SYSTEMATIC LITERATURE REVIEW OF EMPIRICAL EVIDENCE IN SOFTWARE ARCHITECTURE

A THESIS PRESENTED TO

FACULTY OF BASIC & APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCES & SOFTWARE ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE

OF

MS IN SOFTWARE ENGINEERING

BY

NADIA QURESHI

236-FBAS/MSSE/F08

SUPERVISED BY

DR. NAVEED IKRAM

CO-SUPERVISED BY

Mr. MUHAMMAD USMAN

DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING
FACULTY OF BASIC & APPLIED SCIENCES
INTERNATIONAL ISLAMIC UNIVERSITY
H-10, ISLAMABAD
(JULY, 2012)

1- software reliability

2- Text processing (computer science)

DEPARTMENT OF COMPUTER SCIENCES & SOFTWARE ENGINEERING INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

Date: 4-12-12

Final Approval

This is to certify that we have read the thesis submitted by **NADIA QURESHI,** Registration No. **236-FBAS/MSSE/F08.** It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of **MS in Software Engineering.**

Committee:

External Examiner:

Internal Examiner:

Supervisor:

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

DEDICATION

I would like to dedicate my research work to the

HOLIEST man Ever Born on Earth, PROPHET MUHAMMAD (Peace Be Upon Him) and
I also dedicate my work to my parents

Their prayers and motivation were source of strength for me to do this research work successfully.

Nadia Qureshi

Nadia Qureshi
236-FBAS/MSSE/F08

A dissertation submitted to the

Department of Computer Science & Software Engineering,

Faculty of Basic and Applied Sciences,

International Islamic University,

Islamabad,

as a partial fulfillment of the requirements

for the award of the degree of

MS in Software Engineering (MSSE)

DECLARATION

I hereby declare that this Thesis **Systematic Literature Review of Empirical Evidence in Software Architecture** neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have written this thesis entirely on the basis of my personal efforts, made under the proficient guidance of my thesis supervisor, **Dr. Naveed Ikram.**

If any part of this research thesis proved to be copied or found to be a research of some other individual, I shall standby the consequences.

No portion of the research work presented in this thesis report has been submitted in support of any other degree or qualification of this or any other university or institute of learning.

Nadia Qureshi

Nadia Qureshi
236-FBAS/MSSE/F08

ABSTRACT

Software architecture (SA) plays pivotal role in development and maintenance of large software systems. Architectural decisions impact all subsequent phases in software development life cycle. Structure of large software systems has been in discussion since early 70s but software architecture started emerging as separate discipline since mid-90s. Since mid-90s, researchers have been publishing empirical studies (i.e. case studies, experiments, experience reports) in SA discipline. Empirical literature has been aggregated in other disciplines of software engineering but no such effort has been attempted in SA. Objective of this study is to aggregate and synthesize the empirical literature of software architecture to report the trends, patterns and knowledge gaps. To synthesize the empirical work in SA, a systematic literature review (SLR) has been conducted. This study reports the results of SLR based on 247 included primary studies. Based on the percentage of 247 primary studies we identified that SA Evaluation (28%), non- functional requirements related work (22%) and SA Design (12%) are relatively mature sub areas of SA. Most of the empirical work (59%) in SA uses case study research method. Experiments (20%) and experience reports (14%) are also employed in empirical SA work. SA discipline is in maturing phase. Few sub areas of SA are mature and some areas are new and being developed. We have also identified few emerging trends in SA i.e. service oriented architecture, Product line architecture, Aspect oriented architecture, model driven architecture.

ACKNOWLEDGEMENT

In the name of Allah, Most Gracious, Most Merciful, whose blessings made it possible for me to complete this thesis. I am thankful to Almighty Allah who blessed me to contribute in the knowledge of world. I would like to express my gratitude to my supervisor Dr. Naveed Ikram for support, guidance and motivation throughout this thesis. His ideas and directions were of great value, without his guidance and support it was not possible for me to complete this research work. I am also thankful to Mr. Muhammad Usman and Miss Munera Bano for their guidelines; they were always there to help me in improving my research work. I am also grateful to all my friends and fellow students at IIU who motivated and helped me during this thesis. Lastly I cannot express my gratitude enough to my family for their patience, support and understanding throughout this thesis.

Nadia Qureshi

Nadia Qureshi
236-FBAS/MSSE/F08

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SLR	Systematic Literature Review
SA	Software Architecture
RQ	Research Question
EBSE	Evidence Based Software Engineering
ADL	Architecture Description Language
AKM	Architecture Knowledge Management
GSD	Global Software Development
SEBoK	Software Engineering Body of Knowledge
WCSA	Working Conference on Software Architecture
ECSA	European Conference on Software Architecture
SHARK	Sharing and Reusing Architecture Knowledge
QoSA	Conference Series on Quality of Software Architecture
LNCS	Lecture Notes on Computer Sciences
JSS	Journal of System and Software
ICSE	International Conference on Software Engineering
RE	Requirement Engineering

CHAPTER # 1 INTRODUCTION

1 INTRODUCTION

Software Architecture (SA) is a separate discipline in Software Engineering (SE). There are many definitions of SA in literature [1][2][3][4]; a definition of SA states that, "*The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them*" [5]. Software Architecture acts as a skeleton for the software development. SA needs to be created early during the software development and then the whole development process revolves around this skeleton, keeping into account the constraints and facilities implied by the software architecture. Few decades back there was nothing like architecture. The concept of software architecture was first introduced in 1968 when layering was used in program development [6] then this concept was enhanced and structure of software was introduced [7][8]. Increasing complexity and software quality needs urged the practitioners to opt modularity and ultimately it turned into the form which is now called software architecture. SA emerged as a separate discipline in 1990 [9][3]. Software architecture is responsible for incorporating quality in software by accommodating quality attributes and functional requirements. Moreover software architecture must have to accommodate the continuous changing needs so it should be flexible enough to evolve.

Software Architecture is very important in software development and it has positive impact on overall development and if properly selected SA can lead to success. Research shows that SA has positive impact on at least five aspects of software development i.e. understanding, reuse, evolution, analysis and management [10]. SA helps to enhance the understandability by simplifying the complex systems, it improves our ability to comprehend large and complex systems by presenting them at high level of abstraction at which level the design can be understood easily [2][3]. Architects model various requirements during architecting phase and resultant architecture acts as a bridge among various stakeholders in terms of communication. Requirement engineers, designers, testers, managers etc all deal with same architecture but these stakeholders have their own perspective. Architecture provides a wide range of styles and patterns that enhance reusability aspect in software development. Along with these styles, domain reference architectures do the samė, product family shares the common architectures. Instead of developing an application from scratch, available components compatible with the architecture are used. Architectures are transformable in accordance with the everyday changing technologies. This quality of SA makes modifications and evolution of software very easy. Management activities like task allocation, work breakdown structures, scheduling, budgeting etc are controlled by SA. Quality requirements are entertained by Software Architecture rather almost every stage of software development is either dependent or facilitated by SA.

1.1 Problem Description

Academia and industry both are well aware of the importance of Software Architecture that is the reason that a lot of empirical literature exists in various sub areas of software architecture. This empirical literature is not aggregated yet, therefore there is a need to summarize and aggregate this literature to find out the actual status of the field, to identify gaps, scope for the further research and to find out the quality of the evidence. There is much work in the literature that points towards the need of systematically gather empirical evidence in software Engineering [11] [12]. This is the reason to undertake this systematic literature review.

Software Engineering is shifting towards evidence based software engineering and that is why there is lots of empirical evidence in software architecture area. In the past, researchers have reported state-of-the-art in SA and reported the status of the field but there was no such roadmap that tells about the empirical evidence in SA literature. Such type of literature synthesizing attempts has been made in other areas of software engineering like GSD etc. There is existing literature that point outs the need for this kind of aggregation of empirical evidence in SA [11] [12].

1.2 Background and Motivation

The main motive to undertake this systematic review was to identify all empirical research related to software architecture, aggregate the empirical studies and summaries the evidence for future 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. These studies are described as follows:

The concept of software architecture as a separate discipline started to emerge in 1990 [9][3] and defended later [14]. Since then the key research areas of software architecture and its future directions have been identified time to time by conducting literature and industrial surveys [9][10][16]. The research paradigms used in software architecture research have been focused by identifying the types of research questions which were structured to use and the research design devised to answer those questions; state-of-the-art in software architecture with a perspective of growth in technology maturation model has also been described [17][18]. The chronological history of the software architecture field, its innovative methods, tools, techniques, software architecture community, papers, books and conferences has already been aggregated [14].

The above mentioned studies aggregated the existing literature of software architecture and each of these studies has different concerns and varying scope. These studies were carried out as normal literature surveys without following a systematic process. None of these studies attempted to aggregate the evidence based software architecture work. Evaluating empirical evidence is equally important for academia and software industry, as systematically gathering and summarizing empirical evidence will help researchers in future research and practitioners will also get quantified measures to make informed decisions [19]. There is much work that points towards the need to systematically gather empirical evidence in Software Engineering [11][12]; therefore conducting the research using a systematic and unbiased methodology is necessary. Mapping study [20] and many SLR exist in the field of software architecture [21][22][15]. They differ from our review in a way that their scope is limited to one sub-area of software architecture as opposed to our review. We focus on whole SA discipline. Moreover our review is focused on only empirically supported evidence based SA studies.

1.3 Study Objectives

Purpose of conducting this systematic literature review is to summarize the existing research and to find the gaps in the literature in software architecture area using a thorough, systematic and unbiased methodology [13]. Systematic literature review findings are used for drawing conclusions that can be more useful and thorough as compare to single study; these conclusions are valuable for future research [13].

1.4 Methodology

This study was conducted by using systematic literature review (SLR) methodology and it was conducted using the guidelines of Kitchenham [13]. SLR is a secondary study in which multiple primary studies are used. Primary studies are searched using search terms. These searched terms are identified from research questions. These primary studies are used to extract the information required to answer the research questions. In SLR a detailed protocol is developed before conducting the actual review. This protocol is a detailed plan of the whole SLR. Then this protocol is executed to find the results. SLR has three main steps [13] that need to be followed while conducting a systematic literature review; these steps are planning, conducting and reporting the review. Systematic review protocol is the outcome of the planning phase. Systematic review protocol is the detailed plan that describes the whole review procedures. It is better to develop a pre-planned protocol before conducting systematic literature review [13]. In conducting phase the detailed plan of the whole SLR (protocol) is executed. The results are reported at the end of the SLR. Detailed sub-phases of SLR are described in the next sub-section.

1.4.1 Formulating the Research Questions (RQ)

Formulation of the RQ is the first step of SLR. It is used to state the research question that depicts the aim and objective of the study [13]. There may be one or multiple RQs for a SLR but at least two questions are considered as good practice for conducting a SLR [13].

1.4.2 Search Strategy

Search strategy is the strategy that is used to search the relevant studies from search databases. This strategy is defined in the planning phase before conducting the SLR. There are phases of defining a search strategy in which major search terms are identified from RQs. Alternate words and varying spellings of these search terms are identified. All the search terms are concatenated by using "AND" and "OR" to form a search string. These search strings are used to search the relevant studies from resources [13].

1.4.3 Study Selection Criteria

During SLR planning, the study selection criterion is defined. This criterion has detailed inclusion and exclusion criteria. Purpose of defining this criterion is to include only relevant paper and to exclude irrelevant one. These criteria are used to include and excluded the studies, these criteria are pre-defined by the researcher at the time of protocol development [13].

1.4.4 Study Quality Assessment

Study quality assessment is a phase in which a checklist is defined to assess the study quality. Included studies are assigned score according to the checklist items. This checklist is also called quality assessment Instrument. Quality assessment checklist can be adopted from previous SLRs or can be customized according to the objective of the SLR. Selected studies (after inclusion exclusion of relevant and irrelevant studies) are assessed for quality [13].

1.4.5 Data Extraction

In this step, data is extracted by using data extraction form (data extraction sheet in our case). Extraction sheet is designed in accordance with the objective of the study so that relevant data is extracted that can answer the research questions [13]. Data or information required to answer the research question is extracted from each study using a data extraction form.

1.4.6 Synthesis

In synthesis step extracted data is gathered and summarized into results. Synthesis can be qualitative or quantitative [13].

1.4.7 Reporting the Review

At the end, a review report is written to report the results of the SLR. This review report is evaluated and disseminated.

1.5 Thesis Organization

Chapter 1 consists of Introduction to Software Architecture are, Study Objectives, Research Questions and SLR Methodology. Chapter 2 explains protocol development phases and its revisions; this chapter elaborates the whole protocol in detail including search strings, study selection criteria, data extraction schemes and synthesis methods etc. Chapter 3 has details of protocol execution explaining all the phases of SLR at execution time. Chapter 4 presents the results and analysis for this research. Chapter 5 concludes the research and explains the outcome of the study listing future directions and implications in software architecture area.

CHAPTER # 2 PROTOCOL DEFINITION

2 PROTOCOL DEFINITION

Protocol definition is the planning phase of the SLR. A detailed plan of the SLR is defined and finalized before conducting the review. Protocol document is the outcome of this phase. This chapter contains the detailed plan i.e. protocol of this SLR.

2.1 SLR Protocol

The methodology used for this study is systematic literature review. This chapter describes the protocol development phase of the SLR. Initially a protocol was developed; this initial draft was refined gradually on the basis of pilot results and external reviewer's comments. The protocol was evaluated by an eminent researcher form of Software Engineering Dr. Muhammad Ali Babar. His area is Empirical Software Engineering and specifically Software Architecture. The reviewer's comments are listed in Appendix B. Protocol is a detailed plan of the whole SLR process which changes over time. This chapter describes the final version of the protocol after making desired changes in the initial version protocol. Initial version of protocol is attached in Appendix D.

2.2 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 architecture 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. The research questions formulated for this research are about state-of-the-art and strength of empirical evidence.

RQ1: What is the state-of-the-art in empirical studies of software architecture?

The purpose of this question is to evaluate the status of the software architecture 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 architecture along with other relevant information in terms of quantity of the studies.

RQ2: What is the strength of empirical evidence reflected in empirical software architecture 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.

The overall evidence based investigation is focused on the type of question given in the guidelines of Kitchenham[13]. "*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 assess the future research scope by aggregating the available literature.

2.3 Search Strategy

In this phase the major search terms will be identified from the RQs and their synonyms and alternate spellings will be used to form the search strings.

2.3.1 Major Search Terms

The steps for extracting search terms are as:

i. Derive major search strings from research questions.

ii. Identify alternative spellings and synonyms for major terms.

iii. When database allows, use the Boolean OR to incorporate alternative spellings and synonyms.

iv. When database allows, use the Boolean AND to link the major terms.

v. Major search terms identified from the RQ in our study's context are: Software Architecture (A), Empirical (B).

vi. Alternative spellings and synonyms for major terms and use of Boolean "And" and "OR": (Software OR System) (Architecture OR Structure OR Design) (Empirical OR Industrial OR Case study OR Experiment OR Experience Report OR Lesson learned)

In order to answer the stated research questions, search strategy need to be defined before conducting the review. Research articles based on empirical evidence, with either professional software developers or students as participants, were the main focus of this literature review. Studies focusing Software architecture were considered. The final search strings were selected on the basis of experience from the pilot search and consisted of the following terms:

A1—software architecture

A2—software structure

A3—software design

A4—system architecture

B1—empirical

B2—case study

B3—industrial

B4—experiment

A5—system structure	B5—experience
A6—system design	B6—lessons learned

The search string consisted of Boolean expression: (A1 OR A2 OR A3 OR A4 OR A5 OR A6) AND (B1 OR B2 OR B3 OR B4 OR B5 OR B6). The terms selected for search were quite general for both dimensions used in the search (A and B as shown above). That is why a high proportion of irrelevant papers were obtained, but this reduced the risk of not finding the relevant papers. For string expression the term "software" and its synonyms were concatenated with the term "architecture" and its synonyms as without this concatenation search result brought hardware architecture and basic engineering architectural studies as well.

