Link
String Query
(Software risk management) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Metrics) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Plans) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Tools) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Barriers) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Risk Management Software) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Project Risk Management) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Techniques) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software System Risk Management) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Development Risk Management) and (Empirical OR Industrial OR

Springer Link
String Query
Experiment OR Case Study OR Survey)
(CMM based Software Risk Management)and(Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Project Risk Management Framework) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Risk Management Enterprise Software) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(Software Risk Management Process) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk management case study) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk management practices) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software engineering risk management) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk assessment) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk identification) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk checklist) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk analysis) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk prioritization) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk exposure) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk reduction) and (Empirical OR Industrial OR Experiment OR Case

Springer Link
String Query
Study OR Survey)
(software risk management planning) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk control) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk plan integration) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk avoidance) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk element planning) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk resolution) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)
(software risk reassessment) and (Empirical OR Industrial OR Experiment OR Case Study OR Survey)

3.1.4 ACM Search Query

For ACM Search string is shown table 6:

Table 6 : Search string for ACM

ACM
String Query
(("Abstract":"Software Risk Management" OR "Abstract":"Software Risk Management Metrics" OR "Abstract":"Software Risk Management Plans" OR "Abstract":"Software Risk Management Tools" OR "Abstract":"Software Risk Management Barriers" OR "Abstract":"Risk Management Software" OR "Abstract":"Software Project Risk Management" OR "Abstract":"Software Risk Management Techniques" OR "Abstract":"Software System

ACM
String Query
Risk Management" OR "Abstract":"Software Development Risk Management" OR "Abstract":"CMM based Software Risk Management" OR "Abstract":"Software Project Risk Management Framework" OR "Abstract":"Risk Management Enterprise Software" OR "Abstract":"Software Risk Management Process" OR "Abstract":"software risk management case study" OR "Abstract":"software risk management practices" OR "Abstract":"software engineering risk management" OR "Abstract":"software risk assessment" OR "Abstract":"software risk identification" OR "Abstract":"software risk checklist" OR "Abstract":"software risk analysis" OR "Abstract":"software risk prioritization" OR "Abstract":"software risk exposure" OR "Abstract":"software risk reduction" OR "Abstract":"software risk management planning" OR "Abstract":"software risk control" OR "Abstract":"software risk plan integration" OR "Abstract":"software risk avoidance" OR "Abstract":"software risk element planning" OR "Abstract":"software risk resolution" OR "Abstract":"software risk reassessment") AND ("Abstract":"Empirical" OR "Abstract":"Industrial" OR "Abstract":"Experiment" OR "Abstract":"Case Study" OR "Abstract":"Survey")

3.1.5 Tools used for automating the search process

To speed up the process of documenting search results End notes tool is used which helped in automating the process. Search results are retrieved in End notes; it helped in documenting the search results for primary studies.

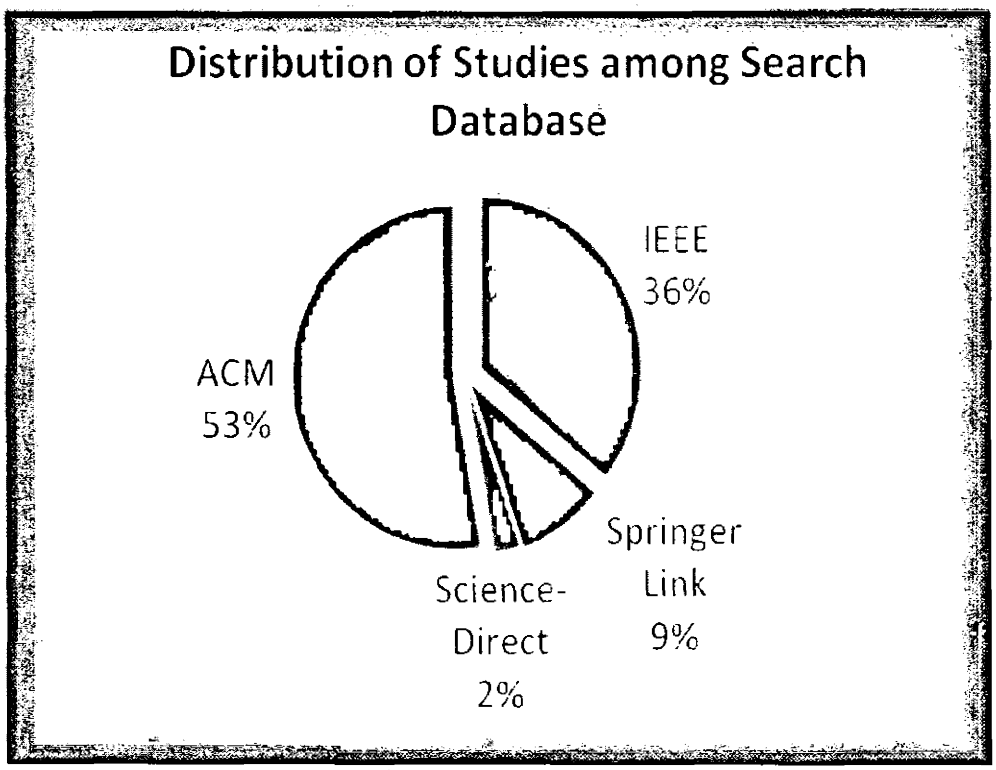
3.1.6 Identified Studies

The total number of primary studies we obtained was 622 out of which 53% of our studies were from ACM, 36% from IEEE, 9% from SpringerLink and 2% was searched from ScienceDirect. The frequency of primary studies found from ACM is greater than other, one of the reason is that ACM publisher's are mostly focused by Computer Scientists including software engineering under the umbrella of CS, whereas, IEEE publish mostly engineering related studies.

Table 7 : Total No. of Studies Obtained after string search

No: of Studies Obtained after String Search	
Resource	No. of Studies
ACM	328
IEEE	223
SpringerLink	55
ScienceDirect	16

Figure 1: Distribution of Studies among Search Database.



3.2 Studies Inclusion/Exclusion Process

The objective of the selection process is to identify the articles relevant for the objectives of the systematic review. The search strings, are quite broad and hence all studies identified are not relevant and do not full fill Inclusion/Exclusion Criteria. Before applying inclusion/Exclusion criteria 70 duplicate studies were discarded from the library. Two level screening is followed for the final selection of studies.

1. Title and abstract screening

2. Full text screening

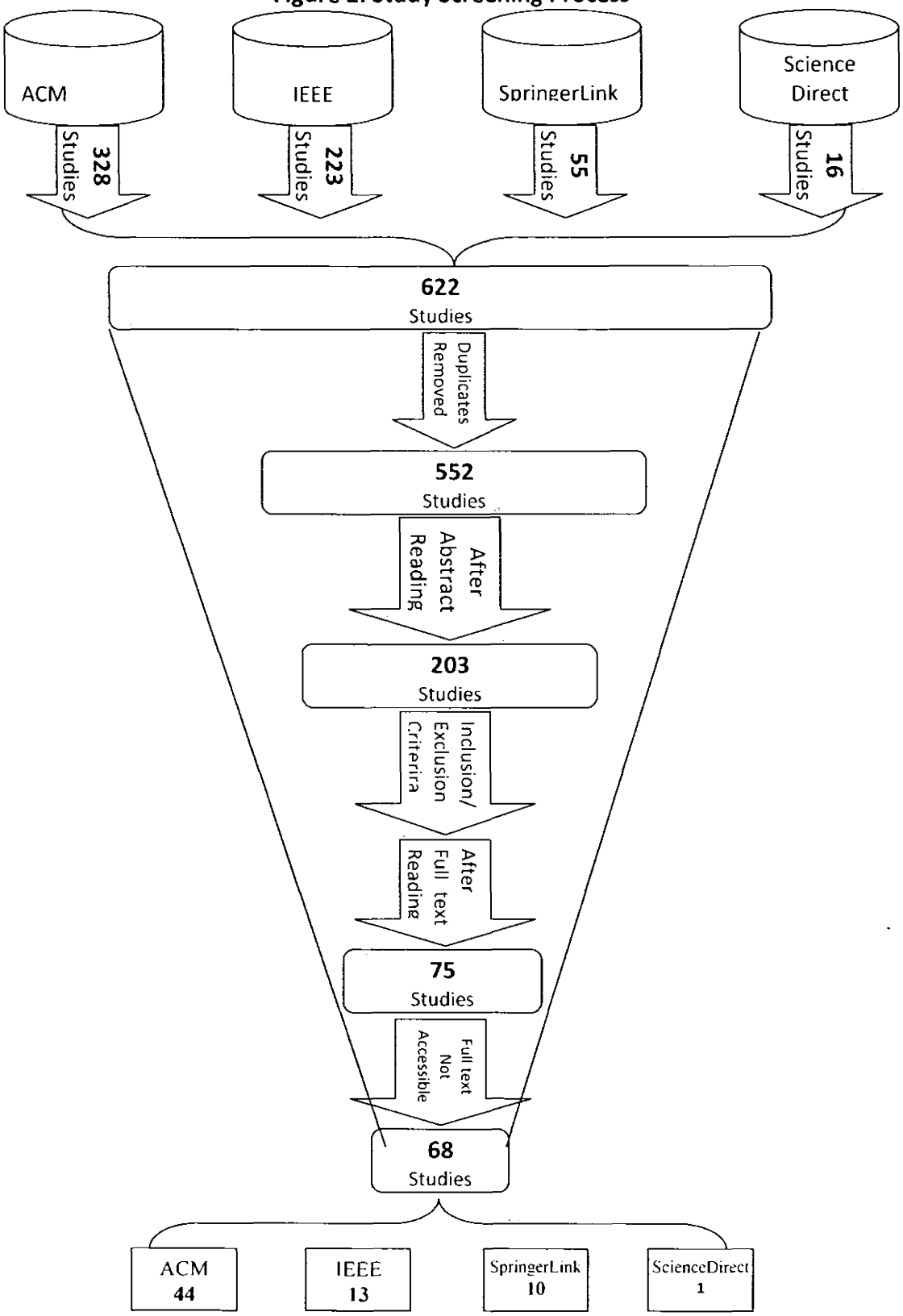
3.2.1 Title and abstract screening

Level 1 searching is performed on title, keywords and abstract. The purpose is to exclude completely irrelevant articles. Abstract level screening provides an easy way to exclude unrelated articles. Applying search string on different databases retrieved 622 studies. Inclusion/exclusion criteria defined in protocol was applied on these studies. Studies that failed to fulfill the criteria i.e., that were unrelated studies were excluded. No. of articles found after level 1 screening were 204.

3.2.2 Full text screening

Inclusion/Exclusion criteria defined in protocol is applied for full text screening. First author performed screening of the papers. Papers not meeting the inclusion criteria and fulfilling the exclusion criteria were rejected and the reason for rejection was recorded. In case of uncertainty about inclusion/exclusion of paper supervisor was consulted. No. of selected articles were 68.

Figure 2: Study Screening Process



Duplicate studies were removed automatically by using Endnotes Software. Other filters were performed manually.

3.3 Study Quality Assessment:

Study Quality assessment was performed by following the criteria defined in protocol. Quality score is presented in tables for each type of studies. The cumulative scores of case study and experiment were 12; scores of experience report and survey were 16. As the number of questions of each study is different so the total number of quality score is not same. Therefore, percentages were used for the overall assessment of quality in order to maintain uniformity. Complete list of studies along with their quality scores is shown in Table 8. In the table, study settings (Case study, Survey, Experiment, Experience Report), percentages of total quality score (QC) obtained is displayed along with study IDs (S-ID) assigned for unique identification of the studies. Percentage column shows the calculated percentage of quality. Generic questions include Q1 to Q4; these questions are related to context of the study, study objectives, clear links among data and interpretations, and future implications. Survey questions includes S1 to S4, the questions are related to survey methodology. Quality experimentation includes E1 to E2. Case study methodology questions are C1 and C2. ER1 to ER4 are the questions about experience reports.

Studies that scored more than 60% were considered as *quality studies*. The studies that scored below 60% were 14 out of 68 therefore the rest of the 54 studies were the *quality studies*. The studies having score between the ranges of 60% to 75% were considered as *above average*. The number of studies in aforementioned range was 28 in number. The Studies that scored more than or equal to 76%, were considered as *good quality studies*. A total of 26 studies were found as good quality studies, whereas, a total of 10 studies attained 100% quality scoring.

The quality scores are depicted in Table 8, as follows:

Table 8: Quality Scores

Quality Scores																		
S#	S-ID	QC	Generic Questions				Case Study		Survey				Experiment		Experience R			
			Q1	Q2	Q3	Q4	C1	C2	S1	S2	S3	S4	E1	E2	ER1	ER2	ER3	ER4
1	P1	83%	2	2	2	0	2	2										
2	P3	88%	2	1	2	2			2	2	2	1						
3	P6	100%	2	2	2	2	2	2										
4	P7	94%	2	2	2	2			2	2	1	2						
5	P9	58%	2	1	2	0	0	2										

Quality Scores																		
S#	S-ID	QC	Generic Questions				Case Study		Survey				Experiment		Experience R			
			Q1	Q2	Q3	Q4	C1	C2	S1	S2	S3	S4	E1	E2	ER1	ER2	ER3	ER4
6	P10	67%	2	1	1	2	0	2										
7	P12	67%	2	1	1	2							2	0				
8	P14	100%	2	2	2	2	2	2										
9	P15	94%	2	2	2	2			2	2	2	1						
10	P16	94%	2	2	2	2			2	2	1	2						
11	P18	100%	2	2	2	2			2	2	2	2						
12	P20	75%	2	0	1	2							2	2				
13	P23	67%	2	1	1	2	0	2										
14	P24	88%	2	2	2	2			2	2	2	0						
15	P27	100%	2	2	2	2	2	2										
16	P29	81%	2	2	2	2			2	2	1	0						
17	P30	100%	2	2	2	2			2	2	2	2						
18	P33	100%	2	2	2	2	2	2										
19	P38	63%	2	2	1	0			2	2	1	0						
20	P40	58%	2	0	1	2	0	2										
21	P43	83%	2	2	2	2	0	2										
22	P44	100%	2	2	2	2			2	2	2	2						
23	P47	75%	2	2	2	2			2	2	0	0						
24	P48	81%	2	2	2	0			2	2	1	2						
25	P49	67%	2	1	1	2	0	2										
26	P50	100%	2	2	2	2			2	2	2	2						
27	P51	100%	2	2	2	2							2	2				
28	P52	83%	2	2	2	0							2	2				
29	P53	69%	2	2	1	0			2	2	2	0						
30	P59	63%	2	2	2	2			2	0	0	0						
31	P63	69%	2	2	1	2			2	2	0	0						
32	P64	88%	2	2	2	2			2	2	2	0						
33	P66	88%	2	2	2	2			2	2	1	1						
34	P69	75%	2	1	2	2	0	2										
35	P71	75%	2	2	1	2			2	2	1	0						
36	P76	56%	2	2	0	2			2	0	1	0						
37	P77	75%	2	2	2	0			2	2	2	0						

Quality Scores																		
S#	S-ID	QC	Generic Questions				Case Study		Survey				Experiment		Experience R			
			Q1	Q2	Q3	Q4	C1	C2	S1	S2	S3	S4	E1	E2	ER1	ER2	ER3	ER4
38	P78	56%	2	2	1	0			2	0	2	0						
39	P79	44%	2	2	1	2			0	0	0	0						
40	P81	69%	2	2	2	2									2	0	1	0
41	P82	50%	2	0	0	2							0	2				
42	P83	92%	2	1	2	2	2	2										
43	P84	75%	2	1	2	2	0	2										
44	P85	67%	2	1	1	2	0	2										
45	P90	31%	2	0	1	0									2	0	0	0
46	P92	75%	2	1	2	2	0	2										
47	P99	38%	2	1	1	0			2	0	0	0						
48	P104	67%	2	1	1	2							0	2				
49	P105	63%	2	0	1	2									2	0	1	2
50	P114	69%	2	1	2	0									2	0	2	2
51	P139	92%	2	2	1	2							2	2				
52	P149	83%	2	1	1	2	2	2										
53	P169	67%	2	1	1	2	0	2										
54	P170	67%	2	1	1	2	0	2										
55	P171	67%	2	1	1	2	0	2										
56	P174	42%	2	2	1	0							0	0				
57	P175	83%	2	1	1	2	2	2										
58	P179	58%	2	0	1	2	0	2										
59	P183	58%	2	0	1	2	0	2										
60	P184	75%	2	2	2	2			2	2	0	0						
61	P185	67%	2	1	1	2	0	2										
62	P188	42%	2	0	1	0	0	2										
63	P192	75%	2	1	2	2	0	2										
64	P194	75%	2	1	2	2	0	2										
65	P198	50%	2	1	1	2			2	0	0	0						
66	P205	50%	2	1	1	0	0	2										
67	P206	75%	2	0	1	2	2	2										
68	P207	100%	2	2	2	2	2	2										

Chapter 4: Results & Analysis

4. Results & Analysis

This chapter describes the results and analysis of this SLR. This section presents the results of RQ1 & RQ2. Results are the raw data extracted from the papers and presented without any processing.