The search strategy has the following decisions that were adopted according to SLR guidelines [13].

Items:	Journal articles, workshop and conference papers.
Apply search on:	Abstract
Language:	The papers written in English.
Publication period:	Since 1972 to 2010

This review considered Journal articles, workshop papers, conference papers and technical reports written in English language and published since 1972 (After D. Parnas explained the decomposition criteria) [7]. Since 1972 structure of system and designs were focused in the published work that is why we decided to used this time duration so that we can obtain maximum number of related studies in out SLR.

2.3.2 Search Resources

Following data bases were searched for the retrieval of the studies: Springerlink, IEEE Explore, ACM Digital library, ScienceDirect and EI Compendex. Databases were selected keeping in view that leading publication channels of Software Architecture are in IEEE, ACM, SpringerLink and ScienceDirect. Some Proceedings of SHARK and ICSE are published in ACM and some in IEEE. QoSA published in ACM. LNCS is available on SpringerLink. JSS is available on ScienceDirect. EI Compendix brings papers from multiple sources, that is why to further minimize the chances of not finding relevant papers we included EI Compendix in our search database resources list.

2.4 Study Selection Criteria

Study selection criteria were multiphase. Included and excluded studies of each phase were stores separately. Study selection criteria details are:

i. Search Strings was applied on the above mentioned databases and obtained references was archived in a Reference Database using reference manger software.

ii. Duplicates were identified and removed.

iii. The titles of studies were assessed by using the inclusion criteria.

iv. In the next step the abstracts were assessed upon inclusion criteria.

v. Full text of studies were assessed based upon inclusion/exclusion criteria

vi. As the inclusion/exclusion criteria was multiphase so the results of each screening phase was maintained in separate libraries. (EndNote libraries).

vii. The papers that were not clearly relevant/irrelevant were included/excluded in discussion meeting with research supervisor and research co-supervisor.

2.4.1 Study Inclusion/Exclusion Criteria

The inclusion and exclusion scheme was multiple phrased. The objective of this multiphase study selection process was to identify the articles relevant to the objectives of this systematic literature review. The search strings were quite broad and hence it was expected that all studies identified will not be included in the final phase. Table 1 shows study selection criteria. This criterion was applied on title and abstract in the first phase and then it was used to screen the studies upon fulltext. The purpose of step by step screening was to ensure that only relevant studies are included in the final SLR inclusion.

Table 1: Study selection criteria

Study Selection Criteria	
Relevance Analysis	Inclusion Criteria
Selection of studies based on the search (these decisions were incorporated in the search string so that the searched studies bring relevant studies)	Only written in English
	Date of publication: 1972—Present
	Only Published work
	Contains the search strings
Screening upon titles	Not editorials, prefaces, discussions, comments, summaries of tutorials, panels or duplicates
Screening upon abstracts	Check focus of the study is SA and empirical evidence exists.
Discussion Meeting	Meeting for the Inclusion/Exclusion of doubtful papers.
Screening upon full text	Presence of empirical data in the paper
	Originality of empirical evidence(only one inclusion for studies with the same results reported multiple times
	Sufficient focus on software architecture

2.5 Search Process Documentation

Search strings were applied on the selected sources and obtained results were saved in different folders on the basis of Database and Screening Phases (multiple phases of inclusion/exclusion criteria). The categorization was implemented by making folders and saving files in these folders. Then after assessment of the studies accepted papers was copied to another folder.

2.6 Quality Instrument for Quality Assessment

Quality Instrument will be used to assign quality score to the studies as a support for data analysis and synthesis. The Quality Instrument consists of 5 sections; the main section contains generic checklist items applicable to all the studies while other 4 sections are specific for research design used in the study. These sections are survey, case study/Action Research/Ethnographic Study, experiment/Quasi Experiment and experience report. These criteria are based upon SLR guidelines [13], along with revised set of items adopted from various checklists that have already been used [23][24][25][26][27]. The detailed checklist is given in Table 2.

Table 2: Quality Assessment Checklist

Quality Assessment Checklist	
Generic [13] [27]	
Are the aims clearly stated?	YES/NO
Are the study participants or observational units adequately described?	YES/NO/PARTIAL
Was the study design appropriate with respect to research aim?	YES/NO/PARTIAL
Are the data collection methods adequately described?	YES/NO/PARTIAL
Are the statistical methods justified by the author?	YES/NO
Is the statistical methods used to analyze the data properly described and referenced?	YES/NO
Are negative findings presented?	YES/NO/PARTIAL
Are all the study questions answered?	YES/NO
Do the researchers explain future implications?	YES/NO
Survey [13] [27]	
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/NO

Quality Assessment Checklist	
Have "drop outs" introduced biasness on result limitation?	YES/NO/NOT APPLICABLE
Experiment/ Quasi Experiment [23][24]	
Were treatments randomly allocated?	YES/NO
If there is a control group, are participants similar to the treatment group participants in terms of variables that may affect study outcomes?	YES/NO
Could lack of blinding introduce bias?	YES/NO
Are the variables used in the study adequately measured (i.e. are the variables likely to be valid and reliable)?	YES/NO
Case Study/ Action Research/ Ethnographic Study [25]	
Is case study context defined?	YES/NO
Are sufficient raw data presented to provide understanding of the case?	YES/NO
Is the case study based on theory and linked to existing literature?	YES/NO
Are ethical issues addressed properly (personal intentions, integrity issues, consent, review board approval)?	YES/NO
Is a clear Chain of evidence established from observations to conclusions?	YES/NO/PARTIAL
Experience Report [26]	
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 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]. Some of the checklist items will be graded on yes/no/partial. Scores will also be assigned according the grades, 2 for yes, 0 for No and 1 for partially. The total sum of the scores will be used for the quality assessment of studies.

The purpose of this study was to aggregate the empirical evidence related to SA. That is why we decided not to exclude any study on the basis of study quality. Therefore we did not decide any pass/fail criterion for quality checklist. We only assigned the quality scores and aggregated it show the clear picture regarding study quality.

2.7 Data Extraction

Data extraction was performed by using data extraction sheet. Extracted data was entered into the MS Excel. Each paper selected for data extraction was assigned a unique ID. The data extraction form obtained information about:

i. Title
ii. Search Database
iii. Journal / conference
iv. Year of publication
v. Publication channel
vi. Geographical area
vii. Affiliated Organization
viii. Method of Study
ix. Quality assessment ranking
x. Software Architecture Area
xi. Software Architecture Sub Area
xii. Emerging Trends
xiii. Application Domain
xiv. Data collection Method
xv. Study participants
xvi. Research Out put
xvii. Research Type

The data was extracted in extraction sheet. The extracted data items were specifically relevant to research questions. The data was extracted with the help of a classification scheme. This scheme captured data regarding some generic data, data about empirical settings of the study, data specific to Software architecture area. All the extracted data was relevant to the objective of the study and in useful to answer the RQs. Table 3 shows Data Extraction scheme and extracted data fields.

Table 3: Data Extraction Scheme

Data Extraction Scheme		
Extracted Data Type	Corresponding Section	Description of Extracted Data
Generic Information	Generic Section	Study ID
		Title
		Year
		Affiliation Organization
		Location
		Publication Chanel
		Database
		Quality Assessment Ranking
Software Architecture Area Information	Software Architectural Background	SA Main Area
		SA sub-Area
		Application Domain
		Research Output
		Emerging Trends
Empirical Method information	Study Settings	Empirical Method
		Data Collection Method
		Study Participants
		Research Type

Based upon the above mentioned categorization scheme data extraction sheet was designed. In this sheet some fields/questions were given with a set of known answers from the software architecture field. The details of those fields with known set of answers are described in table 4.

Table 4: Fields of Data Extraction Sheet

Data Extraction Sheet Items		
Data Extraction Item	Known Set of Answers	Description
Study ID		Unique ID of the study Assigned after study selection
Title		Study Title
Year		Year of publication
Affiliation Organization		Organization name to which the study is affiliated
Location		Study country of origin
Publication Chanel		Journal/conference name where study was published
Database	IEEE, ACM, SpringerLink, ScienceDirect, IE Compendex	Resource from where study was obtained
Quality Assessment Ranking		Aggregated Quality Score of the Study
SA Main Area	Gen. Software Architecture	=>These SA Areas were Initially identified from SEBoK. But as the study proceeded we included other areas as well because we obtained studies related to other areas of SA which were initially not listed in SEBoK, like SA and RE, SA Transformations etc. =>Views and ADLs were collectively named as documentation and Description. => SA and RE is not any Area in SA but it was included because we got many studies that were relevant to SA and Requirement Engineering as well.
	SA Design	
	SA Analysis	
	SA AKM	
	Software Architecting Process	
	SA Erosion	
	SA Evolution	
	SA Re-engineering	
	SA Conformance	
	SA Recovery	
	SA Transformations	
	SA description and documentation	
	SA misalignment	
	SA Issues and Challenges	
	SA & RE	
SA sub-Area	Gen. Software Architecture	Initially this field was not included in the data extraction but as data extraction proceeded it was noticed that almost each study dealt with at least 2 areas of SA that is why this field was introduced in extraction sheet.
	SA Design	
	SA Analysis	
	SA AKM	
	Software Architecting Process	
	SA Erosion	

Data Extraction Sheet Items		
	SA Evolution	
	SA Re-engineering	
	SA Conformance	
	SA Recovery	
	SA Transformations	
	SA description and documentation	
	SA misalignment	
	SA Issues and Challenges	
	SA & RE	
	Usability	
	Modifiability	
	Reliability	
	Stability	
	Flexibility	
	Maintainability	
	Dependability	
	Interoperability	
	Availability	
	Portability	
	Performance	
	NFRs (illitics)	
Application Domain	Medical	This field was used to capture the domain where the study was performed. =>In Automation studies were from industry or academia where the system/application was an automation of certain process =>Generic type of domain deals with applications like compilers, middleware etc =>Study conducted in more than one domains was listed as multiple domains
	Robotics	
	Defense	
	MIS	
	Avionics	
	Automotive	
	Automation	
	Generic	
	Telecom	
	Web	
	Finance	
	Multiple Domains	
Research Output	New Language	An output scheme was decided to be

Data Extraction Sheet Items		
	New Tool	used to capture the output of the study. The study may propose new tool/technique/ model/method/ process/framework/language or it can modify existing tool/technique/ model/method/ process/framework/language or can use these. That is why this scheme was used to capture the output of the study and it worked very well.
	New Technique	
	New Model	
	New Framework	
	New Process	
	Modification of Language	
	Modification of Tool	
	Modification of Technique	
	Modification of Model	
	Modification of Framework	
	Modification of Process	
	Usage Experience of Language	
	Usage Experience of Tool	
	Usage Experience of Technique	
	Usage Experience of Model	
	Usage Experience of Model	
	Usage Experience of Process	
	Guidelines	
Emerging Trends	Product Line	This field was used to capture the technology and trends focused during the study. The list of known answers were populated as the extraction process proceeded and these techniques were encountered in the studies
	Distributed	
	SOA	
	GSD	
	Component Based	
	Model Driven	
	Aspect Oriented	
	Pervasive Computing	
	Open Source	
	Agile	
	Ontology Driven	
Empirical Method	Case Study	Action Research, Ethnographic Study and Quasi Experiment were included this list after these were encountered as research method in some studies.
	Experiment	
	Experience Report	
	Survey	
	Action Research	

Data Extraction Sheet Items		
	Ethnographic Study	
	Quasi Experiment	
Data Collection Method	Interview	=>Focus group was added in the list during data extraction. =>Unclear was added in the list as so many studies lack a clear description of data collection method
	Focus Group	
	Archive Analysis	
	Observation	
	Questionnaire	
	Unclear	
Study Participants	Academia	
	Industry	
	Mixed	
Research Type	Validation Research	The research type classification was adopted from existing research [28][29]
	Opinion Paper	
	Philosophical Paper	
	Experience Paper	
	Solution Proposal	

Data extraction and quality assessment were the prime responsibility of the first author. Second author (co-supervisor) validated the study selection and data extraction. In case of any problem first and second author consulted third author (supervisor) for arbitration. In analysis all the three authors participated.

2.8 Piloting:

Piloting was done during searching the studies from various resources to validate the correctness of search string. Some papers were identified as included papers and then string was applied on the database to check that it has searched those known papers from the SA area. The second piloting was done for selection criteria and data extraction. Few papers were selected on the basis of inclusion and exclusion criteria. Data was extracted by using data extraction sheet to verify whether the extracted data is appropriate for answering the research questions.

2.9 Data Analysis and Synthesis

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

2.10 Validation of Review Process

The Review process needs to be evaluated by some experts in the area that is why an internal reviewer (Supervisor) and one external reviewer (Dr. Muhammad Ali Babar) reviewed the SLR protocol before execution of the protocol. Moreover this protocol was sent to an international peer- reviewed conference from where we got comments from three reviewers. Protocol for this SLR is published [30].

i. The protocol was initially evaluated by research supervisor Dr. Naveed Ikram.

ii. It was then sent for external evaluation to an independent reviewer Dr. Muhammad Ali Babar. He sent back the protocol document after reviewing it along with his comments upon various sections of the protocol.

iii. Protocol was updated according to the comments of external reviewer and its final version was executed to conduct SLR. The protocol described in this chapter is the final version of protocol.

The comments received by the external reviewer are listed in appendix B along with the answered solutions. The protocol was updated based upon external reviewer's suggestions and resent to the reviewer.

Summary

This chapter covers the whole SLR planning phase and its refinements. We started planning phase with identification of major search terms from research questions. Then the synonyms and alternate spellings of all the major terms were identified. These terms and synonyms were concatenated using Boolean operators. A string was formed by using all the search terms and Boolean "OR" and "AND". The search decisions were finalized in which time duration of search, targeted search databases, languages and targeted search items were decided. After search decisions a whole screening criterion was decided. In order to select relevant studies a multi phase inclusion and exclusion criterion was decided to be used for inclusion and exclusion of the studies. After that screening data extraction schemes and data extraction items were decided. There were three type of information that was decided to be extracted from the selected studies. This was generic information about the study like title, country, year, publication channel etc and the software architecture related information like specific sub area of SA, type of output of the study, domain of the study etc. Finally the extraction items related to empirical evidence were finalized like type of research, study settings, types of participants etc. A tentative analysis plan of the study was also decided in the plan. In SLR the output of the planning phase is SLR Protocol and this chapter elaborates the whole protocol. This protocol was revised several times according to the comments of research supervisor and according to an external independent reviewer's comments. This chapter consisted of final version of the protocol which was then executed in the conducting phase of SLR.

CHAPTER # 3 PROTOCOL EXECUTION

3 PROTOCOL EXECUTION

After the development of initial draft of protocol this was revised three times. These revisions were based upon piloting results and external reviewer's comments. Execution of Protocol started after all its details were finalized. This chapter describes the step by step execution of our SLR protocol.

3.1 Search String Application

Automated search was conducted by using search string. Search String was applied on selected database to identify the relevant studies. Advanced/Query search options available on these databases were used because advanced search provide customized option selection. In query search we can incorporate our basic relevance checking decision within the query string so that we may obtain maximum possible relevant studies.

The syntax for string on databases were not uniform, different Databases support their own pre-specified format for search string. Although help regarding the format was available on website but it took much time to understand that format and then customize the string according to that format. There was a single string for our SLR that was used to search the relevant studies. As mentioned before that syntax of the string was different for each database moreover string was parsed into sub-strings to accommodate the size/limit constraints applied by some databases. For example SpringerLink supports 100 characters only that is why string was parsed into multiple sub-strings. The syntax of strings that were applied on various databases is as:

3.1.1 ACM Search Query

(Abstract:" software architecture" OR Abstract:"software structure" OR Abstract:"software design" OR Abstract:"system architecture" OR Abstract:"system structure" OR Abstract:"system design") and (Abstract:"case study" OR Abstract:"experience report" OR Abstract:"lesson learned" OR Abstract:"empirical" OR Abstract:"industrial" OR Abstract:"experiment")

3.1.2 ScienceDirect Search Query

abstract({Software Architecture} OR {Software structure} OR {Software design} OR {System Architecture} OR {System structure} OR {System design}) and abstract(empirical OR industrial OR experiment OR {case study} OR {experience report} OR {lesson learned})

3.1.3 IEEE Search Query

("Abstract":"software architecture" OR "Abstract":"software structure" OR "Abstract":"software design" OR "Abstract":"system architecture" OR "Abstract":"system structure" OR "Abstract":"system design") AND ("Abstract":"empirical" OR "Abstract":"case study" OR "Abstract":"experiment" OR "Abstract":"industrial" OR "Abstract":"experience report"" OR "Abstract":"lessons learned")

3.1.4 EI Compendex Search Query

({software Architecture} OR {Software structure} OR {software design} OR {system Architecture} OR {System structure} OR {system design})) AND (Empirical OR Industrial OR Experiment OR {Case study} OR {Experience Report} OR {Lessons learned})

3.1.5 SpringerLink Search Query

'ab:(("software architecture" or “software structure" or “software design" or “system architecture " or “system structure" or “system design") and ("empirical" or "industrial" or "case study" or "experiment" or “experience report” or “lesson learned”))'

Strings were applied upon title, abstract and fulltext; interestingly string returned the results with very large difference in terms of no. of studies. In case where strings were applied upon title the number of papers was quite small and there was a chance to miss certain relevant paper during search. When fulltext was considered for search then obtained numbers of studies were very large which brought many irrelevant studies. It seemed infeasible and laborious to go with this decision. That is why Abstract was considered to be the target for our search. Number of studies obtained when string was applied upon abstract was neither too large nor too small. Therefore, we got a reasonable no. of studies to apply inclusion/exclusion criteria.

3.1.6 Identified Studies

References obtained in the result of this string search was imported into reference management software, in our case we used Endnote software. All the results were imported and saved into Endnote libraries. We obtained 34% of our studies from IEEE, 31% from ACM and 24% was searched from EI Compendex. Table 5 shows the no of obtained studies.

Table 5: Total No. of Studies Obtained after String Search

No. of Studies obtained	
Resource	No. of studies
IEEE	1930
ACM	1761
Sciencedirect	259
Springerlink	308
EI Compendex	1359
Total	5617

The actual no of studies that we obtained were 5617 from all five sources. Table 5 depicts the actual no. of studies that were obtained from each database while figure 1 depicts the distribution of studies among the various search databases.

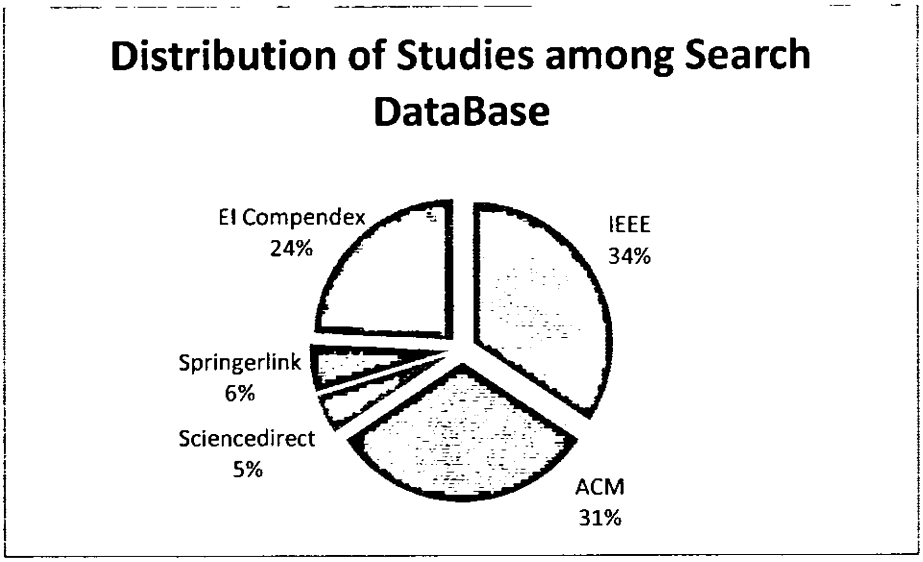


Figure 1: Studies obtained after database search

3.2 Studies Inclusion/Exclusion Process

A total of 5617 studies were obtained from search. There were 930 duplicates identified and discarded from the library. Then study screening was performed. We planned to screen the studies upon title, abstract and full text but actually screening upon title was not easy to decide relevance and irrelevance upon title of the study. Titles were not that explanatory; that is why Study Screening was performed in two phases. In first phase studies were screened upon title and abstract (non full text) and then upon full text. The detail of the screening process is described as:

3.2.1 Title and Abstract Screening

In this phase obtained studies were screened upon titles and abstracts. Each study title and abstract was assessed to find out the relevance of the study according to the focus of this SLR. If it was found irrelevant then the study was excluded from the main library. The studies which were found relevant were copied to another library. This process was repeated for all the studies. There were some papers that we were unable to assess upon non full text screening for relevance or irrelevance; those papers were kept for next level of screening.

3.2.2 Fulltext Screening

A total of 449 studies were selected for fulltext screening. Relevant studies were copied to library containing included studies otherwise it was excluded. In this phase thee fulltext reading of the study was performed to decide about the inclusion/exclusion of the study. The papers which were unclear for inclusion/exclusion in this phase were decided in consensus meeting to be included or excluded. Mostly issues that we encountered during inclusion/exclusion were in "case studies". We came across many studies in which researchers used examples to elaborate their research work but they named it as a case study. As our focus was empirical work, which is why we needed to critically consider each aspect of such studies to decide that evidence provided is empirical or just example explanation. Fig. 2 depicts the whole process of screening. List of included studies is attached in appendix C. 449 studies included in this phase were screen upon multi-level inclusion and exclusion criteria described previously. Various studies were excluded during this screening process and 247 studies made up to the final inclusion. These 247 studies were copied to the new library and were assigned study Ids. Mostly included studies were from IEEE database.

Figure 2 explains the whole screening process step by step. Out of 5617 studies 1130 were excluded as duplicates. From remaining 4487 studies 449 were selected to be included after title and abstract screening. These 449 studies were included in fulltext screening and 266 studies made up to the final inclusion out of which fulltext of 19 studies was inaccessible. So the final inclusion was 247 studies. The whole process is depicted in figure 2.

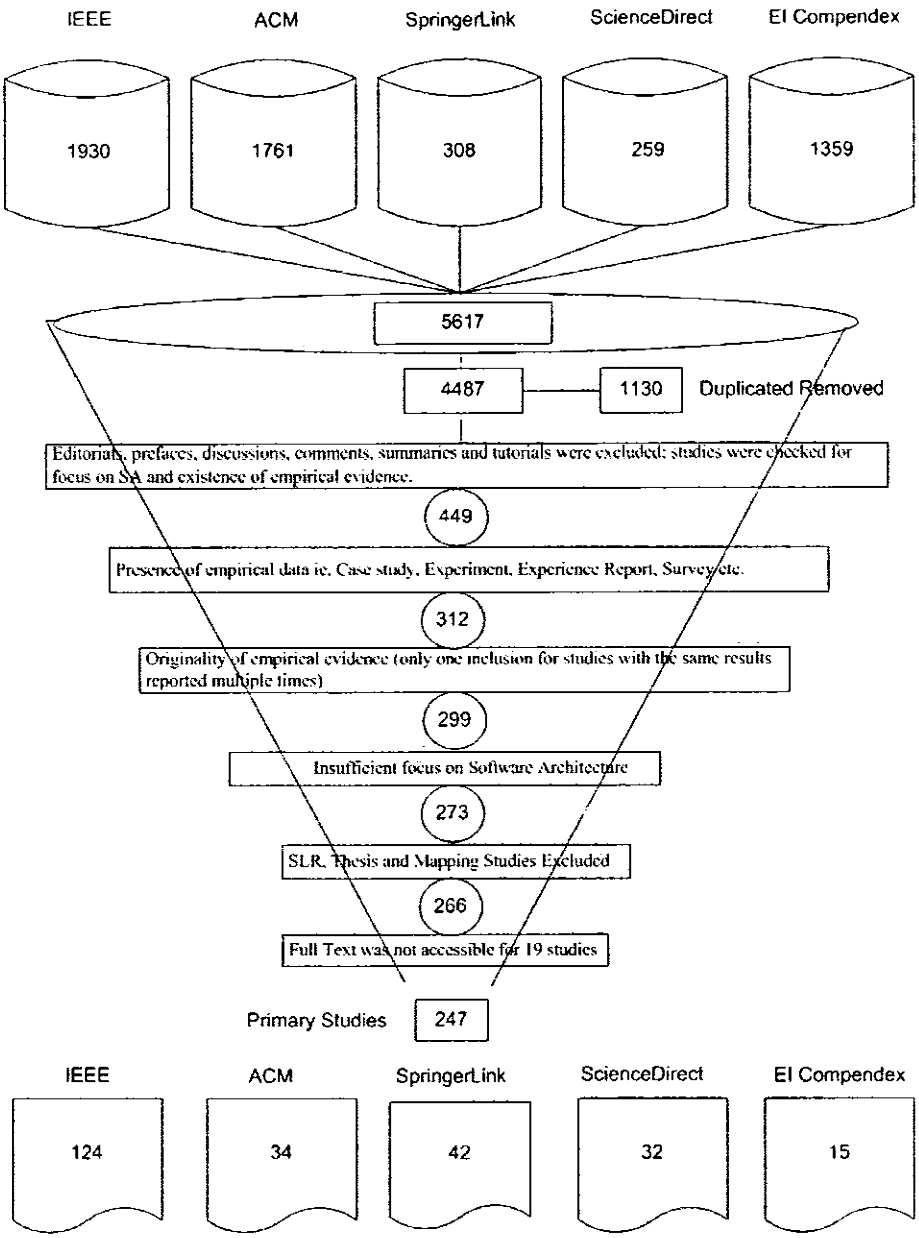

IEEE
ACM
SpringerLink
ScienceDirect
EI Compendex
1930
1761
308
259
1359
5617
4487
1130
Duplicated Removed
Editorials, prefaces, discussions, comments, summaries and tutorials were excluded; studies were checked for focus on SA and existence of empirical evidence.
449
Presence of empirical data ie. Case study, Experiment, Experience Report, Survey etc.
312
Originality of empirical evidence (only one inclusion for studies with the same results reported multiple times)
299
Insufficient focus on Software Architecture
273
SLR, Thesis and Mapping Studies Excluded
266
Full Text was not accessible for 19 studies
Primary Studies
247
IEEE
ACM
SpringerLink
ScienceDirect
EI Compendex
124
34
42
32
15

Figure 2: Study Screening Process

3.3 Study Quality Assessment

For quality assessment MS Excel sheet was used. All the questions in the quality checklist were written in MS Excel sheet. A drop down list was inserted in excel for ease of use and uniformity. 0,1 and 2 were the options of drop downs; 0 for No, 1 for partial and 2 for Yes. Study ID and study title were also written in quality checklist sheet for identification. Total scores for case study and experiment were 12, for experience report it was 16 and for survey it was 18. Total no. of quality score is not same as the number of question was different in accordance with study settings. To maintain uniformity percentage was used for overall quality assessment. Studies above 60% quality score was considered as quality studies. The whole list of studies along with their quality scores is in appendix A. First column contains study Id assigned for unique identification of the studies. Study settings, total quality score and sum of the obtained scores are displayed in the next few columns. Percentage of the quality score was calculated and displayed in the %age column. Q1 to Q4 are the generic question related to context of the study, study objectives, clear links among data and interpretations, and future implications. S1 to S5 are the questions related to survey methodology. E1 and E2 are about the quality of experimentation. C1 and C2 are questions related to case study methodology. ER1 to ER4 are the questions about experience reports.

Studies scoring more than 60% were considered as quality studies on the whole. A total of 47 studies out of 247 studies scored below 60% and rest 200 studies were quality studies. More than 60% to 75% were considered as above average. 118 studies scored more than 60% and less that 75%. Score more than 76% was considered as good quality. 62 studies were good quality studies. 19 studies obtained 100% quality scoring. Although we did not defined any pass/fail criteria.

3.4 Data Extraction

Data-Extraction was performed for 230 included primary studies. An Excel sheet was used to extract and store data. Most of the questions contained their answer options in drop down list in MS Excel sheet. Whenever we encountered a new category or option in any study, we added it in our drop down list for further studies. That is how we maintained uniformity and eliminated the chances or typing errors during extraction. For each study we just type the study id, title and its generic information. All the extraction data items to answer the research questions were selected from lists. One Extraction item was "data collection method" used in the study setting. This extraction item was not clearly mentioned in most of the studies that is why we added a term "Unclear" in the drop down list of collection methods.

The data extraction sheet was designed to extract data specifically relevant to research questions. The data about empirical settings of the study was used to answer the RQ2 and the data specific to Software architecture area was used to answer the RQ1.

100% Data extraction is done by one researcher. In 50 % of the studies researcher extracted the data and discussed it in discussion meeting that is how 50% of the data extraction validation was done. Moreover an expert of the SA area (co-supervisor) confirmed the results of the SLR. Additionally the dependency of the various results upon each other also confirms the validity of extracted data.

Summary

This chapter explains the conducting process of SLR. Initially the search strings was customized for format and was run on the data bases. The obtained references were saved in to separate libraries in reference manager software. For this purpose Endnote reference manger software was used and we obtained 5617 studies. All the libraries were combined and duplicates were removed from the main library. Then screening of the studies was started. In the first phase title and abstract screening was performed. Many studies were excluded at this stage. Then full text screening was performed and at the end we obtained 247 studies selected for data extraction. Data was extracted according to the decided data extraction schemes.. There were three type of information that was extracted from the included studies; generic information, information about SA and empirical evidence of the study. The generic information extracted about the study was study title, country, year, publication channel etc. Software architecture related information was extracted listing specific sub area of SA, type of output of the study, domain of the study etc. Finally the extraction items related to empirical evidence were extracted including type of research, study settings, types of participants etc. Extracted data items were saved in the excel sheets and used in analysis.

CHAPTER # 4 RESULTS AND ANALYSIS

4 RESULTS AND ANALYSIS

Extracted data was synthesized to answer the research questions. The objective of the first research question was to evaluate the status of the software architecture field with an empirical perspective. RQ1 for this SLR is as:

4.1 State-of-the-art in Software Architecture

We aimed to find out the state of the art of software architecture field to provide guidance for future progress in this area. Extracted data was synthesized quantitatively in terms of frequency of occurrence to depict the mature and underdeveloped areas of software architecture along with other relevant information in terms of quantity of the studies. Here is some statistics that we obtained from our study.

4.1.1 Yearly Distribution of Studies

Software Architecture as a separated discipline originated in 1990 but the initially modularity concepts and structure of system were used in early 70's. That is why we applied search from 1972 to 2010. We obtained studies older than 1995 but those studies were excluded during inclusion/exclusion phase. From the figure it is clear that software architecture research boosted after 2001. From 2001 to 2005 the number of publications per year was around 15 and this no. kept on increasing per year. From 2006 onwards the number of publications per year was around 25 studies. 2009 was the year with maximum number of publications and it was 39.

Figure 3 depicts that SA is in maturing phase and number of publication increased gradually year by year. It is important to note that we included empirical studies and excluded examples and studies lacking appropriate evidence. Even than the number of publication per year are adequate.