4.1 What is the state-of-the-art in empirical studies of software risk management?

We aimed to understand the existing research directions within software risk management and specifically empirical research on the topic. To answer RQ.1, an analysis based on different perspectives of the empirical literature in the software risk management field is presented. Extracted data was synthesized quantitatively in terms of frequency of occurrence to depict the mature and underdeveloped areas of software risk management in terms of frequency of the studies. Obtained statistical results from the study are presented here.

4.1.1 Yearly distribution of studies

The concept of SRM was formally introduced by Boehm in 1989 [10], therefore, we considered the time period from 1989 to 2011 for this study, in order to cater all empirical studies from the beginning of the software risk management area. As described in figure 1, few empirical studies of software risk management were accomplished during the era of 90's. There were 4 empirical studies before 1995, number of empirical studies gradually increases and there were 9 studies found from the year 1996 to 2000. Number of empirical studies has been increased dramatically after 2000. A total of 18 empirical studies of software risk management have been found from year 2001 to 2005 whereas, 37 studies have been found from year 2006 to 2011. Trend of empirical work in the area of software risk management have been increased almost double in every six-year range, which shows the importance of the field.

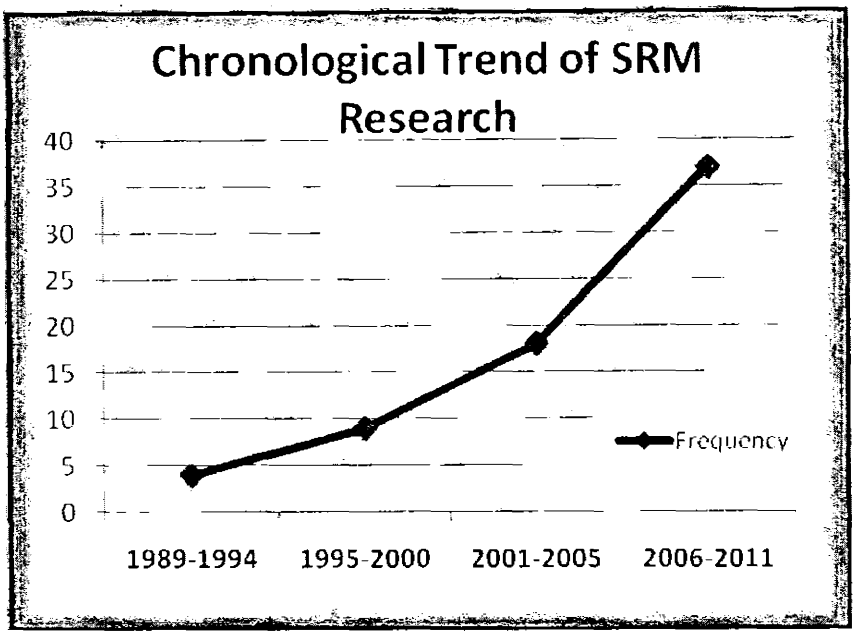


Figure 3: Chronology of SRM Studies

4.1.2 Country-wise distribution of studies

Empirical research of software risk management has been applied globally. Few countries like USA invested more in this study. Figure 2 shows the distribution of studies in different countries. 17 studies were carried out collaboratively in which mostly the groups were from UK, USA, Finland, Canada, Norway and Germany were involved. We found 15 empirical studies which were carried out in USA, one of the main reasoning behind these much number of studies in USA can be the availability of resources for SRM as well as willingness of all the stakeholders to go through the phases of SRM, as per our opinion. Further, some other countries like china, Belgium, Italy, Australia, Malaysia also contributed prominently in the study of software risk management.

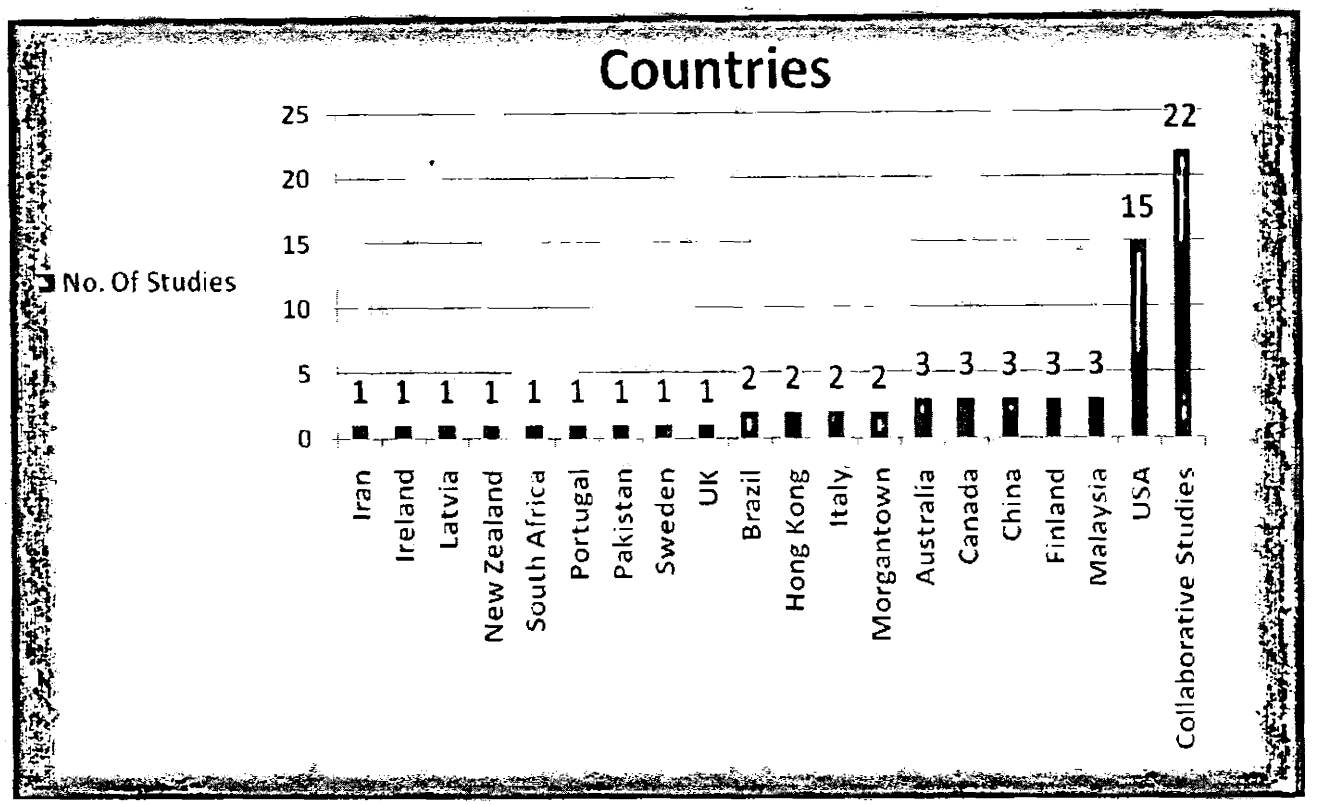


Figure 4: Country-wise distribution of studies

4.1.3 Software Risk Management Areas

Software risk management areas were investigated to understand what is known and what is not known about software risk management. A qualitative analysis of each study was done during the data retrieval process. There are no pre-defined areas of software risk management that's why we explored the topics addressed by the studies. We found a large number of studies in the area of software risk assessment such as risk identification, risk analysis and risk prioritization. Many studies did not describe all these sub areas of risk assessment, most of the studies are focusing on specific technique, tool, method for particular area of software risk management.

Table 9: Software Risk Management Areas Extracted From the Studies

(A study discusses multiple areas of SRM)

Software Risk Management Areas	Papers	Number of Studies
Risk Assessment	P1, P3, P6, P10, P12, P15, P16, P23,P24,P33,P40,P43,P47,P50, P51, P52, P, P59, P71, P78, P79, P81, P82, P83, P84, P85, P90, P92, P99, P104, P105, P114, P149, P170, P171, P175, P179, P183, P185, P188, P192, P198, P205, P206	43
Risk Identification Techniques	P1, P6, P7, P12, P14, P15, P43, P49, P50, P59, P69, P71, P76, P78, P79, P81, P83, P84, P85, P92, P104, P139, P169, P170, P171, P174, P175, P179, P183, P205, P206	31
Risk Control	P1, P6, P7, P9, P14, P18, P49, P59, P63, P64, P69, P71, P78, P79, P83, P85, P90, P169, P170, P174, P179, P183, P184, P185, P188, P194, P206	27
Risk control techniques	P1, P7, P9, P14, P18, P49, P59, P63, P64, P69, P71, P78, P79, P83, P85, P169, P170, P174, P179, P183, P184, P185, P188, P194, P206	25
Risk analysis techniques	P1, P6, P10, P12, P14, P20, P23, P49, P59, P71, P78, P79, P83, P84, P85, P92, P104, P139, P170, P171, P175, P206	22
Risk Prioritization techniques	P1, P6,P7, P12, P14, P16, P18, P24,P29, P48, P49, P59, P64, P69, P79, P83, P84, P85, P149, P169, P170	21

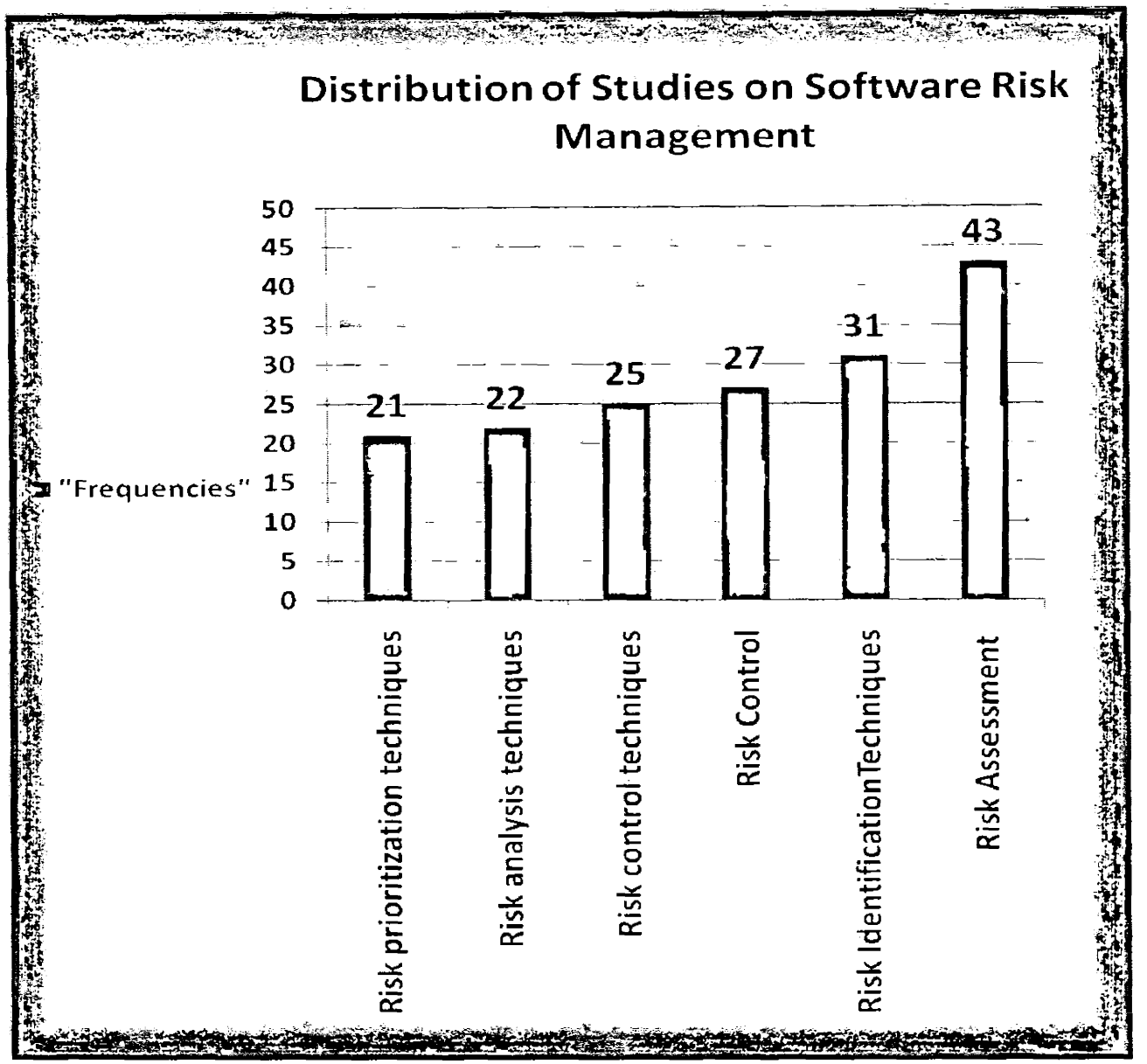


Figure 5: Distribution of Studies on SRM

(A study discusses multiple areas of SRM)

Fields of areas of SRM were set in Data Extraction form. Then filters were applied on each field to know the exact number of studies in specific areas of software risk management.

Software risk assessment is the most studied area of software risk management (43 papers) followed by software risk identification (31 papers) and Risk Control (27 papers), respectively. Risk prioritization Techniques is the least studied area of software risk management covered by 21 studies, one reason can be the simplicity of widely used prioritization scheme which is

ordering. Figure 5 shows the distribution of studies according to the areas of software risk management.

4.1.4 Domain of the studies

To identify distribution of studies among different application domains pie chart is plotted. Figure 6 shows that a large number of generic studies (38 papers, 54%) were found. Generic option is used where software risks management in general applied on software projects and where software risk management is applied in different development phases of software engineering like requirements gathering, architecture level etc. Information technology (10 papers, 14 %) and information system (6 papers, 9%) domain are the most explored fields for software risk management.