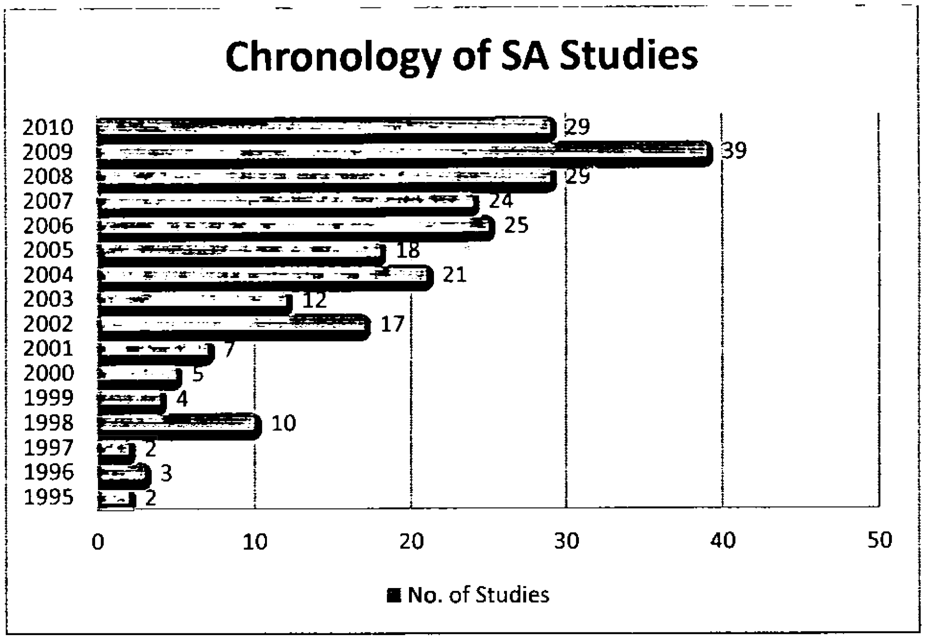


Figure 3: Yearly distribution of studies

4.1.2 Country-wise Distribution of Studies

Table 6 depicts the regions where most of the research has been done in the field of Software Architecture. A Noticeable percentage of our included studies were from USA. Carnegie Mellon University, University of Southern California and Robert Bosch Corporation are some leading institutions involved in SA research. Netherland is another country involved in SA research. In Netherlands VU University, University of Amsterdam and University of Groningen actively participated in SA Research. We got a large no of studies from Finland, Canada, Italy, Germany, UK, Australia, China, Sweden and Denmark. All these countries worked alone and in collaboration with other countries as well. The obtained results shows that most active research groups of SA research are in USA and Europe (mostly in Finland and Netherland). It is interesting to note that most of the telecom related industry is flourishing in this region of Europe (Finland and Netherland) from where we obtained most of our studies about SA.

Table 6: Country-wise List of SA Studies

Country-wise Distribution of SA Studies		
Country	**No. Of Studies**	**Study ID**
Norway	2	43, 154
Belgium	4	22, 205, 211, 219
Cyprus	1	104
Russia	1	124
Switzerland	2	132, 195
India	1	136
Maryland	1	7
UAE	1	3
Argentina	2	170, 225
Venezuela	2	1, 90
Chile	2	155, 156
Iceland	4	57.1, 57.2, 57.3, 131
Portugal	3	142, 177, 221
France	3	47, 61, 186
Japan	3	52, 76, 97
Pakistan	3	101, 233, 234
Greece	4	61, 104, 139, 166
Spain	5	105, 176, 179, 184, 213
Austria	4	9, 10, 43, 133
Ireland	7	5, 8, 15, 98, 133, 194, 196
Brazil	8	6, 27, 40, 45, 138, 144, 221, 226
Korea	8	72, 73, 74, 75, 81, 82, 114, 188
Denmark	10	18.1, 18.2, 19, 39, 57.1, 57.2, 57.3, 128, 131, 209
Sweden	12	21, 25, 44, 86, 91, 93, 96, 169, 182, 185, 194, 232
China	12	29, 30, 59, 67, 68, 80, 108, 135, 137, 148, 165, 180
Australia	12	9, 10, 11, 12,13, 14, 63, 123, 141, 178, 199, 224
UK	15	27, 40, 54, 77, 121, 138, 143, 145, 149.1, 149.2, 196, 215, 216, 221, 228
Germany	13	20, 29, 49, 52, 56, 58, 66, 152, 157, 163, 167, 187, 198
Italy	17	6, 16, 28, 33, 36, 37, 39, 62, 103, 115,125, 138, 140, 146, 176, 236, 237
Canada	23	3, 24, 39, 50, 70, 71, 87, 88, 100, 109, 113, 134, 145, 151, 155, 156, 162, 171, 204, 206, 215, 217, 223
Finland	19	33, 41, 92, 99, 102, 112, 116, 147, 159, 160, 167, 174, 190.1, 190.2, 191, 193, 201,

Country-wise Distribution of SA Studies		
Country	No. Of Studies	Study ID
		222, 235
Netherlands	30	2, 23, 34.1, 34.2, 35, 51, 55, 64, 69, 78, 79, 84, 107, 118, 122, 141, 153, 164, 167, 168, 175, 177, 202.1, 202.2, 202.3, 202.4, 210, 227, 228, 229
USA	65	4, 5, 15, 17, 21, 26, 31, 32, 34.1, 34.2, 35, 36, 38, 42, 46, 48, 53, 60, 66, 71, 72, 74, 83, 85, 89, 94, 95, 98, 106, 110, 111, 117, 118, 119, 120, 126, 127, 129, 130, 146, 150, 158, 161, 165, 167, 170, 172, 173, 181, 183, 185, 189, 192, 197, 200, 203, 207, 208, 212, 214, 218, 220, 225, 230, 231

4.1.3 Software Architecture Areas

Figure 4 depicts the area wise distribution of studies. SA Evaluation, non functional requirement

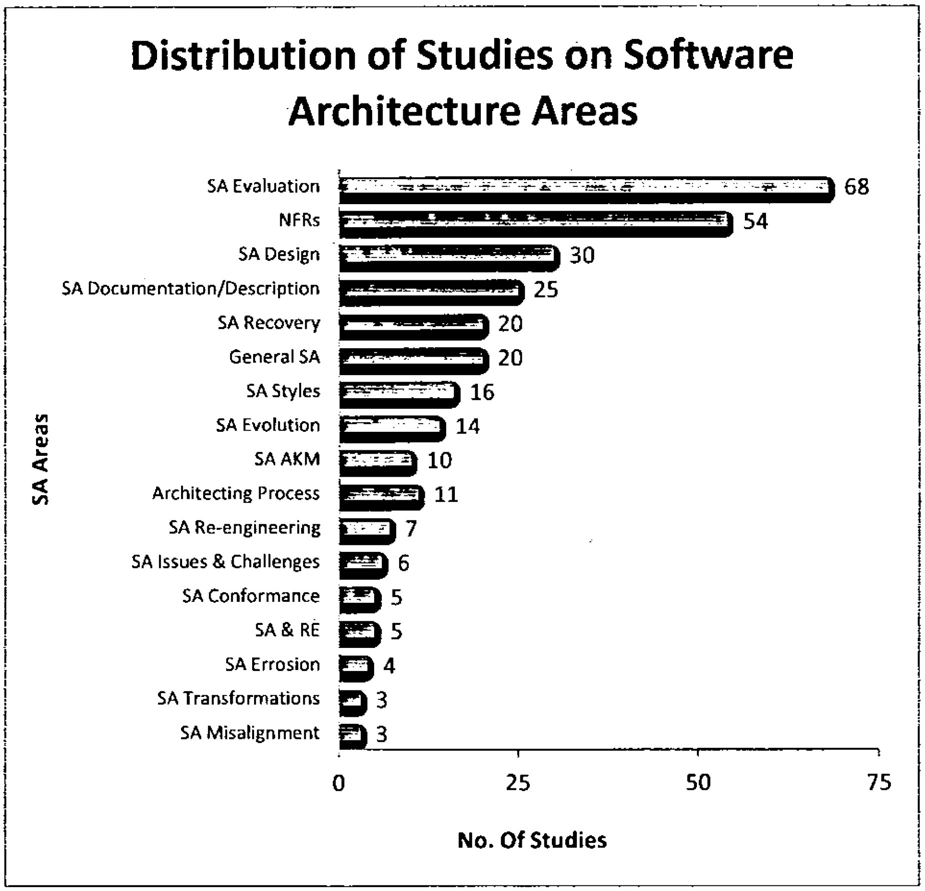


Figure 4: SA Sub Areas

and software architecture designs are the most focused areas. These three sub-areas include the core activities of software architecture that are performed on must basis when dealing with architecture. That is why these are the most researched areas of SA. Other areas with notable number of publications are SA recovery and SA description/documentation. SA description/documentation includes ADLs and viewpoints etc.

4.1.4 Major Publication Channels

Major publication channel was WICSA/ECSA from where we got maximum no. of studies. 23% of our included studies are published in WICSA/ECSA. The Journal of system and software and SHARK was also among our leading channels of publication. We got 11% of our studies from LNCS in which almost all the studies were obtained from different conferences except 11 studies which were obtained from QoSA. QoSA contributed 8% in our overall studies percentage. These statistics clearly depict that our top most channels were specifically related to software architecture research. Moreover ICSE was also one of the major channels of publication for our included studies. Figure 5 depicts the distribution of studies among various publication channels.

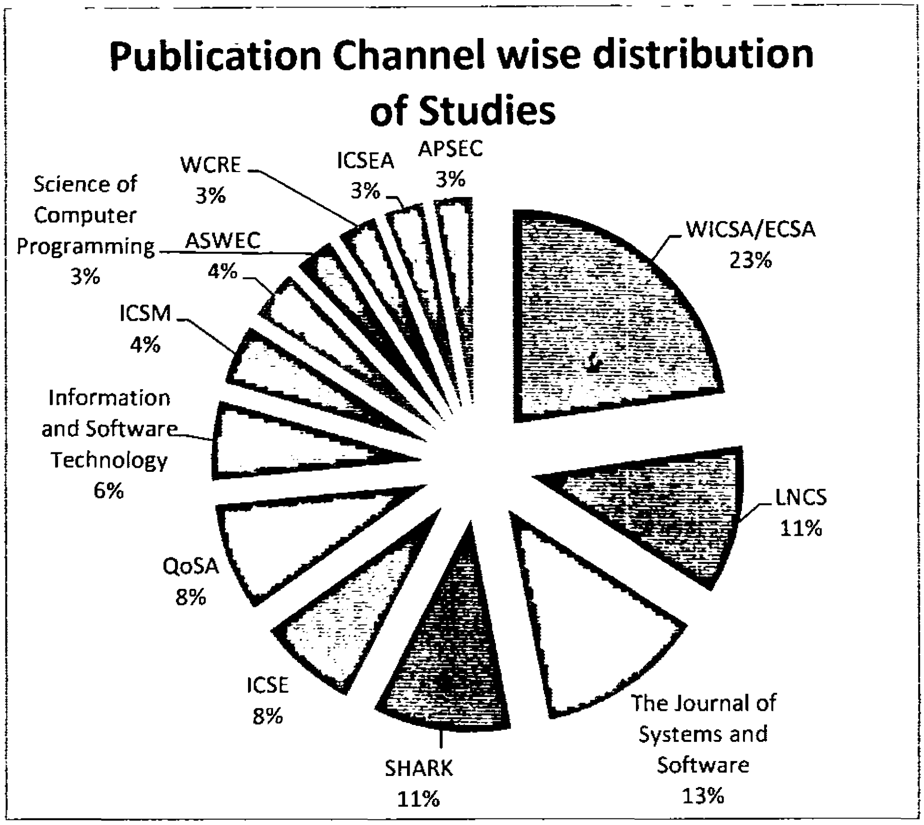


Figure 5: Major publication channels

4.1.5 Domain of Studies

Domains in which SA research has been done are mostly Telecom, Finance and web. Whenever the study was conducted in more than one domain then "multiple domains" was selected. Generic was the option used for systems/applications used to support computer and software related tasks like compilers, middleware, testing tools etc. Therefore the major part of SA research was done on these types of applications as Generic is the domain for most of the studies. It means that SA researchers mostly experimented on testing tools, middleware, groupware, operating systems. In case of other domains the major focus was telecom industry and banks. Online shopping carts, web sites and automotive industry (automated guided vehicles or other automotive industry) are other domains used in SA research.

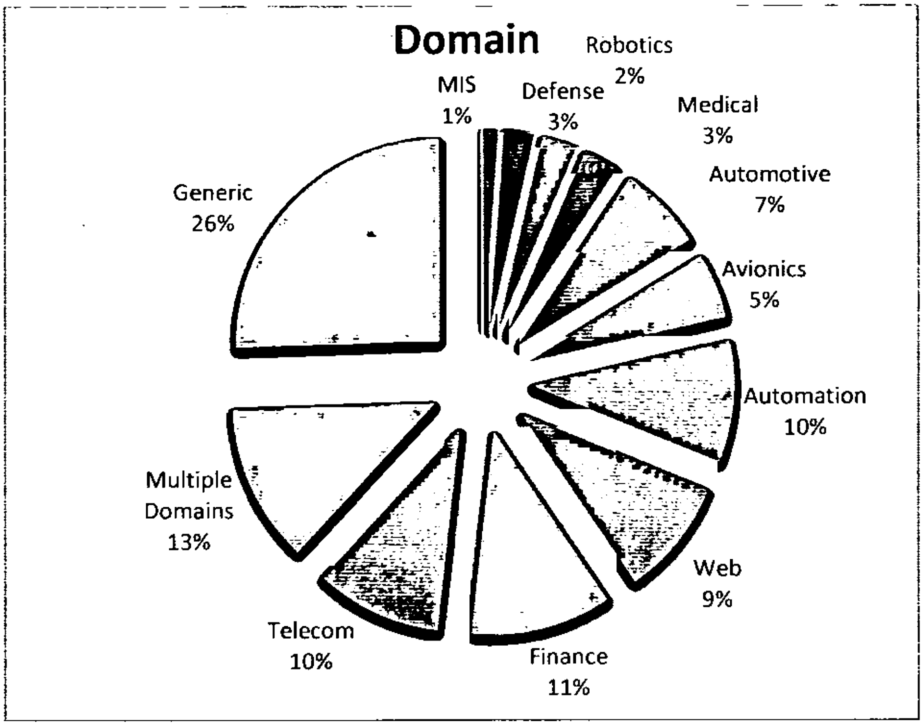


Figure 6: Domains in which SA research was done

4.1.6 Emerging Trends

Emerging trends was used to capture the usage of changing technologies in the field of software engineering and SA. Researchers and industry kept on using new technologies and mixing already existing technologies and trend for betterment. In Emerging trends Product Line, Distributed and Service oriented architecture was the trends mostly focused by the researchers. Figure 7 describes other trends as well that were addressed in the obtained studies.

We captured that whenever some new technology trend is originated in software engineering the SA community welcomed it and adopted it in their research. Many new technology trends were experimented with each other to find the solutions of unsolved problems. This adoption resulted in many new sub disciplines of SA like product line SA and SOA.