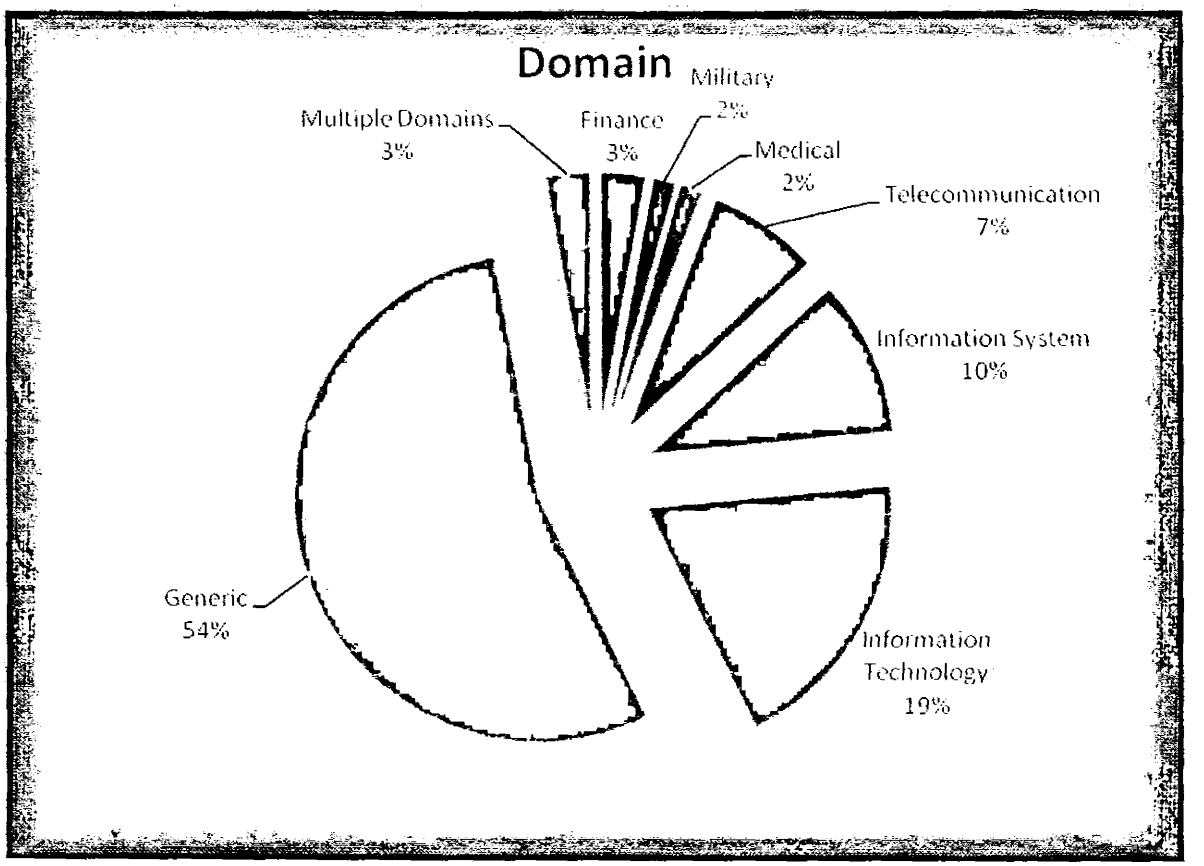


Figure 6 : Domain of the studies

4.1.5 Output of the studies

To find the state of the art of software risk management field, following results are obtained in terms of SLR of selected 68 studies. In software risk management area, risks are managed by means of SRM methodologies, models, frameworks, processes, tools, practices, techniques, approaches, metrics and instrument. Few studies only identified the risks and generated risk factors/items list. Figure 7 shows the distribution of studies and the frequency of research outputs.

Table 10 : Outputs Extracted from the Studies

Outputs	Papers	Number of Studies
SRM Methodologies	P6, P9, P10, P14, P23, P27, P43, P49, P84, P92, P139, P174, P175, P179, P188	15
SRM Model	P15, P30, P40, P47, P52, P71, P82, P83, P85, P90, P104, P174	12
SRM Framework	P7, P29, P33, P59, P69, P76, P79, P84, P183, P206	10
SRM Processes	P12, P43, P99, P105, P169, P170, P188, P206	8
Software Tool	P9, P12, P40, P50, P51, P105, P169, P205, P206	9
SRM Practices	P50, P78, P81, P175, P194	5
SRM Techniques	P20	1
SRM Approach	P1, P192	2
SRM Instrument	P3	1
Risk factors	P16,P18, P24, P38, P44, P48, P53, P63, P64, P66, P77, P184, P185, P198, P207	15
Metrics	P114, P149, P170, P171	4

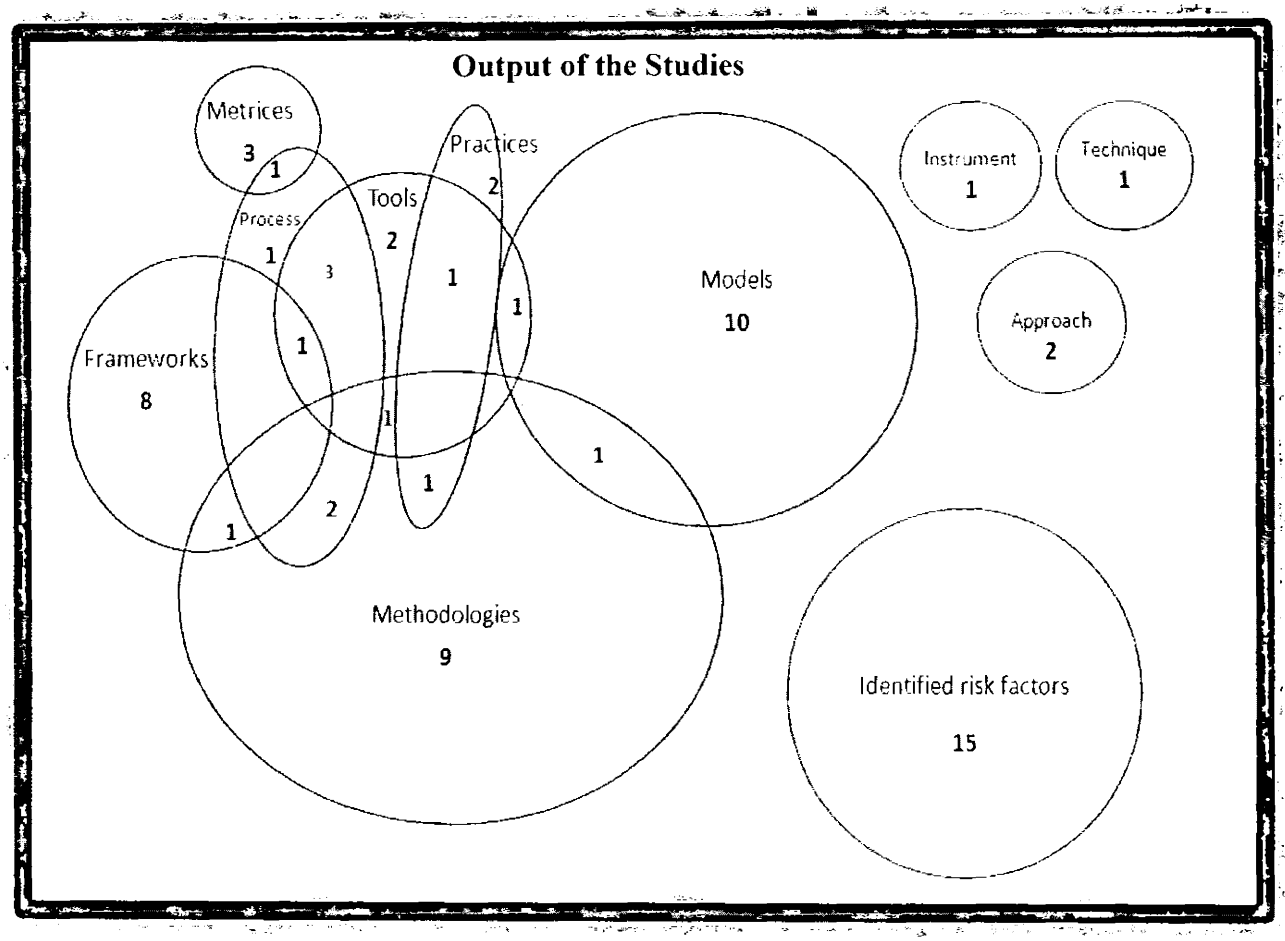


Figure 7: Output of the studies

SLR results show that the most frequent means used in the management of software risks are: methodologies, models, frameworks and risk factors list. We found 15 methodologies in which "riskit" method is the most popular method and widely used to manage risks in software development. There are 10 software risk management frameworks identified out of which four frameworks were described for the purpose of risk categorization. In 15 studies, risk factors were identified for the purposes of finding new risks in specific areas and produce a ranked ordered list of risks.

4.2 What is the strength of empirical evidence reflected in empirical software risk management literature?

In this systematic review nature of empirical evidence presented in software risk management studies was investigated and analyzed the type of studies conducted. Different empirical studies provide different strengths of evidence. Practitioners should take the strength of evidence into consideration when deciding on software risk management practices based on the existing literature. Thus it is important to understand the state of research methods, study participants, data collection methods and research approaches when studying software risk management.

4.2.1 Study participants

Data was collected from each selected study about study participants to that the study was conducted in industry or academia. There were some studies which were conducted by the mutual effort of industry and academia, which are referred as mixed studies. From the extracted data we found that most of the research is going on in academia as Figure 8 depicts that 79% research has been done by academia. Industry's contribution was 18% and mixed studies were only 3%.

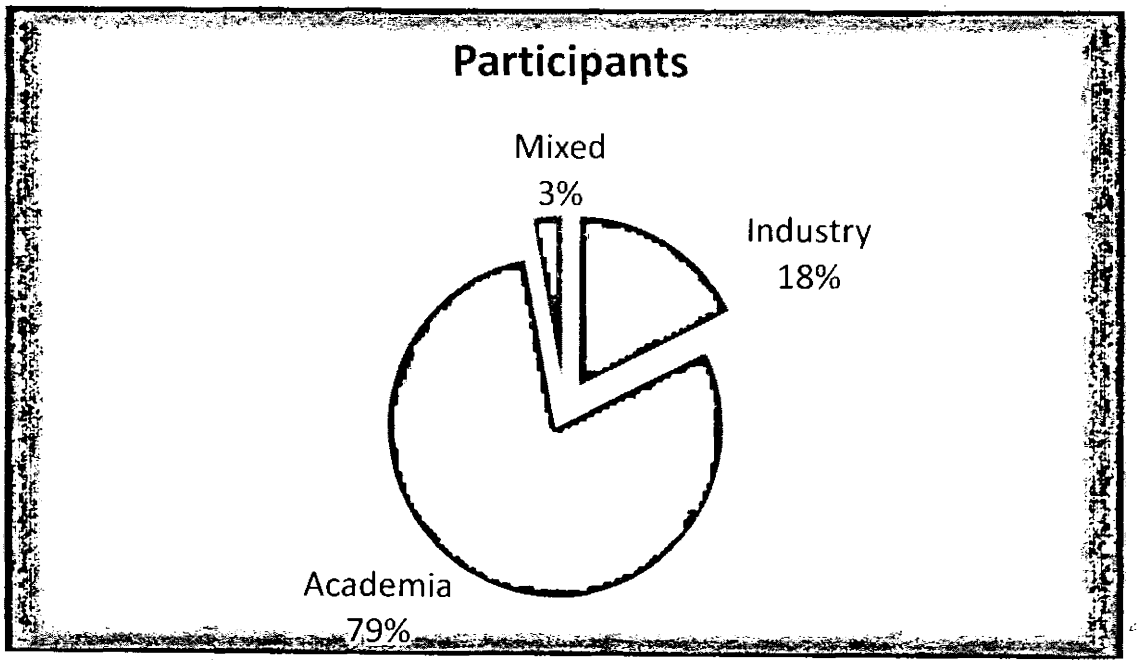


Figure 8 : Study Participants

4.2.2 Research Methods

We captured sources of empirical evidence to understand the viability of the offered findings. The analysis of the studies shows that the majority of research is exploratory case studies. We identified 28 case studies and in 9 of these the researchers mention that they have performed interviews as data collection method of the case study. In eight of these 28 studies the researchers mentioned that they performed a document analysis, and so forth. It has also been observed in this systematic review that survey is the second most common method.

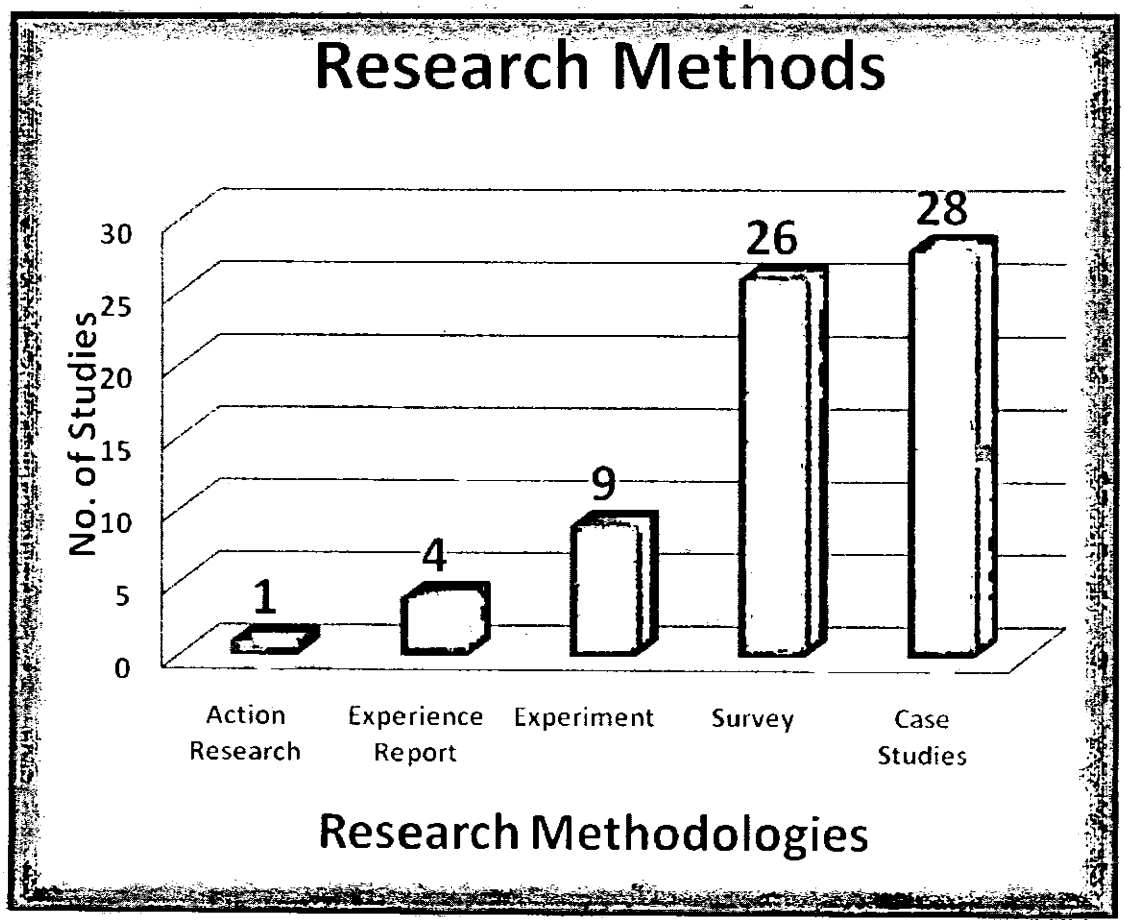


Figure 9 : Distribution of Research method on Studies

4.2.3 Data collection methods

Figure 10 shows that the questionnaire is the common research tool used by 22 papers which had different types such as web based questionnaire and self administered questionnaire, scaling questionnaire, and open ended questionnaires. Other most common method used in empirical studies of this survey is interviews used by 21 studies.

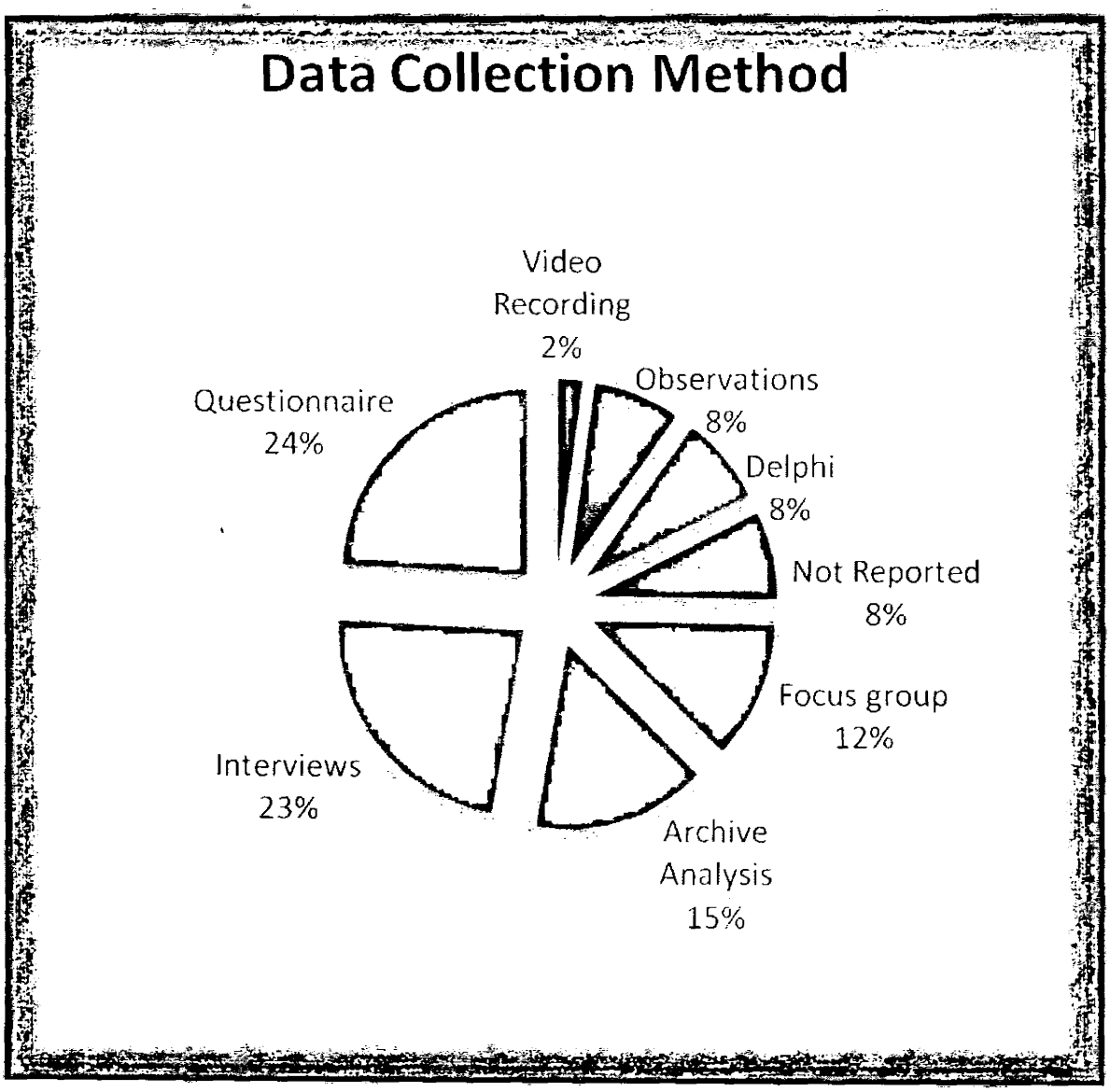


Figure 10 : Data Collection methods

4.2.4 Combined Data Collection Methods

Data collection methods were used as a individual single method in some studies while many researchers used two or more than two methods in combination. We analyzed these individual

and hybrid combinational methods to clearly understand the sources of data collection. When data is collected using multiple data collection methods and multiple sources then it is believed to be repeatable process and this is also linked with the strength of the evidence. We summarized the various data collection methods used in different study settings like data collection methods used in case study, survey and experiments.

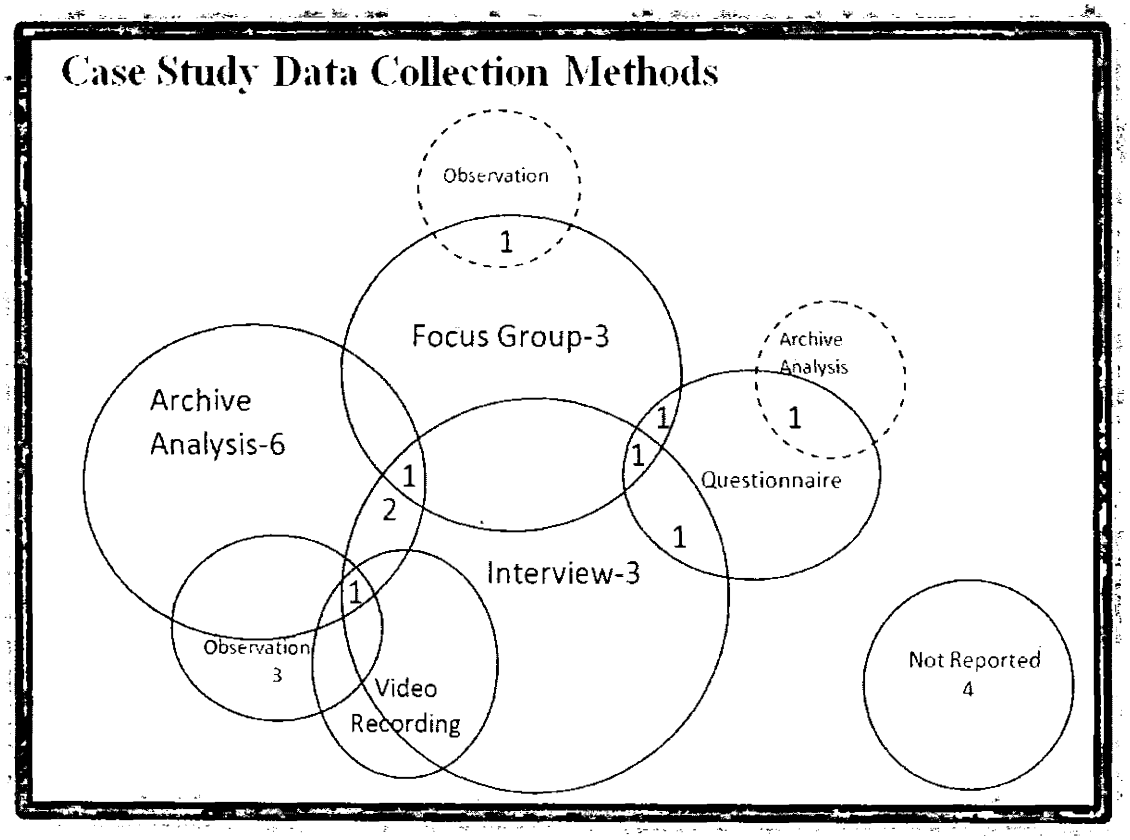


Figure 11: Data Collection methods used in case studies

Venn diagram has been used to depict the mixed methods used by the researchers. For case study data collection methods used by various researchers are depicted in Figure 11. Venn diagram in figure 11 shows that in 3 studies interviews were conducted as a standalone data collection method. In 2 studies interviews were conducted with Archive analysis and in one study interviews, archive analysis and focus groups were used. In one study interviews, video recording, archive analysis and observations were used as data collection methods. In 3 studies observations were used as a standalone data collection method. Focus group is used in 3 studies

as standalone method to collect data. In one study focus group and observations were used collectively for data collection. One study used questionnaire and interview both and one study used focus group and questionnaire as a data collection methods. Questionnaire is also used with archive analysis for data collection in one study. There are 4 studies that did not explicitly mentioned their data collection method that is why we used "not reported" as term to depict these studies.

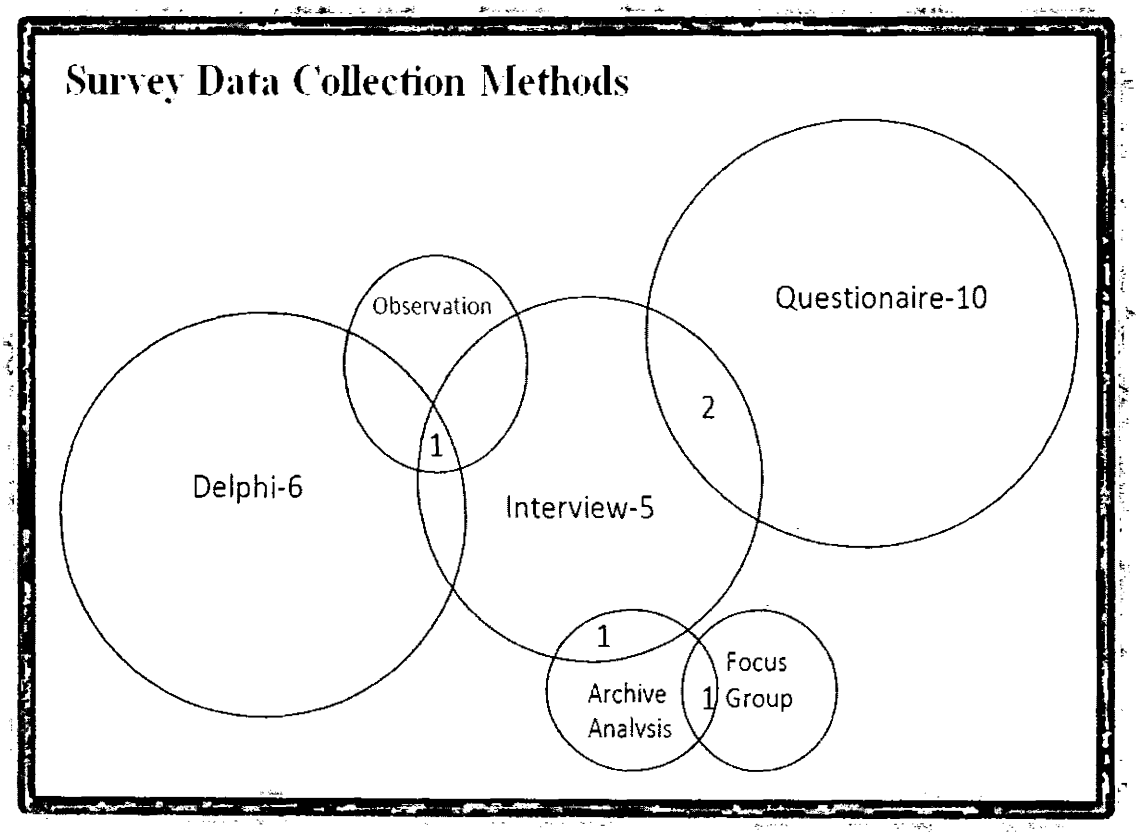


Figure 12: Data collection methods used in experiments

In case of survey the data collection methods used by various researchers are shown in Venn diagram of Figure 12. It is clear from the figure that 10 studies conducted questionnaire as the only data collection method. In 2 studies questionnaire was used with interviews. In 5 studies interviews are used as a standalone data collection method. In 1 study interviews, observation and Delphi were used as data collection methods. One study used interviews and archive analysis

collectively for data collection and in one study focus group and archive analysis were used as data collection methods.

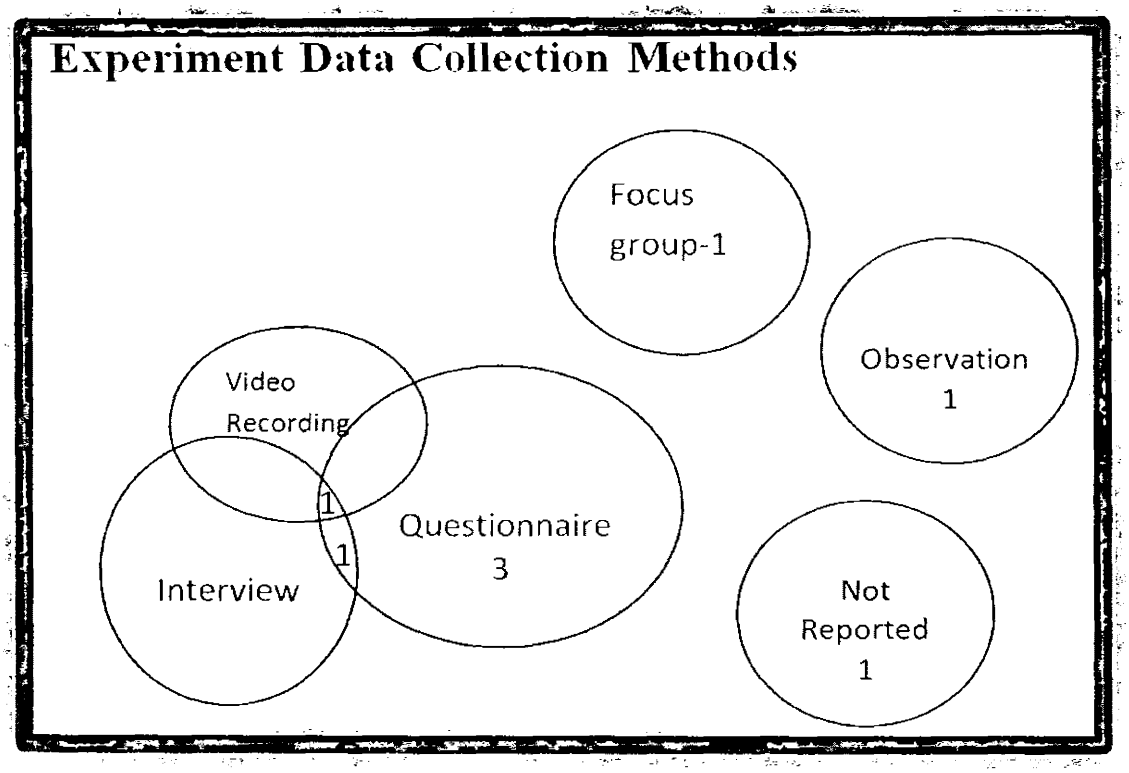


Figure 13: Data Collection methods used in Experiments

In experiments 3 studies used Questionnaire as a single data collection method and in one study questionnaire and interview were used collectively. Questionnaire was used with video recording in one study as data collection method. One study used focus group and one study used observation as standalone data collection method. In one study data collection method was not reported.

For experience report two studies used focus group as a single data collection method and in one study archive analysis was used as a standalone data collection method where as there were two studies that did not reported data collection method. We obtained one study as action research and in that study data collection method was also not mentioned. That is why Venn diagrams has not been drawn for experience report and for action research.

4.2.5 Research Approaches

This research also evaluated the ratio among empirically-based versus empirically-evaluated research in the SRM field. Empirically-based is referred to a study basing its conclusions on empirical data, but not performing any actual empirical evaluation and evaluating a practice, a method, a framework or a tool is referred to as empirically-evaluated research. Majority of the studies are based on empirical data. Only 21 out of 68 studies were classified as empirically evaluated research, i.e. where the researchers actually evaluate a method, technique or tool for software risk management. It is notable that most of studies performing empirical based studies are from academia participants.