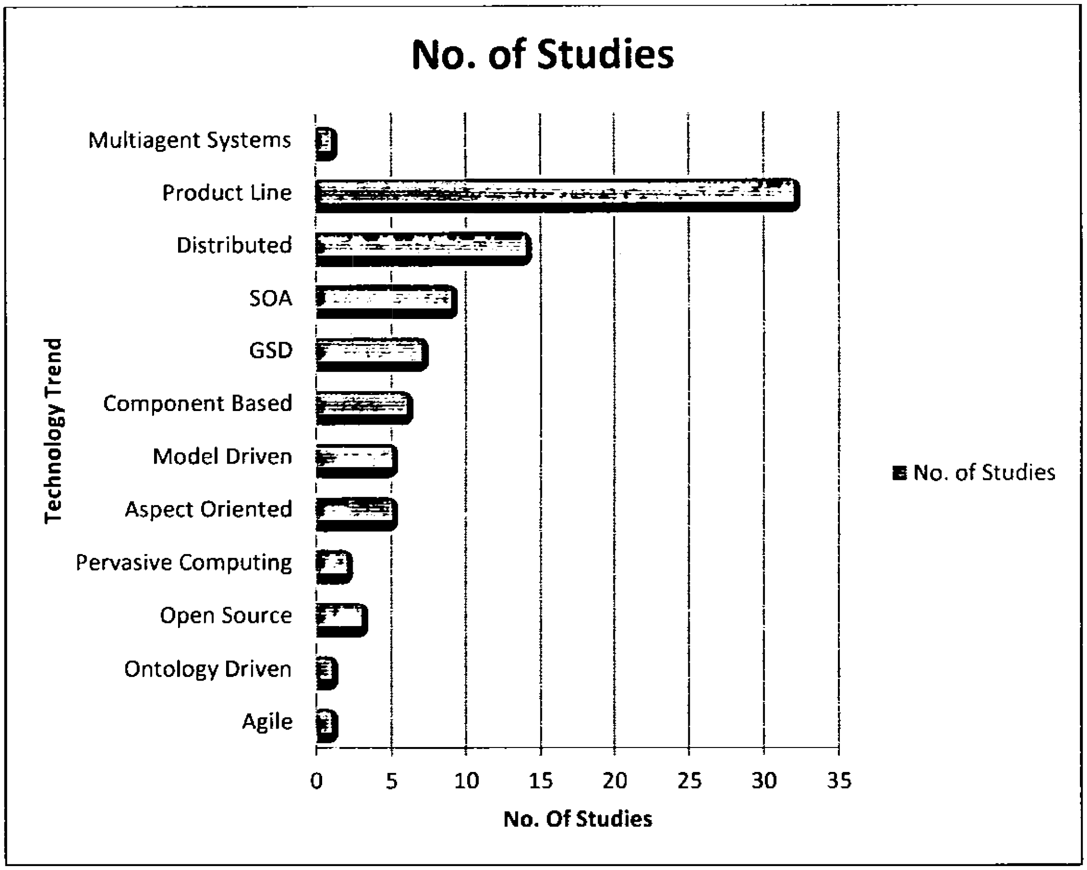


Figure 7: Emerging Trends

4.1.7 Output of Studies

In output of the study it was proposed that the output can be some new tool/ technique/process/language/framework/model or usage experience of some tool/technique/process/language/framework/model or its modification. A total of 85 studies were usage experience of techniques. In 49 studies, researchers proposed some new techniques. In 22 studies, the experiences of using tools were shared. In 18 studies, guidelines were proposed to be used in the future research. Other outputs that were mostly produced by the included studies were usage experience of model, usage experience of tool, new mode, new framework etc. Table 7 lists all the output types and frequency of the studies that produced that type of output along with the Study ID.

Table 7: Output Types Extracted From the Obtained Studies

Output of the Studies		
Research Output	No. of Studies	Study ID
New Language	4	21,67, 137, 211
Modification of Tool	2	7, 164
Modification of Framework	2	43, 72, 80
Modification of Model	2	36, 165
Modification of Language	2	4
Modification of Process	4	7, 25, 48, 164
Usage Experience of Language	4	106, 175, 198, 203
New Process	4	20, 63, 213, 222
New Tool	7	17, 22, 29, 38, 52, 93, 191
Usage Experience of Framework	8	37, 74, 104, 108, 113, 120, 121, 207
Modification of Technique	10	2, 52, 70, 75, 82, 100, 103, 135, 163, 205
Usage Experience of Process	9	3, 6, 8, 23, 77, 99, 110, 126, 169,
New Framework	12	10, 29, 32, 38, 40, 62, 73, 111, 184, 217, 223, 232
New Model	14	59, 60, 68, 71, 83, 87, 105, 136, 150, 167, 186, 192, 195, 212
Usage Experience of Model	13	44, 96, 107, 109, 115, 116, 159, 173, 176, 193, 227, 228, 234
Usage Experience of Tool	22	13, 14, 15, 26, 52, 56, 74, 89, 97, 98, 155, 156, 160, 170, 187, 218, 220, 224, 225, 230, 231, 238
Guidelines	18	33, 39, 50, 117, 151, 162, 171, 172, 185, 188, 196, 202.1, 202.2, 202.3, 202.4, 209, 215, 235
New Technique	49	22, 27, 30, 34.1, 34.2, 35, 45, 47, 51, 54, 58, 61, 64, 67, 76, 81, 91, 92, 93, 114, 125, 127, 131, 132, 137, 139, 143, 147, 148, 149.1, 149.2, 157, 158, 161, 166, 168, 174, 178, 179, 180, 181, 183, 189, 191, 204, 206, 208, 214, 226

Output of the Studies		
Research Output	No. of Studies	Study ID
Usage Experience of Technique	85	1, 5, 9, 11, 12, 16, 18.1, 18.2, 19, 24, 26, 28, 31, 41, 42, 46, 51, 51, 52, 53, 55, 56, 57.1, 57.2, 57.3, 58, 64, 65, 66, 69, 78, 79, 84, 85, 86, 88, 90, 94, 95, 101, 102, 112, 118; 119, 122, 123, 124, 128, 129, 130, 133, 134, 138, 140, 141, 142, 144, 145, 146, 149.1, 149.2, 152, 153, 154, 155, 156, 158, 160, 161, 177, 182, 190.1, 190.2, 194, 197, 199, 200, 201, 210, 216, 219, 221, 229, 233, 236, 237

4.2 Strength of Empirical Evidence

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 are same but extracted data is different. To answer this question data is 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.

4.2.1 Study participants

Data was captured from each selected study about study participants to know that the study was conducted in industry or academia. There were some studies which were conducted as a mutual effort in collaboration with industry and academia both.

From the extracted data we found that most of the research is going on in academia. Figure 8 depicts that 62% research has been done by the academia. Industry's contribution was 13% and both industry and academia had been involved in collaboration with each other in 25% studies. Although emerging trends adoption shows that industry is working in SA area but most probably they are not publishing and sharing their contributions with SA research community. That is why the industrial participation percentage is low as depicted by the figure 8. Practitioners need to disseminate their research results with SA research community for betterment of SA area.

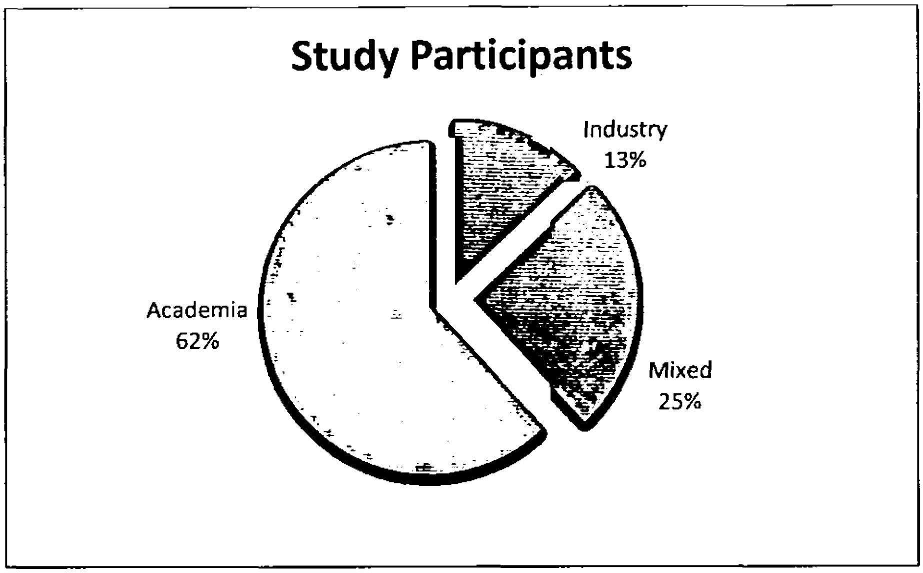


Figure 8: Study Participants

4.2.2 Study Settings

Case Study was the research methodology used for most of the studies this may be due to the reason that researchers wanted to research on SA in real world environment. Experiment and experience report were other two leading study settings for the SA research.

Figure 9 depicts that 145 case studies are reported in SA empirical research. 49 Experiments and 35 Experience reports are recorded from included primary studies. Ethnography and Action research are rather new study design that SA community has recently been encountered. We found 2 ethnographic studies and 5 action researches in SA studies. In Surveys we only included industrial surveys reporting practitioners' experiences (literature survey were excluded). As SA cannot be studied isolated in lab environment that is why we encountered most percentages in case study research. Researchers worked on SA along with its context in real life environment. Although we got experimentation as well in good percentage but that was mostly in case of architecture evaluation sub area of software architecture.

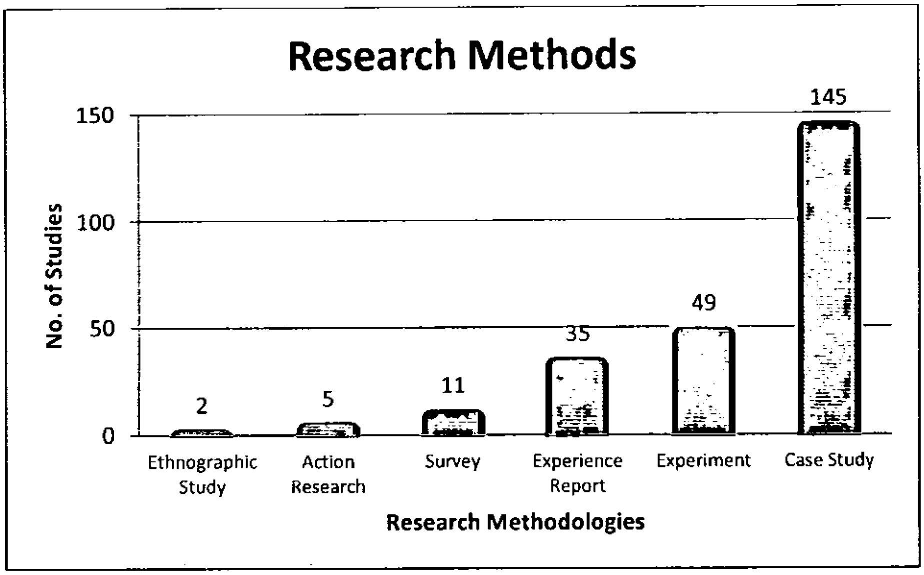


Figure 9: Study Settings

4.2.3 Data collection methods

Various data collection methods were used to capture the data. Interview, archive analysis and observations were most widely used data collection methods but surprisingly 38% studies did not clearly mention about data collection method that is why "unclear" is in large proportion in figure 10. Most of the retrieved studies did not mention any information regarding data collection methods, data storages tools and the procedures to obtain data. On the other hand some studies not only mentioned the data collection methods and tools but they also told about the sequence in which various data collection tools were used; this was the case where more than one data collection methods had been used. Data collection is linked with the strength of the evidence; it is used to know about the clarity of the data collection procedure and its repeatability.

Studies in which data collection method was not reported were mainly "case studies". It is observed that whenever data collection method is not mentioned explicitly in a study it also lacks other contextual and methodological details of the case study. Such lapses in "study reporting" resulted in less quality scores. As quality scoring is based upon contextual details and sound methodological details.

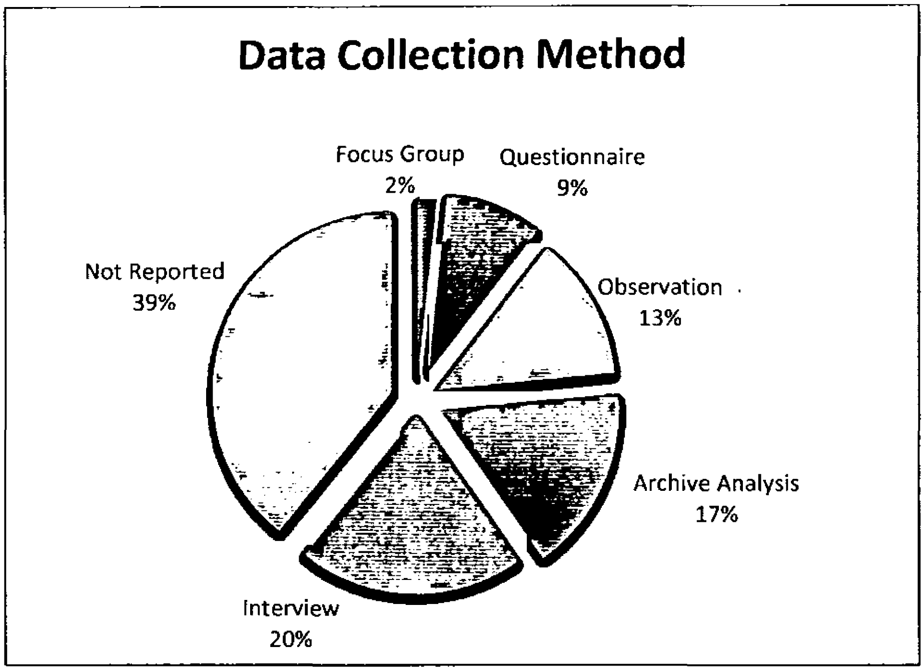


Figure 10: Data collection methods

4.2.4 Combined Data Collection Methods

Data collection methods were used as a standalone single method in some studies while many researchers used two or more than two methods in combination. We analyzed these standalone 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 evidence. We summarized the various data collection methods used in different study settings like data collection methods used in case study, data collection methods used in experiment and data collection methods used in survey etc.

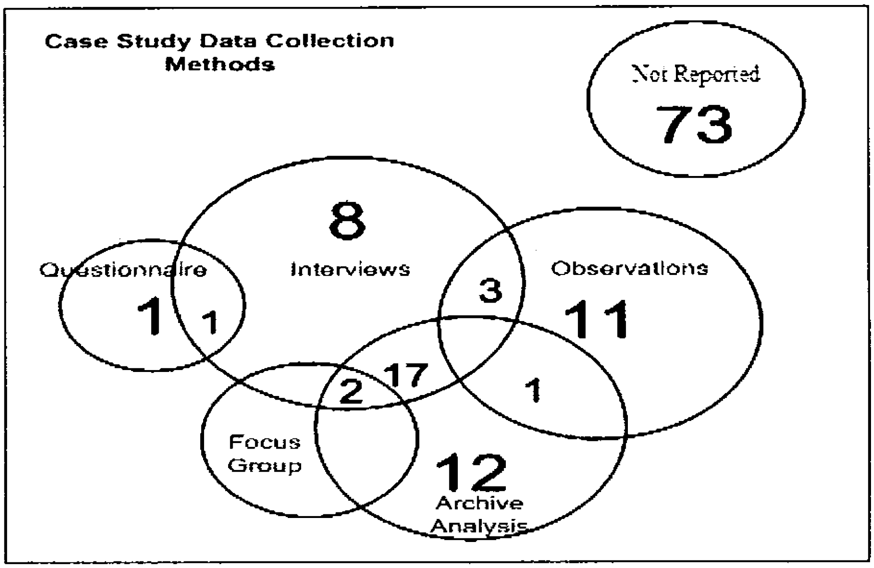


Figure 11: Data collection methods used in case studies

We used Venn diagram 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 tells that in 8 studies interviews were conducted as standalone data collection method. In three studies interviews were combined with observations. In 17 studies interview and archive analysis were used collectively to collect data. One study used questionnaire and interview both and one study only used questionnaire in the case study. 12 studies used archive analysis as a single method for data collection. In 2 studies focus group, interviews and archive analysis were the three methods mutually used to collect the data during the case study. 11 case studies used observation as main data collection method. Many studies didn't explicitly mention their data collection method that is why we used unclear as term to depict such studies where data collection method was not clear. 73 studies didn't clearly write about their data collection methods.