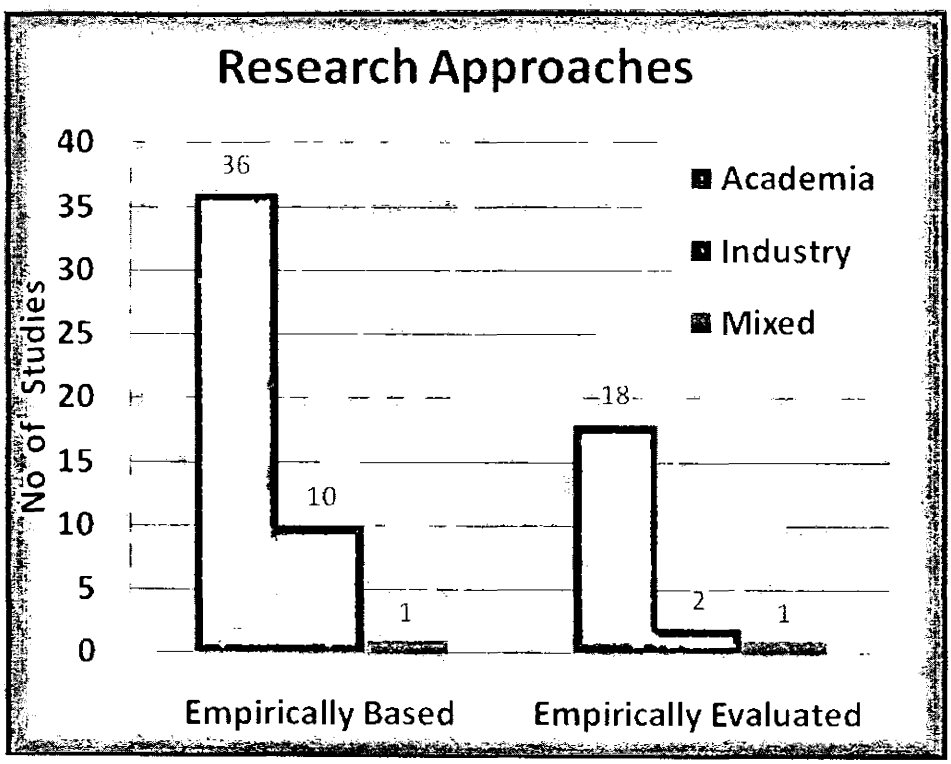


Figure 14 : Research Approaches

4.3 Software Risk Management vs. Study Settings

Goal of second research question (RQ2) was to investigate the strength of empirical evidence in software risk management. A bubble chart is plotted to show the distribution of research methods in software risk management. We found that a case study is most common research method used in all areas of risk management followed by surveys and experiments respectively.

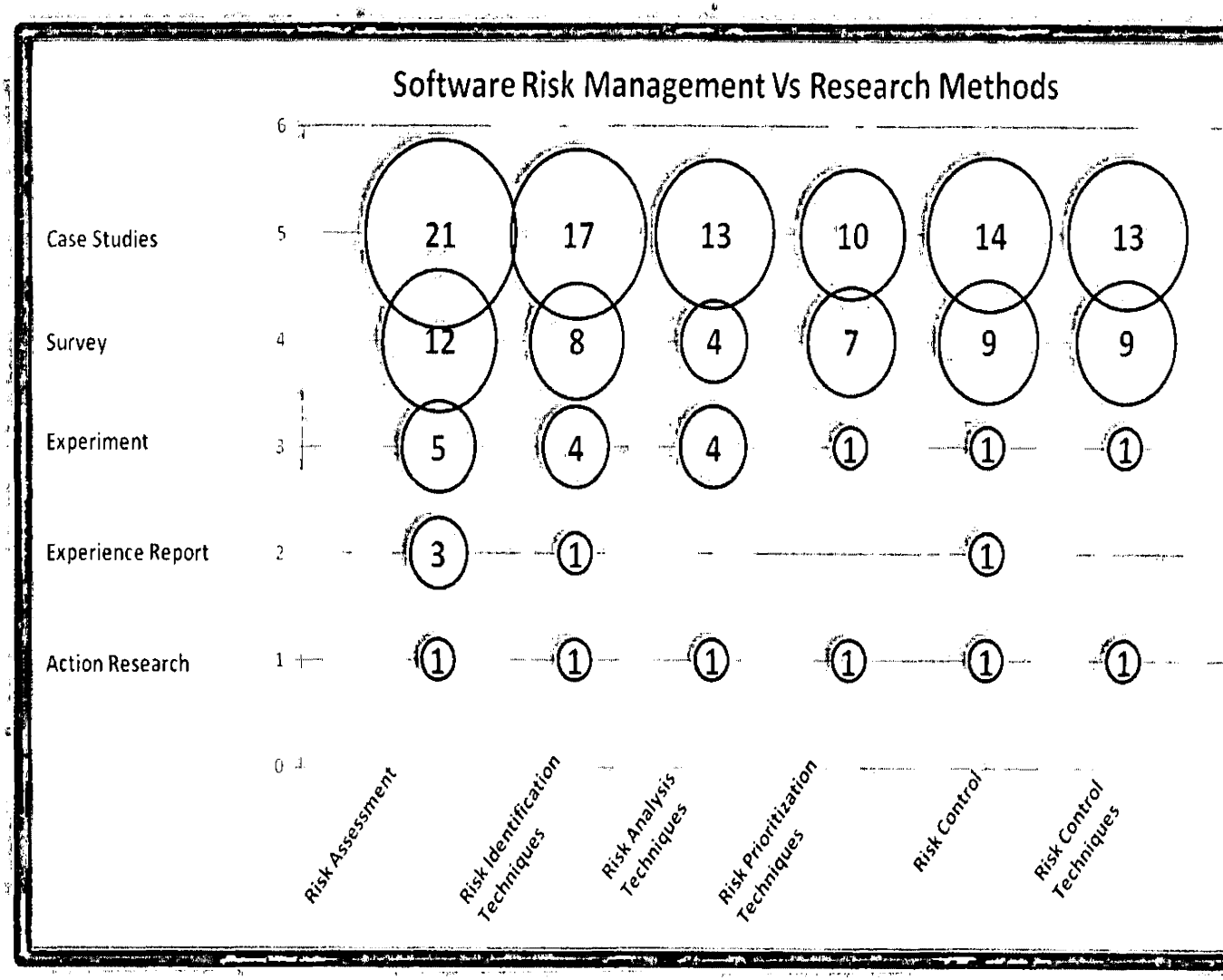

Software Risk Management Vs Research Methods
Case Studies
Survey
Experiment
Experience Report
Action Research
6
5
4
3
2
1
0
21
17
13
10
14
13
12
8
4
7
9
9
5
4
4
1
1
1
3
1
1
1
1
1
1
1
1
1
1
Risk Assessment
Risk Identification Techniques
Risk Analysis Techniques
Risk Prioritization Techniques
Risk Control
Risk Control Techniques

Figure 15 also depicts that Case study and survey are the mostly used research methods in software risk assessment and for risk identification techniques. Experiment is the other methodology that has been used to some extent in risk assessment and risk identification techniques. Figure 12 also shows that there are gaps for action research and experience reports.

4.4 Participants vs. study settings

Figure 16 combines study participants with study settings in a bubble plot. We found that in the field of software risk management major portion of empirical work was done by academia. Figure shows that there were 21 case studies were performed by academia. 24 surveys were conducted from academia. There were only 12 studies from industry in which there were 6 surveys, 3 experience reports, 2 experiments and only one case study.

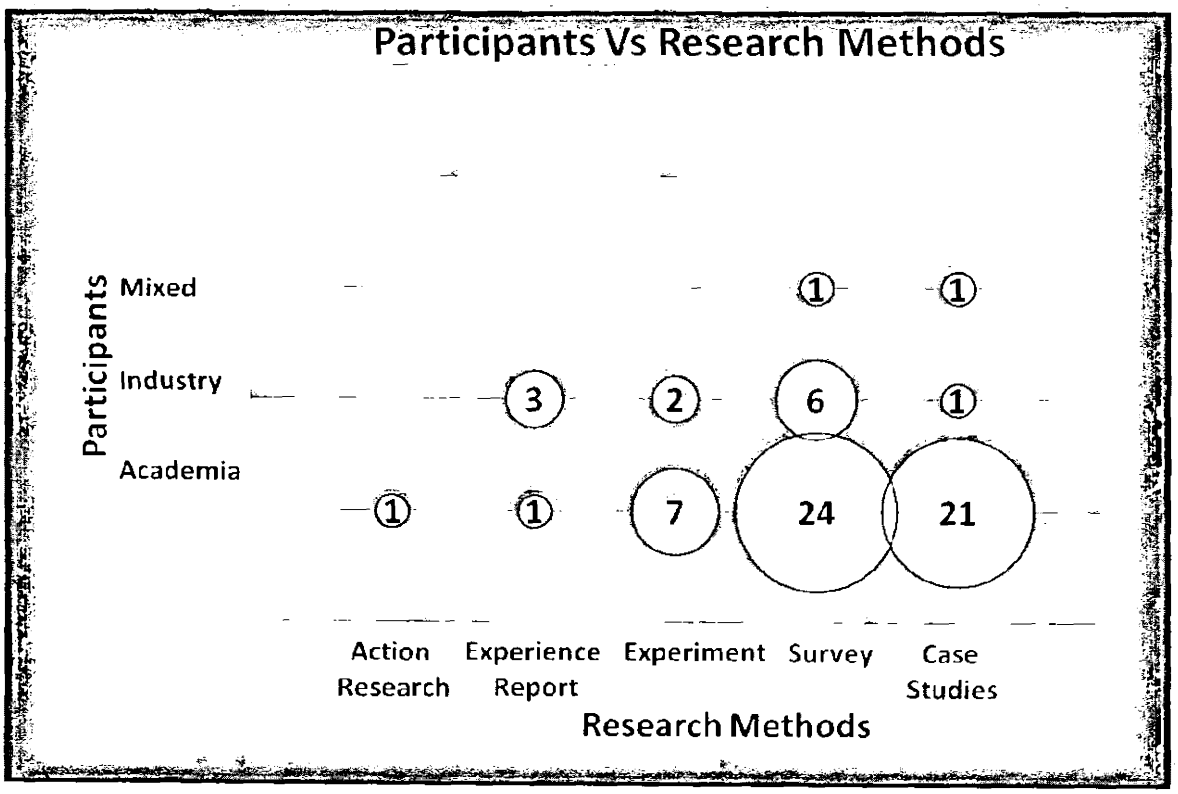


Figure 16: Participants Vs Research Methods

Chapter 5: Conclusion

5. Conclusion

A systematic literature review was conducted to find out the state of art and empirical evidence in the field of software risk management. The objectives of this survey were to provide researchers with an overview of the current state of the methods for software risk management and to provide practitioners with information about the different types of methods, tool, frameworks, techniques etc that could be applied in the process of software risk management. A protocol was defined to conduct this systematic literature review. The timelines of SLR were from 1989 to April 2011. Studies were searched in IEEE, ACM, ScienceDirect and SpringerLink databases. Studies obtained after search were screened for relevance. A predefined inclusion/exclusion criterion was used for study screening. After screening the data was extracted according to data extraction scheme. Extracted data items were stored in Excel sheet. Extracted data was synthesized to answer the research questions. The principal findings of the systematic review are as follows.

5.1 Principal Findings

The results were obtained to answer the research questions. Quantitative analysis, frequency tables and graphs are presented in results chapter to answer research question one (RQ1). Research methods, types of participants, and strength of evidence were discussed to answer second research question (RQ2). The major findings of our study are as:

i. Software risk management is a systematic process. Software risks are managed by following different phases such as risk assessment, risk identification, risk analysis, and risk control.

ii. Software risk assessment, software risk identification and software risk control are the areas that are most researched in the past

iii. Software control techniques and software prioritization techniques are the areas where there is clear lack of empirical studies.

iv. Multiple studies were found collaborative studies in which authors of different countries have contributed for the research of software risk management.

v. USA, Canada, Finland, Norway and Germany are the major countries working in Software risk management research

vi. A number of software risk methodologies (15), frameworks (10), models (12), tools (9) and processes (7) has been proposed and used.

vii. In the field of software risk management, 9 out of 15 risk management methodologies and 11 out of 12 models are defined for software risk assessment.

viii. Whereas 3 out of 10 frameworks are defined for risk categorization purposes.

ix. "Riskit" is the popular methodology used among all the risk management methodologies. Number of usage of Riskit methodology is 4 out of 15.

x. Approximately 80% empirical studies were from academia and rest of the empirical work is from industry and from the collaboration of academia and industry.Because applying software risk management techniques requires a lot of resources, skills at all organizational levels and adequate knowledge of stakeholders.

xi. Mostly the obtained studies were case studies and surveys where participants were from academia.

xii. Interviews and questionnaires were commonly used data collection methods.

xiii. It is notable that majority of the studies are empirical based. Only 21 out of 68 studies were empirically evaluated i.e. where the researchers actually evaluate a method, technique or tool for software risk management.

xiv. Majority of the studies excluded in final screening of full text were due to lack of empirical evidence.

Major contributions are that most of the study objectives are achieved. Findings (i), (ii) and (iii) shows that first two research objective (A&B) are achieved. Whereas findings (vi), (vii), (viii) and (ix) represents that third objective (C) of research is achieved.

5.2 Implications

Software risk management is essential for the successful delivery of software development projects. Good projects may fail without managing risk properly that's why software risk management become very decisive field and quality of product, timelines and product cost have become very serious issue. This is the reason that software industry is focusing on risk management techniques to avoid project failures, budget overruns and time limitation issues. Empirical software engineering is rapidly gaining popularity now days. In software risk management number of empirical work is done but this work is not aggregated. We attempted to aggregate this empirical evidence so that the future researchers can consult this aggregation rather than go through tedious work of fining all the empirical work. This work is also important because it covers the whole empirical work done in all software risk management areas.

5.3 Future Directions

The results in the research could help project managers and practitioners to incorporate the risk factors into their software development methodologies. The obtained results identify gaps in software risk management fields therefore; software risk prioritization and software risk control are the software risk management areas that need more empirical work in the future. Further research however might be possible in the direction of categorizing the risk factors which were identified in the previous work.

References

[1]. Barry W. Boehm, "*Software risk management* ", IEEE Computer Society Press, 1993

[2]. Ray C. Williams, George J. Pandelios, Sandra G. Behrens *Evaluation (SRE) Method Description* (Version 2.0), CMU/SEI-99-TR-029, ESC-TR-99-029, 1999

[3]. Ayad Ali Keshlaf, Steve Riddle, "*Risk Management for Web and Distributed Software Development Projects*" The Fifth International Conference on Internet Monitoring and Protection, 2010 IEEE

[4]. Md. Forhad Rabbi, Khan Olid Bin Mannan, "*A Review of Software Risk Management for Selection of best Tools and Techniques*" , Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008 IEEE

[5]. Jingyue Li, Reidar Conradi[1,2], Odd Petter N. Slyngstadi, Marco Torchiano[3], Maurizio Morisio[3], and Christian Bunse, " *Preliminary Results from a State-of-the-Practice Survey on Risk Management in Off-the-Shelf Component-Based Development*"

[6]. Jingyue Li, Reidar Conradi, Odd Petter N. Slyngstad, Marco Torchiano, Maurizio Morisio and Christian Bunse, "A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 2, MARCH/APRIL 2008

[7]. Mira Kajko-Mattsson and Jaana Nyfjord, "State of Software Risk Management Practice", IAENG International Journal of Computer Science, 35:4, IJCS_35_4_02

[8]. Konstantina Georgieva, Ayaz Farooq, and Reiner R. Dumke, "Analysis of the Risk Assessment Methods – A Survey" , A. Abran et al. (Eds.): IWSM/Mensura 2009, LNCS 5891, pp. 76–86, 2009. © Springer-Verlag Berlin Heidelberg 2009

[9]. Alter S., Ginzberg M. (1978), Managing Uncertainty in MIS implementation, *Sloan Management Review*, Fall.

[10]. Boehm B. W. (1989), *Software Risk Management, Tutorial*, IEEE Computer Society Press, Los Alamitos, California.

[11]. Boehm B. W., Ross R. (1989), Theory-W Software Project Management: Principles and Examples, *IEEE Transactions on Software Engineering*, Vol. 15, No. 7

[12]. Charette R. N. (1989), Software Engineering Risk Analysis and Management, Intertext Publications, McGraw-Hill.

[13]. Chester Simmons, "*Risk Manager, Risk Management*" http://home2.btconnect.com/managingstandard/risk_1.htm

[14]. Richard E. Fairley, "Software Risk Management", Software, IEEE, California State University, Volume 22, Issue 3, May-June 2005 Page(s):101 – 101

[15]. Ronald P. Higuera Yacov Y. Haimes "*Software Risk Management*" Technical Report CMU/SEI-96-TR-012 ESC-TR-96-012 June 1996

[16]. Geoffrey G. Roy, "A Risk Management Framework for Software Engineering Practice" School of Engineering Science, Murdoch University, Perth, Australia 6150. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04) , 2004 IEEE

[17]. Zhang Jun-guang, Xu Zhen-chao, "Method Study of Software Project Risk Management" 201O International Conference on Computer Application and System Modeling (ICCASM 2010)

[18]. Boehm B. W. (1991), Software Risk Management: Principles and Practices, *IEEE Software*, January

[19]. Shukor Sanim Mohd Fauzi Nuraminah Ramli M. Hairul Nizam M. Nasir, "Assessing Software Risk Management Practices in a Small Scale Project" , 2008 IEEE

[20]. Y.H. Kwak , J. Stoddard, "*Project risk management: lessons learned from software development environment*" Project Management Program, Department of Management Science, Monroe Hall 403, School of Business and Public Management, The George Washington University, Washington, DC 20052, USA Agilent Technologies, 2679 Monument Drive, Santa Rosa, CA 95407, USA

[21]. Sarah Beecham et al, "Protocol of a Systematic Literature Review of Motivation in Software Engineering," *Technical Report No. 452* School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire, 2006

[22]. S.Keele, "Giudelines for performing Systematic Literature Reviews in Software Engineering", 2007

[23]. B. A. Kitchenham, O.P. Brereton, D. Budgen, and Z. Li, "An evaluation of quality checklist proposals-A participant-observer case study," in *13th International Conference on Evaluation and Assessment in Software Engineering*, 2009.

[24]. B. Kitchenham, D. I. K. Sjoberg, O. P. Brereton, D. Budgen, T. Dybaa, M. Host, D. Pfahl, and P. Runeson, "Can we evaluate the quality of software engineering experiments?," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010*, p.2.

[25]. M. Host and P. Runeson, "Checklists for software engineering case study research," in *Empirical Software engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 2007*, pp. 1-10.

[26]. D. Budgen and C. Zhang, "Preliminary reporting guidelines for experience papers," in *Proceedings of EASE*, 2009, vol. 2009, pp.1-10.

[27]. T. Dyb\aa and T. Dingosoyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no.9, pp. 833-859, 2008.

[28]. Darja smite et al, "Empirical evidence in global software engineering: A systematic Review" , *Empirical software engineering.*

Appendix

A. Citations

S#	SID	Citation
1	P1	Boehm, B. W. and R. Ross, "*Theory-W Software Project Management Principles and Examples*", Software Engineering, IEEE Transactions, Vol. 15, Number. 7, pp. 902-916, 1989.
2	P3	Barki, H., S. Rivard, et al, "*Toward an assessment of software development risk*", Journal of Management Information Systems, JSTOR, pp. 203-225, 1993.
3	P6	Kontio, J., G. Getto, et al, "Experiences in improving risk management processes using the concepts of the Riskit method", ACM SIGSOFT Software Engineering Notes, ACM, Vol. 23, Number. 6, pp. 163-174, 1998.
4	P7	Keil, M. and Cule, P.E. and Lyytinen, K. and Schmidt, R.C., "A framework for identifying software project risks", Communications of the ACM, Vol. 41, Number. 11, pp.76-83, 1998.
5	P9	Cornford, S. L., M. S. Feather, et al. (2000). "Design and Development Assessment."
6	P10	Yacoub, S.M. and Ammar, H.H. and Robinson, T., "*A Methodology for Architectural-Level Risk Assessment Using Dynamic Metrics", Software Reliability Engineering, ISSRE 2000, Proceedings. 11th International Symposium on, IEEE, pp. 210-221.*
7	P12	Kansala, K., "*Integrating Risk Assessment with Cost Estimation*", Software, IEEE, Vol. 14, Number. 3, pp.61-67, 1997
8	P14	Freimut, B. and Hartkopf, S. and Kaiser, P. and Kontio, J. and Kobitzsch, W., (2001). "*An industrial case study of implementing software risk management*", ACM SIGSOFT Software Engineering Notes, Vol. 26, Number. 5, pp. 277-287, 2001
9	P15	Barki, H. and Rivard, S. and Talbot, J., *"An Integrative Contingency Model of Software Project Risk Management"*, Journal of Management Information Systems, ME Sharpe, Vol. 17, Number. 4, pp. 37-70, 2001
10	P16	Schmidt, R. and Lyytinen, K. and Keil, M. and Cule, P., *"Identifying Software Project Risks: An International Delphi Study"*, Journal of management information systems, ME Sharpe, Vol. 17, Number. 4, pp.5-36, 2001
11	P18	Addison, T. and Vallabh, S., "*Controlling software project risks: an empirical study of methods used by experienced project managers*", Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, South African Institute for Computer Scientists and Information Technologists, pp. 128-140, 2002
12	P20	Neumann, D. E., *"An Enhanced Neural Network Technique for Software Risk Analysis"*, Software Engineering, IEEE Transactions on, Vol. 28, Number. 9, pp. 904-912, 2002
13	P23	Yacoub, S.M. and Ammar, H.H., *"A Methodology for Architecture-Level Reliability Risk Analysis"*, Software Engineering, IEEE Transactions on, Vol. 28, Number. 6, pp. 529-547, 2002
14	P24	Mursu, A. and Lyytinen, K. and Soriyan, HA and Korpela, M., *"Identifying software project risks in Nigeria: an international comparative study"*, European Journal of Information Systems, Palgrave Macmillan, Vol. 12, Number. 3, pp. 182-194, 2003
15	P27	Kontio, J. and Jokinen, J.P. and Rosendahl, E., "*Visualizing and Formalizing Risk Information: An Experiment*", Software Metrics, 2004. Proceedings. 10th International Symposium on, IEEE, pp. 196-206, 2004
16	P29	Wallace, L. and Keil, M., "*Software project risks and their effect on outcomes*", Communications of the ACM, Vol. 47, Number. 4, pp. 68-73, 2004
17	P30	Wallace, L. and Keil, M. and Rai, A., "*Understanding software project risk: a cluster analysis*", Information & Management, Elsevier, Vol. 42, Number. 1, pp. 115-125, 2004
18	P33	Aubert, B.A. and Patry, M. and Rivard, S., "*A framework for information technology outsourcing risk management*", ACM SIGMIS Database, Vol. 36, Number. 4, pp. 9-28, 2005
19	P38	Taylor, H., "*The move to outsourced IT projects: key risks from the provider perspective*", Proceedings of the 2005 ACM SIGMIS CPR conference on Computer personnel research, ACM, pp. 149-154, 2005

S#	SID	Citation
20	P40	Wong, W.E. and Qi, Y. and Cooper, K., "*Source code-based software risk assessing*", Proceedings of the 2005 ACM symposium on Applied computing, ACM, pp. 1485-1490, 2005
21	P43	Li, M. and Huang, M. and Shu, F. and Li, J., "*A risk-driven method for eXtreme programming release planning*", Proceedings of the 28th international conference on Software engineering, ACM, pp. 423-430, 2006
22	P44	Uwadia, C.O. and Ifinedo, P.E. and Nwamarah, G.M. and Eseyin, E.G. and Sawyerr, A., "*Risk factors in the collaborative development of management information systems for Nigerian universities*", Information Technology for Development, Taylor & Francis, Vol. 12, Number. 2, pp. 91-111, 2006
23	P47	Gemino, A. and Reich, B.H. and Sauer, C., "*A Temporal Model of Information Technology Project Performance*", Journal of Management Information Systems, ME Sharpe, Vol. 24, Number. 3, pp. 9-44, 2007
24	P48	Arshad, N.H. and Mohamed, A. and ZAIHA, M., "*Risk factors in software development projects*", Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, pp. 51-56, 2007
25	P49	Ferreira, C. and Marques, P. and Martins, A. and Rita, S. and Grilo, B. and Araujo, R. and Sazedj, P. and Pinto, H., "*Ontology design risk analysis*", On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Springer, pp. 522-533, 2007
26	P50	Costa, H.R. and Barros, M.O. and Travassos, G.H., "*Evaluating software project portfolio risks*", Journal of Systems and Software, Elsevier, Vol. 80, Number. 1, pp. 16-31, 2007
27	P51	Du, S. and Keil, M. and Mathiassen, L. and Shen, Y. and Tiwana, A., "*Attention-shaping tools, expertise, and perceived control in IT project risk assessment*", Decision Support Systems, Elsevier, Vol. 43, Number. 1, pp. 269-283, 2007
28	P52	Hu, Y. and Huang, J. and Chen, J. and Liu, M. and Xie, K., "*Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches*", Natural Computation, 2007. ICNC 2007. Third International Conference on, IEEE, Vol. 3, pp. 358-362, 2007
29	P53	Kappelman, L.A. and McKeeman, R. and Zhang, L., "*Early Warning Signs of it Project Failure: The Dominant Dozen*", Information systems management, Taylor & Francis, Vol. 23, Number. 4, pp. 31-36, 2006
30	P59	Aris, S.R.H.S. and Arshad, N.H. and Mohamed, A., "*Conceptual framework on risk management in IT outsourcing projects*", management, Vol. 36, Number. 37, pp. 38, 2008
31	P63	Slyngstad, O. and Li, J. and Conradi, R. and Babar, M., "*Identifying and Understanding Architectural Risks in Software Evolution: An Empirical Study*", Product-Focused Software Process Improvement, Springer, pp. 400-414, 2008
32	P64	Slyngstad, O.P.N. and Conradi, R. and Babar, M.A. and Clerc, V. and van Vliet, H., "*Risks and Risk Management in Software Architecture Evolution: An Industrial Survey*", Software Engineering Conference, 2008. APSEC'08. 15th Asia-Pacific, IEEE, pp. 101-108, 2008
33	P66	Odzaly, E.E. and Greer, D. and Sage, P., "*Software risk management barriers: An empirical study*", Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on, IEEE, pp. 418-421, 2009
34	P69	Ho, L.T. and Lin, G. and Nagalingam, S., "*A risk mitigation framework for integrated-enterprise systems implementation for the manufacturing environment*", International Journal of Business Information Systems, Inderscience, Vol. 4, Number. 3, pp. 290-310, 2009
35	P71	Islam, S. and Joarder, M.M.A. and Houmb, S.H., "*Goal and Risk Factors in Offshore Outsourced Software Development from Vendor's Viewpoint*", Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, IEEE, pp. 347- 352, 2009
36	P76	Warkentin, M. and Moore, R.S. and Bekkering, E. and Johnston, A.C., "*Analysis of systems development project risks: an integrative framework*", ACM SIGMIS Database, Vol. 40, Number. 2, pp.8-27, 2009
37	P77	Nakatsu, R.T. and Iacovou, C.L., "*A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study*", Information & Management, Elsevier, Vol. 46, Number. 1, pp. 57-68, 2009
38	P78	Arshad, N.H. and Mohamed, A. and Mansor, R., "*The effects of implementing organizational structural and risk management strategies in information system projects*", risk, Vol. 9, Number. 4,

S#.	SID	Citation
		pp. 3, 2009
39	P79	Sun, S., "*Study on Software Project Risk Priority Management and Framework Based on Information Management System*", Information Science and Engineering (ICISE), 2009 1st International Conference on, IEEE, pp. 2402-2405, 2009
40	P81	Williams, L. and Gegick, M. and Meneely, A., "*Protection Poker: Structuring Software Security Risk Assessment and Knowledge Transfer*", Engineering Secure Software and Systems, Springer, pp. 122-134, 2009
41	P82	Zhang, Y. and Shi, X., "*Offshore software outsourcing risk evaluation: an experimental approach base on linear mixed model*", Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on, IEEE, Vol. 1, pp. 505- 509, 2009
42	P83	Basit, A. and Murtaza, G. and Ikram, N., "*Validation of VRRM process model*", Proceedings of the 9th WSEAS international conference on Software engineering, parallel and distributed systems, World Scientific and Engineering Academy and Society (WSEAS), pp. 190-195, 2010
43	P84	Büyüközkan, G. and D. Ruan., "*Choquet integral based aggregation approach to software development risk assessment*", Information Sciences, Elsevier, Vol. 180, Number. 3, pp. 441-451, 2010
44	P85	Mc Caffery, F. and Burton, J. and Richardson, I., "*Risk management capability model for the development of medical device software*", Software Quality Journal, Springer, Vol. 18, Number. 1, pp. 81- 107, 2010
45	P90	Royce, W., "*TRW's Ada Process Model for incremental development of large software systems*", Proceedings of the 12th international conference on Software engineering,
46	P92	Sherer, S. A. (1992). "*Assessing the risk of software failure in a funds transfer application.*" System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on. IEEE Computer Society Press, pp. 2- 11, 1990
47	P99	Kobee, R.M. and Schrank, M.J. and Anderson, A.C. and Boyce Jr, G.W., "*A risk-based approach for the evaluation on software maintenance options for militarized avionics*", Digital Avionics Systems Conference, 1995., 14th DASC, IEEE, pp. 326- 330, 1995
48	P104	Hudepohl, J.P. and Aud, S.J. and Khoshgoftaar, T.M. and Allen, E.B. and Mayrand, J., "*Integrating metrics and models for software risk assessment*", Software Reliability Engineering, Proceedings., Seventh International Symposium on, IEEE, pp. 93- 98, 1996
49	P105	Mayrand, J. and Coallier, F., "*System acquisition based on software product assessment*", Proceedings of the 18th international conference on Software engineering, IEEE Computer Society, pp. 210- 219, 1996
50	P114	Weyuker, E. J., "*Predicting project risk from architecture reviews.*" Software Metrics Symposium, Proceedings. Sixth International, IEEE, pp. 82- 90, 1999
51	P139	Kontio, J. and Jokinen, J.P. and Rosendahl, E., "*Visualizing and formalizing risk information: an experiment*", Software Metrics, Proceedings. 10th International Symposium on, IEEE, pp. 196- 206, 2004
52	P149	Song, X. and Stinson, M. and Lee, R. and Albee, P., "*A Qualitative Analysis of Privilege Escalation*", Information Reuse and Integration, IEEE International Conference on, IEEE, pp. 363- 368, 2006
53	P169	Persson, J.S. and Mathiassen, L., "*A Process for Managing Risks in Distributed Teams*", Software, IEEE, Vol. 27, Number. 1, pp. 20- 29, 2010
54	P170	Souza, E. and Gusmao, C. and Alves, K. and Venancio, J. and Melo, R., "*Measurement and control for risk-based test cases and activities.*" Test Workshop, LATW '09, 10th Latin American, IEEE, pp. 1- 6, 2009
55	P171	Kwan, T.W. and Leung, H.K.N., "*Measuring Risks within a Program Consisting of Multiple Interdependent Projects*", Computational Intelligence and Software Engineering, CiSE 2009, International Conference on, IEEE, pp. 1- 7, 2009
56	P174	Bhuiyan, M. and Rana, S. and Krishna, A., "*Evaluating effectiveness of risk identification and management using organisational models*", Computer Research and Development (ICCRD), 3rd International Conference on, IEEE, Vol. 4, pp. 278- 282, 2011
57	P175	Kwan, T.W. and Leung, H.K.N., "*A Risk Management Methodology for Project Risk Dependencies*", Software Engineering, IEEE Transactions, Vol. 37, Number. 5, pp. 635- 648, 2011
58	P179	Nedstam, J. and Host, M. and Regnell, B. and Nilsson, J., "*A Case Study on Scenario-based Process*