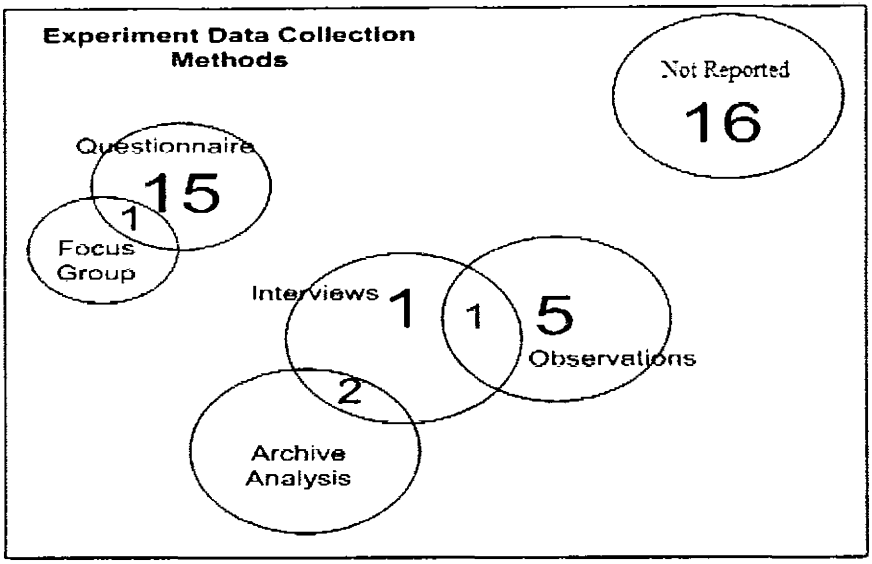


Figure 12: Data collection methods used in experiments

In case of experiment the data collection methods used by various researchers are shown in Venn diagram of Figure 12. It is clear from the figure that 1 study conducted interviews as the only data collection method. In 2 studies interviews were combined with archive analysis. In another study interview and observation was used collectively. Five studies used observation as a single method for data collection. 15 studies used questionnaire as data collection method in their experiment. In one study questionnaire was combined with focus group. Studies that didn't explicitly mention their data collection method were 16 in number.

For experience reports various data collection methods were used as depicted in the Venn diagram in Figure 13. In 8 studies interviews were conducted as the only data collection method used for their experience report. In 5 studies interviews were combined with archive analysis. One study used interview and observation collectively. Four studies used observation as a single method for data collection and in two studies archive analysis were used as a single data collection method. Studies with unclear data collection method were 10.

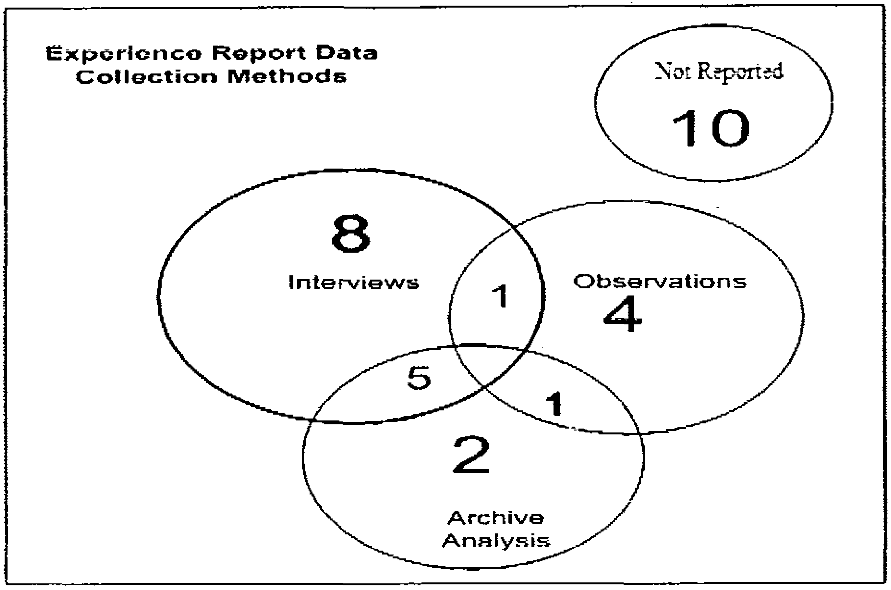


Figure 13: Data collection methods used in experience reports

In surveys 6 studies used questionnaire as the only data collection methods. In one study questionnaire was combined with interviews and in another study it was combined with archive analysis. Two studies conducted interviews and one used focus group to collect data.

Figure 14 depicts data collection methods used in the study with research method as survey. Our Study did not included literature surveys. The included survey studies were all industrial surveys where development and architecture teams' experiences were shared by the authors. Clearly questionnaire is the most widely used data collection method in case of surveys. Second most used data collection method is interviews. Data collection methods other than questionnaire and interviews include archive analysis and focus group.

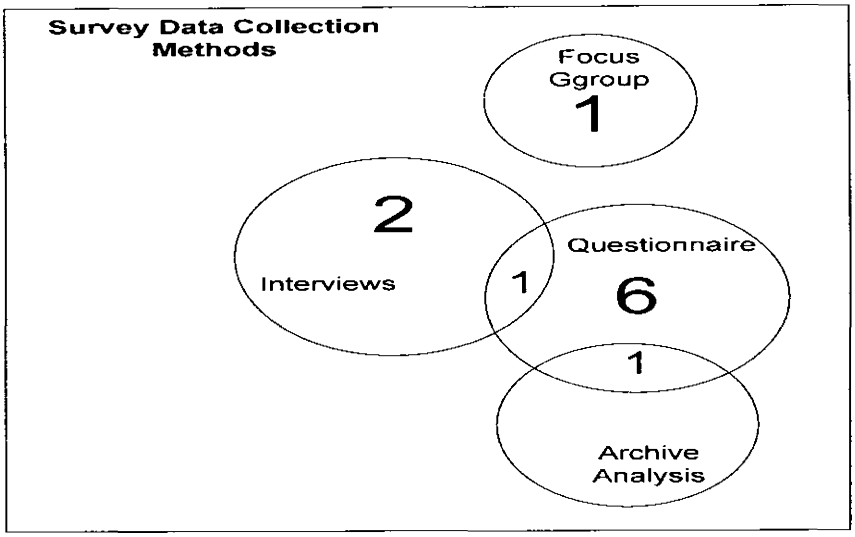


Figure 14: Data collection methods used in surveys

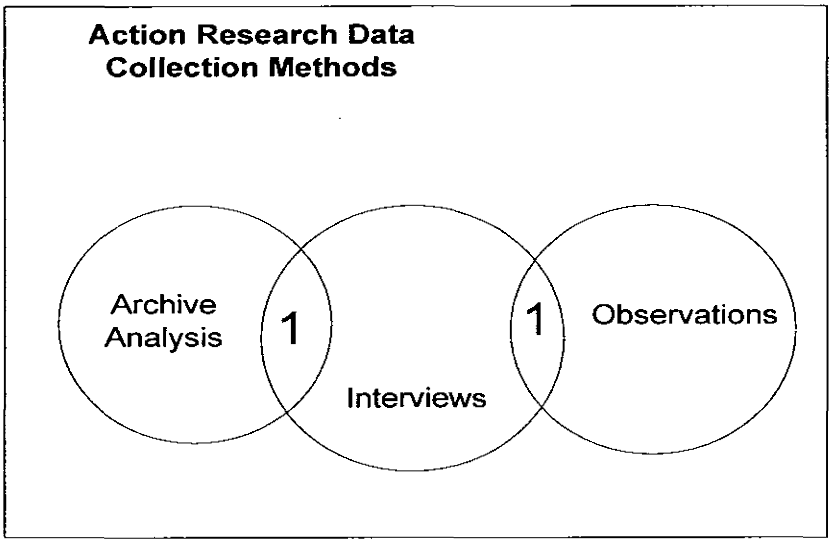


Figure 15: Data collection methods used in action research

We obtained two studies as action research, in one study the interview and archive analysis were used collectively while in the other study interview and observations were used to collect data. Figure 15 plots the data collection methods for action research. As action research is a new research method for SA community that is why numbers of obtained studies are only 2. Most probably in future there will be more studies on action research as well.

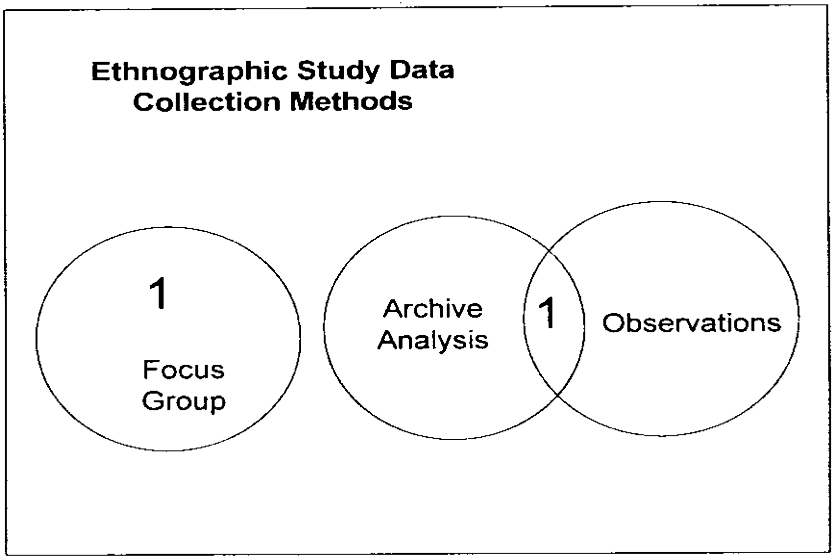


Figure 16: Data collection methods used in ethnographic study

For ethnographic study one focus group was conducted to collect data. In one study observations and archive analysis were used collectively for data collection. Figure 16 shows the data collection methods for ethnographic study.

4.2.5 Research Type

Evaluation research was the type of most of the studies i.e. 125 studies. 66 Studies were validation research and 40 proposed solutions for problems in SA area. As the focus of our research was empirical studies so we did not considered opinion papers. We did not find any philosophical paper with empirical focus. Figure 17 depicts the graph plotted for research types. Total 50 % of the reported research is evaluation research. Almost 30% of the studies were solution proposals. Experiences reported by SA community are almost 15% of the total reported research types. As the focus of our research was empirical studies so we did not considered opinion papers. Same is the case with validation research, it is usually done in lab environment and we focused empirical work that is why we did not obtained that much number of studies in validation research.

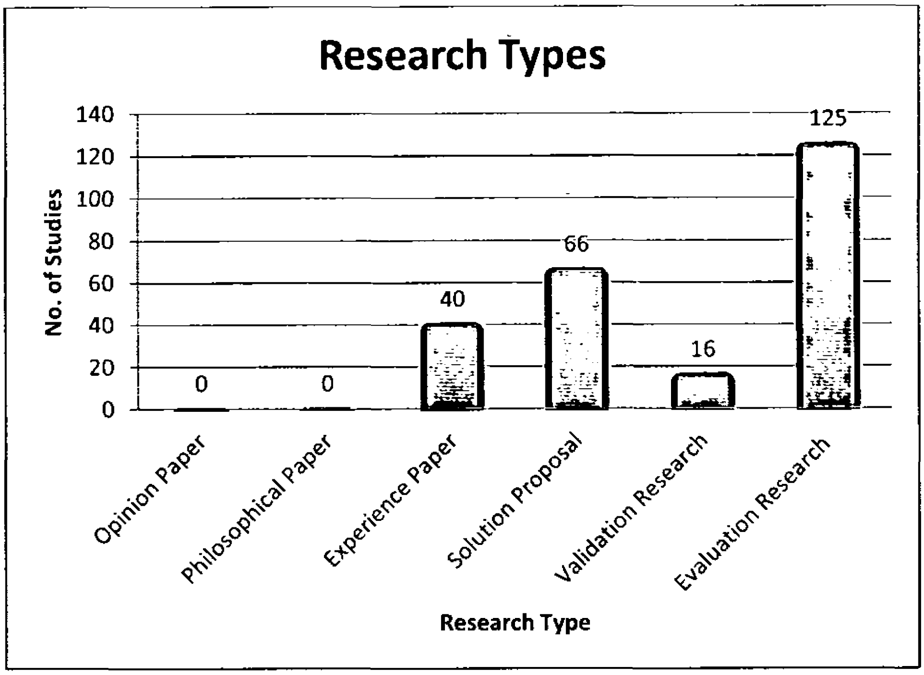


Figure 17: Research Types

*The research type classification was adopted from existing research [28][29]

4.3 Software Architecture vs. Study Settings

To identify gaps and clusters bubble charts were plotted combining different data extraction items. This combination of two different data items is important to summarize the information. In Figure 18 a bubble plot is shown that is plotted by combining software architecture areas with the research methods that have been used for research in the past. Large bubbles are saturated areas of SA and generally referred as clusters. Form Figure 18 it is clear that SA evaluation, SA design, and non functional requirements are mature areas in the field of software architecture. SA evolution, recovery and SA description are developing areas. While the empty spaces show the gaps in SA Erosion, SA transformation, SA reengineering, AKM, and SA conformance are the area that need to be focused in future.

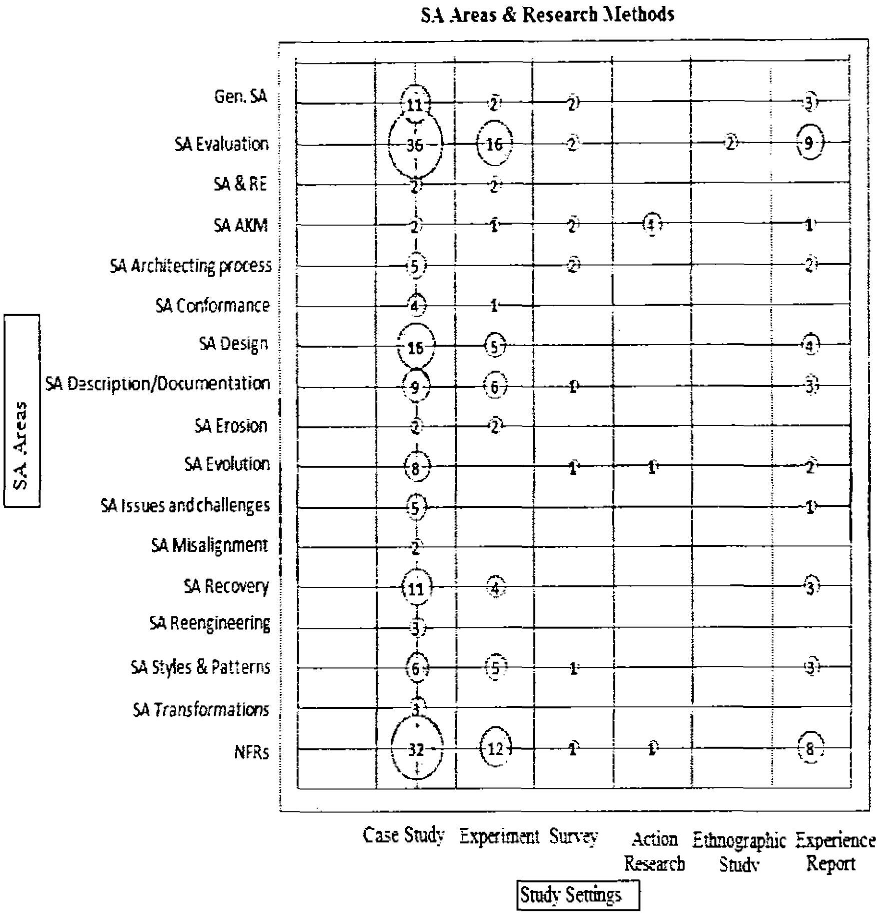


Figure 18: SA Areas and Research Methodology bubble plot

Figure 18 also depicts that case study and experiment are the research methods mostly used in SA research. Experience report is the other methodology that has been used to some extent in SA research. Survey, ethnographic study and action research are the methodologies that were seldom used in the past. The empty area in the chart clearly points towards the much needed areas regarding research. Figure 18 shows clusters for case study and experiment study settings while there are gaps for ethnographic studies and action research. Action research and ethnographic study are the methods that are gaining interest from the researchers now. Hopefully in the next few years the number of publication in these research methods will also increase. Core areas of SA are rather mature as compare to new areas but overall SA is in maturing phase.