S#	SID	Citation
		Flexibility Assessment for Risk Reduction", Product Focused Software Process Improvement, Springer, pp. 42- 56, 2001
59	P183	Port, D. and Yang, Y., "*Empirical Analysis of COTS Activity Effort Sequences*", COTS-Based Software Systems, Springer, pp. 169- 182, 2004
60	P184	Li, J. and Conradi, R. and Slyngstad, O. and Torchiano, M. and Morisio, M. and Bunse, C., *"Preliminary Results from a State-of-the-Practice Survey on Risk Management in Off-the-Shelf Component-Based Development"*, COTS-Based Software Systems, Springer, pp. 278- 288, 2005
61	P185	Šmite, D.,*"A Case Study: Coordination Practices in Global Software Development"*, Product Focused Software Process Improvement, Springer, pp. 25- 46, 2005
62	P188	Taylor, P. and Greer, D. and Sage, P. and Coleman, G. and McDaid, K. and Lawthers, I. and Corr, R., *"Applying an Agility/Discipline Assessment for a Small Software Organisation"*, Product-Focused Software Process Improvement, Springer, pp. 290- 304, 2006
63	P192	Grossi, V. and Romei, A. and Ruggieri, S., *"A Case Study in Sequential Pattern Mining for IT-Operational Risk"*, Machine Learning and Knowledge Discovery in Databases, Springer, pp. 424-439, 2008
64	P194	Hossain, E. and Babar, M.A. and Verner, J., *"How Can Agile Practices Minimize Global Software Development Co-ordination Risks?"*, Software Process Improvement, Springer, pp. 81- 92, 2009
65	P198	Sheng, Z. and Tsuji, H. and Sakurai, A. and Yoshida, K. and Nakatani, T., *"Preliminary Analysis for Risk Finding in Offshore Software Outsourcing from Vendor's Viewpoint"*, Software Engineering Approaches for Offshore and Outsourced Development, Springer, pp. 134- 148, 2009
66	P205	Pourdarab, S. and Nosratabadi, H.E. and Nadali, A., *"Risk Assessment of Information Technology Projects Using Fuzzy Expert System"*, Digital Information and Communication Technology and Its Applications, Springer, pp. 563- 576, 2011
67	P206	{Asnar, Y. and Giorgini, P. and Mylopoulos, J., *"Goal-driven risk assessment in requirements engineering"*, Requirements Engineering, Springer, Vol. 16, Number. 2, pp. 101- 116, 2011
68	P207	Bannerman, P.L., *"Risk and risk management in software projects: A reassessment"*, Journal of Systems and Software, Elsevier, Vol. 81, Number. 12, pp. 2118- 2133, 2008

B. Pilot Study

Sr.#	Year	Title	Evidence
1	2010	Method Study of Software Project Risk Management	Case study
2	2010	Risk Management for Web and Distributed Software Development Projects	Survey review
3	2009	Risk Management in the Trustworthy Software Process: A Novel Risk and Trustworthiness Measurement Model Framework	Case study
4	2009	The Application of Fault Tree Analysis in Software Project Risk Management	Experiment
5	2009	The Role of Software Process Simulation Modeling in Software Risk Management: a Systematic Review	SLR
6	2009	Software Risk Management Barriers: an Emoirical Study	Survey
7	2009	Insight into Risk Management in Five Software Organizations	survey(interview thru open ended questionaire)
8	2008	Risk Management Method using Data from EVM in Software Development Projects	Case study
9	2008	Risks and Risk Management in Software Architecture Evolution: an Industrial Survey	Questionaire based survey
10	2008	Software Risk Management: Practice Contra Standard Models	Questionaire, interviews
11	2008	A Review of Software Risk Management for Selection of best Tools and Techniques	Review
12	2008	A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components	Survey (questionaire)
13	2008	Assessing Software Risk Management Practices in a Small Scale Project	Survey (questionaire)
14	2006	Prompt List for Risk Management in Sri Lankan Software Industry	Questionaire, interview, pilot survey
15	2004	Security Risks: Management and Mitigation in the Software Life Cycle	Case study

C. External Reviewer Comments

Review Comments:

Reviewer: Saad Zafar

Topic: Protocol of Systematic Review of empirical literature in Software Risk Management

Author: Saima Irum

1. Overview
    a. Good topic
    b. Protocol description & discussion should be written more carefully.
    c. Some obvious careless mistakes bring the quality of work considerably down.
2. Preamble
    a. The text is unstructured and has some unnecessary repetition.
3. Background
    a. Some abbreviations are used before they are introduced.
    b. The need for SLR must be clearly articulated.
    c. The SLR motive should be discussed in the light of previous studies (similarities, differences, etc.)
4. Research Questions (in Background Section)
    a. There are two sets of Research Questions given.
    b. I am assuming that these are intended set or questions.
    c. Question 2:
        i. What does the term "from where" mean?
            1. The place or origin?
            2. The institution?
            3. The literature?
    d. How the areas of SRM would be identified?
    e. How would gaps in the literature be identified?
    f. What is the difference between the last two questions?
5. Research Questions
    a. These questions appear to be copied from somewhere else.
    b. The subsequent discussion makes it hard to follow because the text keeps referring to RQ1 & R Q2. However, if we consider the first set of questions then there should be four questions instead of only two.
6. Search Strategy
    a. The search string for RQ1 has an "OR" with "Risk Management" only. This would bring in results from non-software related domains. Has this point been considered?
    b. Same is the case with RQ2.
    c. Step 3:
        i. Where these search strings are coming from? Is this list exhaustive? How?

d. Other sources to be searched:
    i. The feasibility of the following points should be seriously considered:
        1. Contacting key authors
        2. Going for references in primary studies
e. Study selection criteria
    i. Refer to the point which states "Disagreement between reviewers". How many reviewers will be there?
f. Exclusion Criteria:
    i. Refer to the point "we will include highest ranked paper". What would you do if there is a tie?

7. Search Process Documentation
    a. Again, only two questions are referred here!
    b. It puts the credibility and seriousness of the whole document in questions.
    c. This also makes it hard to review.
8. Please evaluate the feasibility of the SLR in light of the available literature.

D. Protocol

Protocol

Software Risk Management: A Systematic Review

Saima Irum

135-FBAS/MSSE-F06

Protocol Version 1.4

12th April 2011

Department of Computer Science and Software Engineering

Faculty of Basic and Applied Sciences

International Islamic University Islamabadhttp://iiu.edu.pk

Review Title:

Protocol of Systematic Review of Empirical Literature in Software Risk Management

Internal advisor

Dr. Naveed Ikram, Associate Professor, RIU, Pakistan

Research Areas: Requirement Engineering, Risk Management, Software Architecture

naveed.ikram@iiu.edu.pk

External Reviewers

Table of Contents

APPENDIX A: SEARCH STRINGS FOR ACM

APPENDIX B: SEARCH STRINGS FOR IEEE XPLORER

Protocol of Systematic Review of Empirical Literature in Software Risk Management

Preamble

Software risk management is essential for the successful delivery of software development projects. During the last ten years, the software industry is paying more and more attention towards software risk management (SRM). According to Barry W. Boehm "software risk management is an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful software operation or major source software rework" [1].

Risk management is a systematic and continuous process. SEI (Software Engineering Institute) risk management paradigm describes this best. SEI paradigm involves some sequential, concurrent and iterative activities. Those are to identify, plan, track, control and communication of risk [2]. Software Risk Management (SRM) is carried out with the help of different SRM processes, models, frameworks, tools, techniques and methodologies e.g., Software Risk Evaluation (SRE), Team Risk Management (TRM), ARMOR, EBIOS Methodology, ProRisk Framework, Riskit Method, SoftRisk, CMMI-RSKM, PMBOK RM Process, GDSP RM Framework, Risk and Performance Model [3],[4]. Risk management techniques get significant attentions because without managing risk properly very good project may fail as SRM has become very crucial field in these days and quality of product, timelines and product cost has become very serious issue. This is the reason that software industry is using risk management techniques to avoid project failures, budget overruns and time limitation issues.

Academia and industry both are well aware of the importance of software risk management that is why there exists lots of literature on various sub areas of software risk management. But there is a need to summarize and aggregate this literature to find out actual status of the field, identify gaps, scope for further research and quality of the work. That is the reason to undertake this systematic literature review. This document provides an outline of the protocol for SLR and it is developed based on the guidelines of (Kitchenham. 2007).

Background

The main motive to undertake this systematic review is to identify gaps and state-of-the-art in empirical research related to software risk management and summaries the existing empirical evidence to provide base for future research and practical use. Similar work exists in several

studies where researchers summarized the available literature and pointed out future directions but the focus of those studies was not empirical evidence and none of the study reported qualitative and quantitative evaluation of data at a time.
Primary research on software risk management focused on defining guidelines for specific tasks [7], [8], [9], [10] but these provide little empirical evidence for the practical usefulness of risk management. Later in 1990's B.W. Boehm, Charette's, Software Engineering Institute (SEI) risk management methods and Hall's risk management principles have increased industry awareness and improved practice. SEI defines risk as the possibility of suffering loss. [5] Richard E. Fairley defined 'risk' as "The probability of incurring a loss or enduring a negative impact." [6] Software risks are increasing as long as software development industry is increasing. Risks in software should be handled properly because they greatly affect software development process and its outcomes. Software development involves five types of risks: 1) financial risk; 2) technical risk; 3) project risk; 4) functionality risk; 5) political risk. Risk Management consists of the processes, methodologies and tools that are used to handle with these risks in the Software Development Life Cycle (SDLC) process of a Software Project.

SEI risk program is very useful in order to improve the process of software acquisition and software development [11]. The basic methodological framework introduced for the software acquisition and software development are: Software Acquisition Capability Maturity Model (SA-CMM^{SM}) and Software Capability Maturity Model (SW-CMM^{SM}). These methodologies are supported by Software Risk Evaluation (SRE), Continuous Risk Management (CRM), Team Risk Management (TRM) practices. The SRE practice, developed by the SEI, is used for identifying, analyzing, communicating, and mitigating software technical risk. Its primary functions are Detection, Specification, Assessment, and Consolidation and supporting functions are Planning, Coordination, Verification, Training and Communication. The Continuous Risk Management (CRM) practice is used for managing project risks and opportunities throughout all the activities of the project. Team Risk Management (TRM) is a risk management based on team oriented activities in which both customer and supplier together apply the methodologies. SRE, CRM and TRM are based on three basic constructs of SEI these are Risk Management Paradigm, Risk Taxonomy, and Risk Clinic. While implementing these methodologies and practices SEI experience shows that software risk is the least measured and managed during the lifecycle of software development.

Geoffrey G. Roy [12] introduced a ProRisk framework for risk management. ProRisk framework provides a process to analyze and identify the key risk factors, outcomes, reactions and the creation of action plan to mitigate these risks. ProRisk Framework is built on a hierarchical model structure defined in the SEI taxonomy of risk and Karolak [4] and involves following activities. 1) Stakeholder Identification, 2) Risk Factor Identification, 3) Risk Tree Model Construction, 4) Calibrating the Model, 5) Estimating the Risk Event Probabilities, 6) Computing Combined Risk Values, 7) Developing Action Plans, 8) Monitoring the Progress, 9)

Operationalizing the Framework. This framework focuses on the business domain and the operational domain of the software development. In business domain level it identifies the economic environment of the organization and the weakness of the organization to expose risks factor. It also identifies the knowledge, experience and confidence of the organization to successfully complete the project. In Operational domain it measures the risk values, identify the key risk factors, identify and describes the action plans to reduce the key risk factors, implement action plans and then re-assess the risk key factors. It is continuous cyclic process so it continuously monitor and document risk properties, and provide support for risk mitigation and management on a continuous basis. This framework can be applied to both small scale and complex projects, with controllable levels of data requirements. ProRisk framework covers the complete life cycle of the Project development and provides support to run risk analysis activities in parallel with the project management activities. It is also supported with a ProRisk tool.

Literature has many studies and surveys on software risk management practices. In [14] authors represent a study of methods that are used in software risk management process from the first step of risk identification to the last step of risk control. It provides a risk checklist for risk identification gives step by step procedure for the risk assessment and provides complete risk management plan. Another survey presented in [4] gives suggestions for the selection of best tools and techniques for software risk management. It provides the limitations and beneficial qualities of Software Risk Evaluation (SRE), Team Risk Management (TRM), Softrisk tool, ARMOR (Analyzer for Reducing Module Operational Risk), Riskit technique and CMM based risk control optimization model.

Implementing risk management means to insert the risk management principles and practices into software development life cycle. Best implementation strategy is to use incremental model because in this way risk management practices can easily be adjusted in organizational culture. [13]. Software Risk Management Practices in a Small Scale Project were accessed in [15]. They have selected a small scale project, purpose of the study was to access the strengths and weakness of the risk practices in that project. They have selected Capability Maturity Model Integration (CMMI) model to measure the gap between current software risk management practices in selected project. After assessment findings they also used Standard CMMI Appraisal Method for Process Improvement (SCAMPI) for the formal characterization. The characterization indicator indicates that the software risk management practices are Partially Implemented in that project.

The practical implementation of the ongoing processes of software risk management is challenging task. Organizations that implement effective tools and techniques in software development project are successful by changing organization culture [16]. Implementing software risk management is a crucial process because it requires resources, skills at all organizational levels, adequate knowledge of stakeholders. That's why literature has many

studies and surveys on software risk management practices. But there is no study in the literature with a focus on empirical evidence. Evaluating empirical evidence is equally important for academia and software industry, as gathering and summarizing empirical evidence systematically will help researchers in future research and practitioners will get quantified measures to make informed decisions.

Research Questions

The research questions are phrased considering the overall objective of this systematic literature review so that these questions can capture the existing empirical knowledge of software risk management field. By answering these research questions, needs and opportunities for future research will be identified from existing empirical literature. Moreover the strength and validity of identified empirical literature will also be identified.

Two Research Questions

RQ.1: **What is the state-of-the-art in empirical studies of software risk management?**

The purpose of this question is to evaluate the status of the software risk management field with an empirical perspective, and provide guidance for future progress in this area. The data obtained as an answer of this question will be evaluated quantitatively in terms of frequency of occurrence and will depict the mature and underdeveloped areas of software risk management along with other relevant information in terms of quantity of the studies.

RQ.2: What is the strength of empirical evidence reflected in empirical software risk management literature?

The aim of this question is to find out the strength of empirical evidence in terms of source of evidence and methods used. Strength of empirical evidence is important for future research. The studies obtained for both of these questions will be same but the main difference is in the perspective, for this question data will be evaluated for quality of work to know what is the source of data and what study design have been used to obtain this evidence etc.

The overall Evidence based investigation is focused on the type of question given by guidelines of (Kitchenham. 2007) "Assessing the frequency or rate of project development factor such as the adoption of a technology of the frequency of project success or failure" And "identify and/or

scope future research activities". So the research questions will be assessing the future research scope by aggregating the available literature.

Structured Questions:

***RQ.1:* What is the state-of-the-art in empirical studies of software risk management?**

Population: software projects

Outcome: Status of the software risk management field

No Intervention, No Comparison

RQ.2: What is the strength of empirical evidence reflected in empirical software risk management literature?