4.4 Participants vs. Study Settings

Figure 19 combines study participants with study settings in a bubble plot. This tells us that academia is mostly involved in the SA research. Industry need to initiate some research activities to fill the gap. Industry is not publishing and researching their work. Moreover case study and experiment are the methods that are being used by academia. Survey, action research and ethnographic studies need to be conducted in industry and in mixed settings of academia and industry. The interesting finding from this plot is that there is a uniformity pattern seen in the experience reports. Almost equal number of reports have been published by academia, industry and mixed setup.

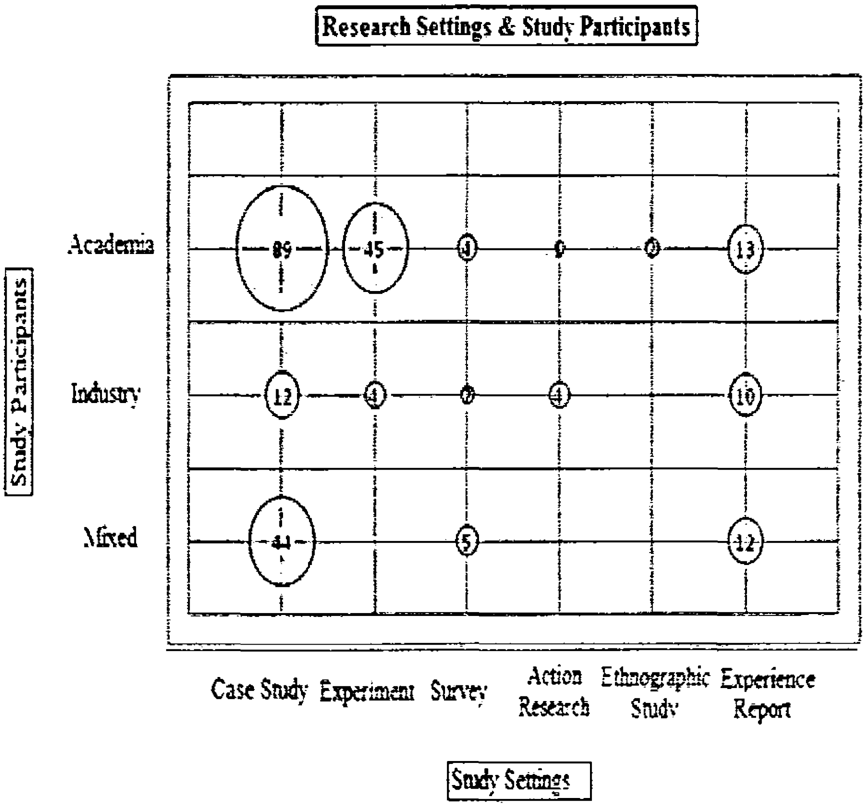


Figure 19: Study Participants and Research methodology bubble plot

4.5 Participants vs. Research Types

Figure 20 depicts the participants with research types. Evaluation is the research type that has been researched very often by academia. Solution proposal is another research type that was researched by academia. In mixed setup significant no of studies has been published in evaluation research. Frequencies of occurrence of all other research types were found almost equal except validation research. We obtained 15 studies from academia in validation research, Industry and mixed participants were not involved in validation research.

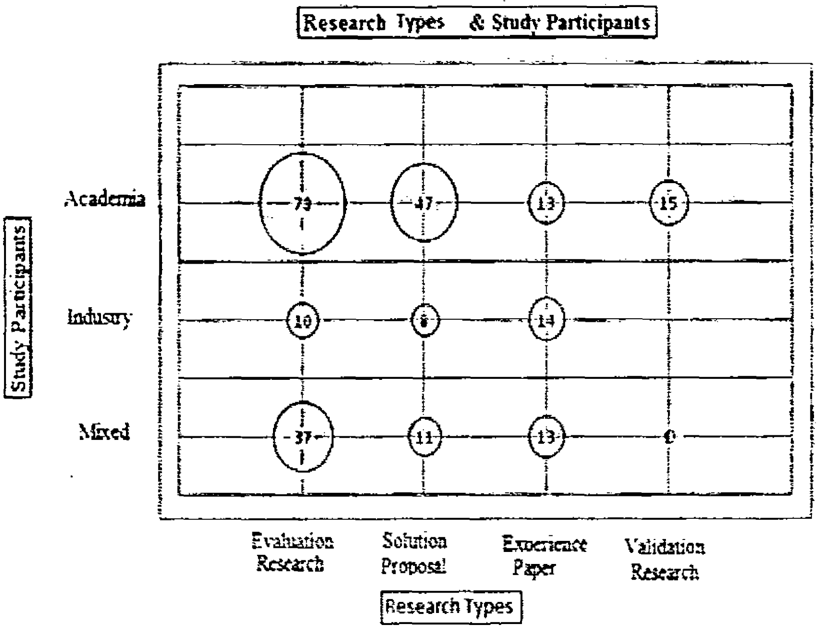


Figure 20: Study Participants and Research Type bubble plot

4.6 Timelines of Software Architecture

From extracted data we come up with timelines of SA area which is given in figure 21. It shows that SA Design and General Software Architecture were the areas with empirical work published in 1995. In 1996 SA Evaluation and SA recovery were reported. Software Architecting Process and non- functional requirement related work was focused in 1997. SA styles and patterns were reported in 1998. SA documentation and SA description related work was published in 1999. SA Evolution and SA re-engineering was reported in 2001, SA related Issues and Challenges and SA along with RE in 2002, SA erosion and SA Transformation in 2003, misalignment in 2005, SA conformance in 2006 and SA knowledge management in 2007. From the timelines it is evident that core architecting activities was focused in the initial years and then the supportive activities of SA were focused later on. AKM is the most recent area of the SA and it has gained much interest since 2007. A number of studies have already been published on AKM.

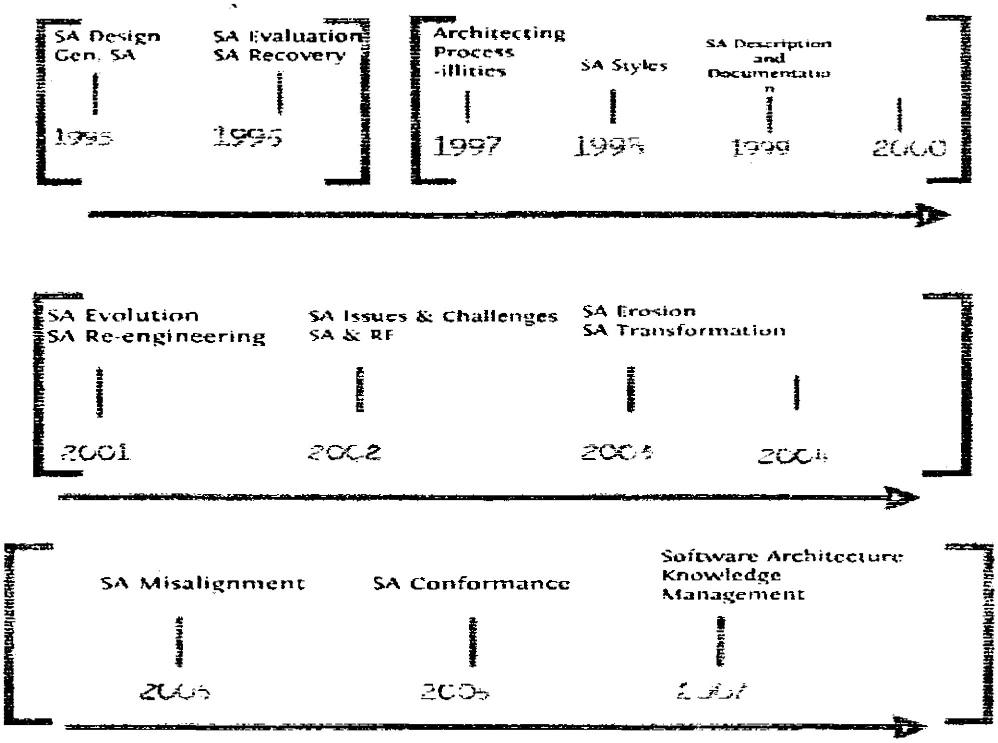


Figure 21: Timelines of Software Architecture Field

4.7 Discussion

This section briefly discusses the results obtained for RQ1 and RQ2 in terms of SA state-of-the-art and strength of empirical evidence.

4.7.1 Status of Software Architecture (RQ1)

Research in SA started in 1972 with the concept of modularity. At that time systems were not large and complex so there was a vague boundary between design and architecture. SA as a separate discipline was recognized in 1990 and advancements in SA research started in mid 90s. Software Architecture related empirical literature included in our SLR also confirms that this field got a boost in mid of 90s. Most Active research communities and institutions working in SA are mostly in USA and Europe. USA, Netherlands and Finland are the regions of eminent research institutions of Software Architecture. Carnegie Mellon University, University of Southern California and Robert Bosch Corporation, VU University, University of Amsterdam and University of Groningen are leading institutions involved in SA research. SHARK, ECSA/WICSA, Journal of systems and software and QoSA are the more related publication channels for reporting SA research. We obtain most of the studies form these publication channels. Most of the studies in our SLR are from these publication channels. Overall SA is in maturing phase. In software architecture, SA Evaluation, non functional Requirement related SA research, SA Design and SA description and documentation are mature and consistently researched areas of SA. It is interesting to note that all these SA sub areas actually constitute the core architecting process. So we can say that SA researchers are focusing on developing techniques at core architecting process. We categorized the output of the primary studies in some common and most widely used output types. We encountered the usage experience of techniques as mostly reported output and new technique is the second most widely used. It indicates that researchers are proposing new techniques; then the whole community uses it in their context and shares their experience about that specific technique. More over SA community is actively implying new technologies and emerging trends within computing discipline. We have obtained large no of studies for emerging trends like product line software architecture, distributed SA and SOA. Model driven SA and aspect oriented SA have attained much interest from the researchers. SA in Agile development, Pervasive Computing, SA in multiagent systems and ontology driven SA are recent emerging trends; SA community has just started working on these Trends.

4.7.2 Empirical Work of Software Architecture (RQ2)

The research method reported in most of the SA studies is case study research. Almost 60% of the studies are case studies in our included our primary studies. That may be because SA needs to be observed in its real environment. The major problem that we encountered was poorly reported case studies. Most of the time data extraction from a case study research was problematic because authors do not explicitly report methodological descriptions. Many authors did not reported their data collection methods, case study context and even details of organization and participants roles. The second most attempted research method is experimentation with 20% experiments in our primary studies. Researchers have also reported their experiences (15%) in the form of reports. Action research and Ethnographic studies are relatively new study settings that need to be attempted by SA community. Most of the research in the field of SA is from Academic institutions, academia is playing important part in SA research and advancement. In most SA researches the tools used to collect the evidence is not reported; although, when it is reported it showed that researchers used stand alone as well as multiple tools for data collection. Interview, observation and archive analysis are the most widely used evidence collection methods. The research types that we encountered are mostly evaluation research and solution proposal. It means that SA community is engaged in finding solution of the problems encountered during software architecting. Then they researched on the same solution by implying it one way or the other to establish some best solutions.

Summary

This chapter contains the analysis phase of SLR. Data extracted from included studies was synthesized into quantitative summaries, bar graphs and bubble plots to convert data into useful information about SA literature. Analysis was performed upon extraction items according to the research questions. In first section the RQ1 related data items were analyzed and then RQ2 related data was analyzed. To draw more useful conclusion the data items of RQ1 and RQ2 were combined at the end.

The clusters obtained in the results shows that mature areas of SA are SA design, non-functional requirements, SA styles etc. The gap in the results shows that SA erosion, SA transformation and requirement conformance are the areas that lack empirical work. USA, Netherland, Finland and Canada are origins were SA research has been done in the past. WICSA/ECSA, SHARK, ICSE and QoSA are leading SA conferences. Most of the research has been done in academia.

CHAPTER # 5 CONCLUSION

5 CONCLUSION

To find out state- of- art in the field of software architecture a systematic literature review was conducted. The timelines of SLR were from 1972 to 2010 and focus was empirical evidence. Studies were searched in IEEE, ACM, ScienceDirect, Springerlink and EI Compendex 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 study are as follows

5.1 Principal Findings

The results were obtained to answer the research questions. Quantitative summaries and frequency tables are presented in results chapter to answer the RQ1. Research types, participants, source of evidence and strength of evidence were discussed to answer the RQ2. The major findings of our study are as:

- SA evaluation, SA Design and non-functional requirements are the areas that are most researched in the past. The most researched areas are still active areas of the SA research. It is evident from last five years' statistics that SA evaluation, SA design and research related to non-functional requirements are still active and researchers are doing more work in these sub areas of SA as compare to other areas. AKM, a relatively new sub area of SA, is also an active area of SA research.
- SA erosion, SA transformations, SA reengineering are the areas where there is a scope to conduct further studies. Areas that came into being as a result of problems and issues in SA

are relatively new areas and there is a need to work more in these sub areas so that the SA community can cope with challenges effectively. These areas are SA erosion, SA conformance, SA transformation and SA reengineering. Interestingly all these SA sub areas are more dependent on requirements, size and complexity of the software.

- Model driven, aspect oriented and service oriented SA are gaining much interest from industry and researchers. Emerging trends are increasingly experimented and researched by SA community. Researchers usually combined already existing techniques with new trends and technologies in various sub areas of SA. Some new trends resulted into new sub areas of SA like SOA (service oriented software architecture), MDA (model driven architecture) etc.
- A large number of new techniques and new tools have been proposed. The second largest output was usage experience of these tools and techniques. Many new frameworks and new process have also been proposed by the research. New techniques and tools have been proposed to solve the problems and to evaluate the issues. Usage experience of the tools and techniques was published to share the usage experience with SA community.
- USA, Netherland, Canada, Germany, Italy, Sweden and Finland are the leading countries with major contributions in SA. USA and Netherland are the countries where most of the SA research has done so far, other contributing countries are Canada, Germany, Italy, Sweden and Finland. Most of the SA research was carried out by academia. Leading institutions where SA research has been done are: Carnegie Mellon University, University of Southern California, Robert Bosch Corporation, VU University, University of Amsterdam and University of Groningen. Moreover WICSA/ECSA, SHARK, ICSE, QoSA were among the leading channels of publications for SA research.

- Academia is playing very important part in SA research. Industry need to publish more works so that researcher can benefit from their work. Most of the published literature was academic although the industry also contributed in literature but relatively small part of the published work was from industry. Industry and academia should go hand in hand but there is a gap in industrial and academic research. There is a lot of research done in the labs that need to be implemented and used in real software development environment. Moreover SA industry is also working on SA research but they usually don't publish their work due to various reasons. Industry should share their experiences with SA community so that others can benefit from their results.
- Evaluation research and solution proposal are the research types that researchers used in SA research.
- Researchers usually do not strictly follow the quality criteria for empirical research. The studies obtained for this SLR were all empirical studies and the major part of these studies were case studies. Most of these case studies could not make up to the inclusion as many researchers have elaborated case scenarios and examples implications to illustrate their findings but they referred such type of studies as case studies. The included case studies lack in quality as context of the study, study setup, data retrieval and recording methods and tools were not listed.
- Publication channels (search Database) do not support reference manger tools and their syntax for advance/string search is not uniform that creates delays and problem in searching relevant studies. Publication channels that we used for our SLR were IEEE, ScienceDirect, SpringerLink, ACM and EI Compendex. All databases have different syntax and rules for string/advance search. There is no such mechanism that automatically generates the string in

appropriate format from major search terms for each database. It is time taking and tedious to read the instruction manual for each database and convert the string in appropriate syntax to run string search. Moreover the syntax of the database is not consistent as well it is updated continuously. For security reasons many database don't support reference manager software which also add in difficulties of importing references.