Population: software projects

Outcome: strength of empirical literature

No Intervention, No Comparison

Search Strategy

Strategy for Search terms

For our search strategy we have taken inspirations from the protocol of Sarah Beecham *[15]* and modified according to our requirements. The search process will be conducted according to the following decided steps:

5. We will derive major search strings from PICO;
6. Identify alternative spellings and synonyms for major terms; also alternative terms used in literature will be considered
7. When database allows, use the Boolean OR to incorporate alternative spellings and synonyms; and when database allows, use the Boolean AND to link the major terms from population, intervention and outcome.
8. Tools used for automating the search process

Step1: Major Search Terms:

RQ1:

- Software Risk Management
- SRM
-

RQ2:

- Software Risk Management
- SRM
- Empirical

Step 2: alternative spellings and synonyms for major terms and use of Boolean And and OR:

RQ1:

((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management plan OR software risk management case study OR software risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment)

RQ2:

((Software Risk Management OR Software Risk Management Metrics OR Software Risk Management Plans OR Software Risk Management Tools OR Software Risk Management Barriers OR Risk Management Software OR Software Project Risk Management OR Software Risk Management Techniques OR Software System Risk Management OR Software Development Risk Management OR CMM based Software Risk Management OR Software Project Risk Management Framework OR Risk Management Enterprise Software OR Software Risk Management Process OR software risk management plan OR software risk management case study OR software risk management practices OR software engineering risk management OR software risk assessment OR software risk identification OR software risk checklist OR software risk analysis OR software risk prioritization OR software risk exposure OR software risk reduction OR software risk management planning OR software risk control OR software risk plan integration OR software risk avoidance OR software risk element planning OR software risk resolution OR software risk reassessment) **AND** (Empirical OR Industrial OR Experiment OR Case Study OR Survey))

Step5: Tools used for automating the search process

To speed up the process of documenting my search results we will be using following two tools which will help in automating the process.

- End notes [for documenting bibliography from searches]; we will retrieve search results in End notes and later on the basis of titles and abstracts we will include or exclude papers. It will help in documenting the search results for both primary searches and for primary studies.
- Zotero [it will help us to collect, manage, and cite our research sources.]

Resources to be searched

A range of data bases has been selected for rigorous search and to reduce the bias. Following data bases will be searched for the retrieval of primary studies:

1. Springer link
2. IEEE Explore
3. ACM Digital library
4. Science Direct (www.sciencedirect.com)

Study selection criteria

- The initial selection will be on the basis of the TITLE and ABSTRACT of the paper.
- All obtained data from the search process will be archived in database according to the journal from which it is retrieved.
- From data base the duplicates will be removed after initial scan of results.
- Inclusion and exclusion criteria will be applied on the results to sort out the accepted papers.
- On accepted papers detail inclusion criteria which is a Quality Instrument for studies, will be applied to see whether they qualify to be included or not.
- Full papers of all studies that are not clearly ineligible will then be obtained.
- The excluded papers and reasons for exclusion will be recorded in a file, and the included papers and study type will be recorded in another file.

Study inclusion and Exclusion Criteria

The criteria are intended to identify those studies that provide direct evidence for the research questions. Following are the inclusion and exclusion criteria for our research questions:

Inclusion criteria:

In order to answer the stated research questions, we have been searching for research articles by reading the abstracts that contain information about software risk management (SRM) and some empirical work done as well. So the paper can be a case study, an experiment, survey, experience report etc. when it will be confirmed after reading abstract that the article is relevant to our research, and then we will study the whole paper. The objective of the selection process was to identify the articles relevant for the objectives of

the systematic review. The search strings, were quite broad and hence it was expected that not all studies identified would make it to the final phase in the selection process. Only those studies will be included that are empirical based i.e. experiment, case study, survey or industrial experience report and where the main focus is Software Risk Management or SRM.

Exclusion criteria:

- Those studies will be excluded that are based on personal expert opinion.
- Literature surveys and books will be excluded.
- Only one inclusion for studies with the same results reported multiple times.
- Multiple studies can be reported in one paper; if any of them do not stand to our inclusion criteria then only that study will be excluded.

Quality Instrument for assessing validity

After initial selection of studies, a more detail criteria is required to judge the quality of study to see whether it is worth considering as evidence to answer our research question or not.

Quality Instrument will be designed for assigning numerical values for factors in the checklist to be evaluated for each study. My main focus is on the **study design.**

The research questions and our inclusion / exclusion criteria suggest us that we are going to have evidence in form of empirical studies like case studies, industrial experience reports etc.

So firstly we will create a check list for assessing the quality of study and assigning numerical values to the questions so we can rank the obtained papers.

If any paper is considered to be very poor in quality it will be excluded at this stage and will be recorded in the file of excluded papers with reasons.

One paper can report multiple studies, in that case those studies will be evaluated individually for their criteria to be included or excluded.

Checklist for assessing criteria of Case Studies

We have followed the checklist provided by Martin and Runeson [16] for reviewing a case study. Final score will rank the studies depending on the weights assigned after checklist is applied to them.

Each question will be marked as "yes=1", "partly=0.5" and "no=0"

1. Are the research questions, objects of study and case study context well defined?
2. Are the data collection procedures sufficient for the purpose (data sources, collection, storage, validation)?
3. Is a clear chain of evidence established from observations to conclusions?
4. Are threats to validity analyses addressed in a systematic way?
5. Are different views taken on the case (multiple collection and analysis methods, multiple authors)?

Quality assessment checklists for other empirical data e.g. survey, experiment and experience report etc will be used from guidelines of Kitchenham [2007].

Search Process Documentation

Primary Search Documentation

The customized search strings will be applied to the data bases according to decided strategy. As the process will go on, the results will be saved by the following decided strategy.

i. Two main folders by name of RQ1 and RQ2 will be created.
ii. Within each folder further sub folders by the name of specified data based or journal will be created.
iii. Within each folder for different search string terms different folders will be created by the name of their IDs.
iv. All records will be maintained for one search string in library of reference manager software (endnotes).
v. Results of that specific search strings will be placed in that folder created by the name of that search string.

vi. Same process will be performed for both research questions and on all data bases.

vii. Duplicates (in papers and studies in papers) will be removed from data base after scanning the records.

viii. After applying inclusion/exclusion criteria within the folder by the name of journal, I will create two folders for included and excluded papers. This will also give indication that which journal gave more evidence then others.

ix. Data base will be updated for included and excluded papers.

x. Reasons for exclusion will be recorded in a file.

xi. All included papers will be moved to one folder.

xii. Conflicts for papers where inclusion or exclusion is ambiguous will be consulted according to the decided rules, another file will be created to record these activities, the decisions will be recorded accordingly and papers will either be accepted or rejected.

Secondary Search Documentation

From accepted primary studies, secondary searches will be made and same procedure will be followed as was followed for the documentation of primary searches.

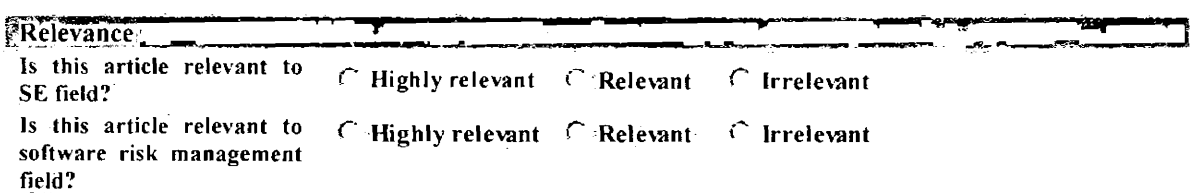
Data Extraction

The data will be extracted with the help of a classification scheme. This scheme captures data regarding relevance, empirical background, Software Risk management background, and focus of the study. In addition, a qualitative evaluation of the papers will also be performed. The qualitative evaluation is useful to cross-check the view of the different researchers performing the review. Each of these areas is further elaborated as:

Extracted Data Type	**Corresponding Section**	**Description of Extracted Data**
Technical and methodological flaws of the study	Relevance	A study contains empirical evidence
		A study is relevant to software risk management

		A study is relevant to SE
		A study does not repeat other included studies(relation to other papers)
Information about the sample, population or participants	Empirical Background	Main method and sub methods
		Background (industry Vs Laboratory)
		Subjects of investigation
		Empirical focus: Empirically-based vs. Empirically-evaluated
	Software Risk Management Background	SRM Processes
		Types of risks
		Number of risks
		SRM Practices
		SRM methodologies
		SRM frameworks
		SRM Models
		SRM techniques
		SRM tools
		Risk Identification
		Risk Assessment
		Risk Control
Central focus of the study and problem addressed	Study	Development methodology
		Focus of the study(practice, Phase or other)
		Evaluation of the study in terms of success
		Application Domain
		Definitions in introduction section
Review of the key results	Qualitative Evaluation	Claims
		Personal Evaluations
		Recommendations

Data Extraction Form



Is this an empirical study?	○ Yes ○ No
Does this article repeat already reviewed article(s)?	○ Yes ○ No

Empirical Background

Main Method	☐ Survey ☐ Case Study ☐ Interviews ☐ Controlled Experiment ☐ Survey
Sub-Method	☐ Survey ☐ Case Study ☐ Interviews ☐ Archive Analysis ☐ Controlled Experiment ☐ Other...
Background	☐ Laboratory ☐ Industry/Real world
Subjects of investigation	☐ Students ☐ Industry/Real world
Empirical focus	☐ Empirically based ☐ Empirically evaluated

Software Risk management Background

SRM Processes	
Types	
Number of risks	
SRM Practices	
SRM methodologies	
SRM frameworks	
SRM Models	
SRM techniques	
Software tool support	
Risk Identification	
Risk Assessment	
Risk Control	

Study

Development Methodology	○ Unclear
Focus of the Study	○ SRM in General ○ Single Practice(s) ○ Development Phase(s) ○ Others...
Success or failure?	☑ Clear success story ☐ Success of practices described ☐ Clear failure story ☐ Failure of practices described ☐ Evidence of SRM related problems ☐ Unclear ☐ Other...
Application Domain	☐ Telecom ☐ Automative ☐ Web ☐ Finance ☐ Automation ☐ Unclear ☐ Other...
Definitions in the introduction-like sections?	☐ No ☐ Software Risk Management ☐ Other related definitions

Qualitative Evaluation

Claims	Narrative
Personal reflection	Narrative
Recommendations	Narrative

Data Analysis and Synthesis

Extracted data will be analyzed using quantitative and qualitative synthesis methods. The research areas in the field of software risk management will be identified along with gaps and future directions. The classification scheme used in data extraction will help to separate the concerns and categories. Relationships among various categories of data will also be pointed out with multiple perspectives. After depicting data in quantitative summaries a thorough qualitative analysis of the data will also be performed to evaluate the strengths of the literature. The expected outcome will contain information like; what's the most widely used empirical method applied by the researchers and practitioners in software risk management? Either researchers or practitioners who are most involved in software risk management research and in which specific sub area of software risk management? What's the source of empirical evidence etc? This information will be depicted in form of systematic maps like Bar graphs, Bubble plots etc.

Validation of Review Process

The final version of the protocol will be formed after performing two steps: Protocol Evaluation and Pilot testing.

Protocol Evaluation

Following strategy will be applied for evaluation;

- The protocol will be initially given for peer review.
- Then will be evaluated from supervisor.
- It would then be sent for external evaluation to selected panel of independent reviewers.

Protocol will be updated into final version after comments from external reviewers and pilot testing.

Schedule of Activities

Planning the Review			
Activity	**Starting Date**	**Completion Date**	**Comments**
Protocol development Version V1.1	20th March 2011	10th April 2011	
Protocol v1.2	20th April 2011	28th April 2011	
Protocol v1.3 [Final version] Sent for external review	20th May 2011	20th july 2011	
Conducting the Review			
Activity	**Starting Date**	**Completion Date**	**Comments**
Evidence Collection and Critical Evaluation			
Primary Searches			
Secondary Searches and Study Selection			
Data Extraction			
Data Analysis and Presentation			
Data Synthesis			
Reporting the Review			
Activity	**Starting Date**	**Completion Date**	**Comments**
First Draft			
Final Draft			

Pilot Testing

Sr.#	Year	Title	Evidence
1	2010	Method Study of Software Project Risk Management	Case study
2	2010	Risk Management for Web and Distributed Software Development Projects	Survey review
3	2009	Risk Management in the Trustworthy Software Process: A Novel Risk and Trustworthiness Measurement Model Framework	Case study
4	2009	The Application of Fault Tree Analysis in Software Project Risk Management	Experiment
5	2009	The Role of Software Process Simulation Modeling in Software Risk Management: a Systematic Review	SLR
6	2009	Software Risk Management Barriers: an Empirical Study	Survey
7	2009	Insight into Risk Management in Five Software Organizations	survey(interview thru open ended questionaire)
8	2008	Risk Management Method using Data from EVM in Software Development Projects	Case study
9	2008	Risks and Risk Management in Software Architecture Evolution: an Industrial Survey	Questionaire based survey
10	2008	Software Risk Management: Practice Contra Standard Models	Questionaire, interviews
11	2008	A Review of Software Risk Management for Selection of best Tools and Techniques	Review
12	2008	A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components	Survey (questionaire)
13	2008	Assessing Software Risk Management Practices in a Small Scale Project	Survey (questionaire)
14	2006	Prompt List for Risk Management in Sri Lankan Software Industry	Questionaire, interview, pilot survey
15	2004	Security Risks: Management and Mitigation in the Software Life Cycle	Case study

References

[1]. Barry W. Boehm, "*Software risk management* ", IEEE Computer Society Press, 1993

[2]. Ray C. Williams, George J. Pandelios, Sandra G. Behrens *Evaluation (SRE) Method Description* (Version 2.0), CMU/SEI-99-TR-029, ESC-TR-99-029, 1999

[3]. Ayad Ali Keshlaf, Steve Riddle, "*Risk Management for Web and Distributed Software Development Projects*" The Fifth International Conference on Internet Monitoring and Protection, 2010 IEEE

[4]. Md. Forhad Rabbi, Khan Olid Bin Mannan, "*A Review of Software Risk Management for Selection of best Tools and Techniques* " , Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008 IEEE

[5]. Chester Simmons, "*Risk Manager, Risk Management*" http://home2.btconnect.com/managingstandard/risk_1.htm

[6]. Richard E. Fairley, "Software Risk Management", Software, IEEE, California State University, Volume 22, Issue 3, May-June 2005 Page(s):101 – 101

[7]. Alter S., Ginzberg M. (1978), Managing Uncertainty in MIS implementation, *Sloan Management Review*, Fall.

[8]. Boehm B. W. (1989), *Software Risk Management, Tutorial*, IEEE Computer Society Press, Los Alamitos, California.

[9]. Boehm B. W., Ross R. (1989), Theory-W Software Project Management: Principles and Examples, *IEEE Transactions on Software Engineering*, Vol. 15, No. 7

[10]. Charette R. N. (1989), Software Engineering Risk Analysis and Management, Intertext Publications, McGraw-Hill.

[11]. Ronald P. Higuera Yacov Y. Haimes "*Software Risk Management*" Technical Report CMU/SEI-96-TR-012 ESC-TR-96-012 June 1996

[12]. Geoffrey G. Roy, "A Risk Management Framework for Software Engineering Practice" School of Engineering Science, Murdoch University, Perth, Australia 6150. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04) , 2004 IEEE

[13]. Boehm B. W. (1991), Software Risk Management: Principles and Practices, *IEEE Software*, January

[14]. Zhang Jun-guang, Xu Zhen-chao, "Method Study of Software Project Risk Management" 2010 International Conference on Computer Application and System Modeling (ICCASM 2010)

[15]. Shukor Sanim Mohd Fauzi Nuraminah Ramli M. Hairul Nizam M. Nasir, "Assessing Software Risk Management Practices in a Small Scale Project" , 2008 IEEE

[16]. Y.H. Kwak , J. Stoddard, "*Project risk management: lessons learned from software development environment*" Project Management Program, Department of Management Science, Monroe Hall 403, School of Business and Public Management, The George Washington University, Washington, DC 20052, USA Agilent Technologies, 2679 Monument Drive, Santa Rosa, CA 95407, USA