5.2 Implications

Software architecture is an important area in software engineering because architecture is the skeleton around which the whole development is done. Many other areas of software engineering depend upon SA. That's why its importance increases. Empirical software engineering is rapidly gaining popularity now a day. In SA area, a large no of empirical work is already done but this work is not aggregated. We attempted to aggregate this empirical evidence on one platform 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 as it covers the whole SA area including almost all sub areas of SA research. This study depicts the overall picture of empirical work done in SA so far.

5.3 Validity Threats

We face major challenges while designing the query. In order to cover the all the literature we identified and used all possible terms that were used for software architecture. The terms like "system", "design" and "structure" were used to refer SA before 1990. To ensure maximum literature coverage we did use these terms and this resulted in large no. of studies from other engineering disciplines. We did laborious work in study inclusion/exclusion phase to screen out those studies but we did not omit those terms form our query string. This decision was taken specifically to minimize the chance of not finding any relevant paper. Moreover most databases have limitations on no. of search term used and no. of character so to accommodate such type of limitations we broke up our query string into sub strings.

We performed query string pilot testing and data extraction pilot testing to enhance our confidence on string and data extraction form. Both the string and the data extraction form were modified based on the pilot testing results. Data extraction form was designed during the protocol development then this form was modified and refined according to the pilot results and reviewer comments.

5.4 Future Directions

Many SLRs already exist in the sub areas of SA e.g. in SA Evaluation, Evolvability etc. The clusters obtained in the results shows that SLRs and mapping studies can be conducted in such mature areas of SA like SA design, non-functional requirements, SA styles etc. The gap in the results shows that erosion, SA transformation and requirement conformance are the areas that need empirical work to be done in the future. Moreover there exist various quality checklists for the quality assessment of the studies in SLRs that need to be standardized. Action research and ethnographic studies need to be conducted in Software Architecture areas.

REFERENCES

[1] C. Gacek, A. Abd-Allah, B. Clark, and B. Boehm, "On the definition of software system architecture," in *Proceedings of the First International Workshop on Architectures for Software Systems*, 1995, pp. 85–94.

[2] D. Garlan and M. Shaw, "An introduction to software architecture," *Advances in software engineering and knowledge engineering*, vol. 1, pp. 1–40, 1993.

[3] D. E. Perry and A. L. Wolf, "Foundations for the study of software architecture," *ACM SIGSOFT Software Engineering Notes*, vol. 17, no. 4, pp. 40–52, 1992.

[4] D. Soni, R. L. Nord, and C. Hofmeister, "Software architecture in industrial applications," in *Software Engineering, 1995. ICSE 1995. 17th International Conference on*, 1995, pp. 196–196.

[5] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice.* Addison-Wesley Longman Publishing Co., Inc., 2003.

[6] E. W. Dijkstra, "The structure of the'THE'-multiprogramming system," *Communications of the ACM*, vol. 11, no. 5, pp. 341–346, 1968.

[7] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, vol. 15, no. 12, pp. 1053–1058, 1972.

[8] D. L. Parnas, "Information distribution aspects of design methodology," *Methods*, vol. 4, no. 5, pp. 6–7, 1971.

[9] P. Kruchten, H. Obbink, and J. Stafford, "The past, present, and future for software architecture," *Software, IEEE*, vol. 23, no. 2, pp. 22–30, 2006.

[10] D. Garlan, "Research directions in software architecture," *ACM Computing Surveys (CSUR)*, vol. 27, no. 2, pp. 257–261, 1995.

[11] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *Software, IEEE*, vol. 22, no. 1, pp. 58–65, 2005.

[12] B. J. Oates, "Widening the scope of evidence gathering in software engineering," in *Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on*, 2003, pp. 59–64.

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

[14] D. E. Perry, "State of the Art: Software Architecture," in *International Conference on Software Engineering*, 1997, vol. 19, pp. 590–591.

[15] H. P. Breivold, I. Crnkovic, and M. Larsson, "A systematic review of software architecture evolution research," *Information and Software Technology*, 2011.

[16] Y. Chen, X. Li, L. Yi, D. Liu, L. Tang, and H. Yang, "A ten-year survey of software architecture," in *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*, 2010, pp. 729–733.

[17] M. Shaw, "The coming-of-age of software architecture research," in *Proceedings of the 23rd international conference on Software engineering*, 2001, p. 656.

[18] M. Shaw and P. Clements, "The golden age of software architecture," *Software, IEEE*, vol. 23, no. 2, pp. 31–39, 2006.

[19] D. Falessi, M. A. Babar, G. Cantone, and P. Kruchten, "Applying empirical software engineering to software architecture: challenges and lessons learned," *Empirical Software Engineering*, vol. 15, no. 3, pp. 250–276, 2010.

[20] E. Y. Nakagawa, D. Feitosa, and K. R. Felizardo, "Using systematic mapping to explore software architecture knowledge," in *Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge*, 2010, pp. 29–36.

[21] R. Farenhorst and R. C. de Boer, "Knowledge management in software architecture: State of the art," *Software Architecture Knowledge Management*, vol. 1, p. 21, 2009.

[22] B. J. Williams and J. C. Carver, "Characterizing software architecture changes: A systematic review," *Information and Software Technology*, vol. 52, no. 1, pp. 31–51, 2010.

[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. Sjøberg, O. P. Brereton, D. Budgen, T. Dyb\aa, M. Höst, 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. 479–481.

[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. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9, pp. 833–859, 2008.

[28] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requirements Engineering*, vol. 11, no. 1, pp. 102–107, 2006.

[29] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering*, 2008, pp. 71–80.

[30] N. Qureshi, N. Ikram, M. Bano, and M. Usman, "Empirical Evidence in Software Architecture: A Systematic Literature Review Protocol," in *ICSEA 2011, The Sixth International Conference on Software Engineering Advances*, 2011, pp. 534–538.

APPENDICES

APPENDIX A: QUALITY SCOREOF STUDIES

Quality Scores																		
		Generic				Survey					Experi		Case-		Experience Report			
ID	%	Q	Q	Q	Q	S	S	S	S	S	E	E	C	C	ER	ER	ER	ER
1	75	2	1	2	2								0	2				
3	75	2	2	2	0	2	2	0	N	2								
4	75	2	2	1	2								0	2				
5	81	2	2	1	2	2	2	0	N	2								
6	75	2	1	2	2								0	2				
7	83	2	2	2	0						2	2						
8	83	2	2	2	2								0	2				
9	10	2	2	2	2						2	2						
10	10	2	2	2	2						2	2						
11	83	2	2	2	0						2	2						
12	10	2	2	2	2						2	2						
13	10	2	2	2	2						2	2						
14	10	2	2	2	2						2	2						
15	75	2	1	1	2										2	2	2	0
16	58	2	1	2	0								0	2				
17	93	2	1	2	2										2	2	2	2
18.1	58	2	1	2	0								2	0				
18.2	58	2	1	2	0								2	0				
19	58	2	1	2	0								2	0				
20	75	2	1	2	2								0	2				
21	75	2	2	2	0										2	2	2	0
22	75	2	1	2	2								0	2				
23	83	2	2	2	2								0	2				
24	58	2	1	2	0								0	2				
25	75	2	1	2	2								0	2				
26	58	2	1	2	0								0	2				
27	58	2	1	2	0								0	2				
28	58	2	1	2	0								0	2				
29	66	2	0	2	2								0	2				
30	66	2	1	1	2						0	2						
31	75	2	1	2	2								0	2				
32	66	2	2	2	0								0	2				
33	83	2	2	2	2								0	2				
34.1	75	2	1	2	0								2	2				
34.2	75	2	1	2	0								2	2				
35	75	2	1	2	2								0	2				

Quality Scores																		
36	75	2	1	2	2								0	2				
37	75	2	1	2	2								0	2				
38	93	2	1	2	2										2	2	2	2
39	81	2	1	2	0										2	2	2	2
40	50	2	0	2	0								0	2				
41	75	2	1	2	2								0	2				
42	88	2	2	2	0	2	2	2	2	2								
43	50	2	1	1	0								0	2				
44	83	2	2	2	2								0	2				
45	83	2	2	2	2								0	2				
46	83	2	2	2	0						2	2						
47	75	2	1	2	2								0	2				
48	83	2	2	2	2								0	2				
49	83	2	2	2	2								0	2				
50	83	2	2	2	0						2	2						
51	83	2	2	2	0								2	2				
52	75	2	2	2	0										2	2	2	0
53	10	2	2	2	2						2	2						
54	75	2	1	2	2								0	2				
55	83	2	2	2	2								0	2				
56	83	2	2	2	2								0	2				
57.1	10	2	2	2	2								2	2				
57.2	10	2	2	2	2								2	2				
57.3	87	2	2	2	2	2	2	0	N	2								
58	75	2	1	2	2								0	2				
59	75	2	1	2	2								0	2				
60	75	2	1	2	2								0	2				
61	75	2	1	2	2								0	2				
62	75	2	1	2	2								0	2				
63	58	2	1	2	0						0	2						
64	10	2	2	2	2						2	2						
65	58	2	1	2	0								0	2				
66	10	2	2	2	2								2	2				
67	66	2	1	1	2								0	2				
68	75	2	1	2	2								0	2				
69	10	2	2	2	2										2	2	2	2
70	75	2	1	2	2								0	2				
71	75	2	1	2	2								0	2				
72	75	2	1	2	2								0	2				
73	58	2	1	2	0						0	2						
74	87	2	2	2	2	2	2	2	N	0								
75	83	2	2	2	2								0	2				
76	58	2	1	2	0								0	2				
77	58	2	1	2	0								0	2				
78	75	2	1	2	2								0	2				
79	75	2	1	2	2								0	2				

Quality Scores																		
80	75	2	1	2	2								0	2				
81	66	2	2	2	0								0	2				
82	75	2	1	2	2								0	2				
83	75	2	1	2	2								0	2				
84	83	2	0	2	2						2	2						
85	81	2	1	2	0										2	2	2	2
86	83	2	2	2	2								0	2				
87	10	2	2	2	2								2	2				
88	75	2	1	2	2								0	2				
89	93	2	1	2	2										2	2	2	2
90	33	2	1	1	0								0	0				
91	75	2	1	2	2								0	2				
92	81	2	1	2	2										2	2	2	0
93	58	2	1	2	0								0	2				
94	75	2	1	2	2								0	2				
95	66	2	0	2	0						2	2						
96	10	2	2	2	2						2	2						
97	66	2	1	1	2								0	2				
98	75	2	1	2	2								0	2				
99	68	2	1	2	0										2	2	2	0
100	58	2	1	2	0								0	2				
101	75	2	1	2	0						2	2						
102	66	2	0	2	2						0	2						
103	75	2	1	2	2								0	2				
104	75	2	1	2	2								0	2				
105	75	2	1	2	2								0	2				
106	41	2	1	2	0						0							
107	68	2	1	2	0										2	2	2	0
108	66	2	1	1	2								0	2				
109	75	2	1	2	2								0	2				
110	66	2	2	2	0								0	2				
111	66	2	0	2	2								0	2				
112	81	2	1	2	2										2	2	2	0
113	62	2	0	2	0										2	2	2	0
114	41	2	0	1	0						0	2						
115	66	2	0	2	2								0	2				
116	83	2	1	1	2								2	2				
117	50	2	1	1	0	2	2	0	N	0								
118	75	2	1	2	2								0	2				
119	75	2	1	2	2								0	2				
120	58	2	1	2	2								0	0				
121	66	2	1	1	2								0	2				
122	81	2	1	2	2										2	2	2	0
123	77	2	2	2	2	2	2	0	0	2								
124	81	2	1	2	2										2	2	2	0
125	75	2	1	2	2								0	2				

Quality Scores																		
126	66	2	2	2	0								0	2				
127	75	2	1	2	2								0	2				
128	75	2	2	2	2	2	2	0	N	0								
129	75	2	1	2	2								0	2				
130	75	2	1	2	2						0	2						
131	75	2	1	2	2						0	2						
132	66	2	1	1	2						0	2						
133	83	2	2	2	0						2	2						
134	75	2	1	2	2								0	2				
135	58	2	1	2	0						0	2						
136	41	2	1	2	0								0	0				
137	66	2	1	1	2								0	2				
138	58	2	1	2	0								0	2				
139	83	2	2	2	2								0	2				
140	50	2	0	2	0								0	2				
141	83	2	2	2	0						2	2						
142	58	2	1	2	2								0	0				
143	68	2	1	2	0										2	2	2	0
144	75	2	1	2	2						0	2						
145	10	2	2	2	2										2	2	2	2
146	75	2	1	2	2								0	2				
147	75	2	1	2	2								0	2				
148	58	2	0	1	2						0	2						
149.	91	2	1	2	2								2	2				
149.	91	2	1	2	2								2	2				
150	91	2	1	2	2						2	2						
151	10	2	2	2	2						2	2						
152	83	2	2	2	2						0	2						
153	68	2	1	2	0										2	2	2	0
154	81	2	1	2	2										2	2	2	0
155	66	2	2	2	0								0	2				
156	75	2	2	1	2								0	2				
157	58	2	1	2	0								0	2				
158	91	2	1	2	2								2	2				
159	93	2	1	2	2										2	2	2	2
160	68	2	1	2	0										2	2	2	0
161	66	2	2	2	2								0	0				
162	10	2	2	2	2						2	2						
163	58	2	1	2	0								0	2				
164	75	2	1	2	2								0	2				
165	75	2	1	2	2								0	2				
166	66	2	2	2	0								0	2				
167	75	2	1	2	0								2	2				
168	75	2	1	2	2								0	2				
169	83	2	2	2	2								0	2				
170	75	2	1	2	0								2	2				

Quality Scores																		
171	58	2	0	1	0								2	2				
172	91	2	1	2	2						2	2						
173	50	2	1	1	0								0	2				
174	75	2	1	2	2								0	2				
175	75	2	1	2	2								0	2				
176	83	2	2	2	0						2	2						
177	58	2	1	2	0								0	2				
178	75	2	1	2	2								0	2				
179	75	2	1	2	2								0	2				
180	75	2	1	2	2						0	2						
181	41	2	1	2	0						0	0						
182	10	2	2	2	2										2	2	2	2
183	75	2	1	2	2								0	2				
184	75	2	1	2	2						0	2						
185	62	2	2	2	0	2	2	0	N	0								
186	91	2	1	2	2						2	2						
187	10	2	2	2	2						2	2						
188	75	2	1	2	2						0	2						
189	91	2	1	2	2								2	2				
190.	93	2	1	2	2										2	2	2	2
190.	93	2	1	2	2										2	2	2	2
191	91	2	1	2	2								2	2				
192	83	2	2	2	2								0	2				
193	50	2	2	2	0								0	0				
194	68	2	2	1	0										2	2	2	0
195	66	2	0	2	2						0	2						
196	83	2	2	2	0						2	2						
197	50	2	1	1	0								0	2				
198	10	2	2	2	2						2	2						
199	93	2	1	2	2										2	2	2	2
200	66	2	2	2	0								0	2				
201	75	2	2	2	0										2	2	2	0
202.	66	2	1	1	2								2	0				
202.	66	2	1	1	2								2	0				
202.	66	2	1	1	2								2	0				
202.	66	2	1	1	2								2	0				
203	81	2	1	2	2										2	2	2	0
204	75	2	1	2	0								2	2				
205	58	2	1	2	0								0	2				
206	91	2	1	2	2								2	2				
207	93	2	1	2	2										2	2	2	2
208	91	2	1	2	2						2	2						
209	50	2	1	1	0								0	2				
210	55	2	2	2	0	2	0	0	0	2								
211	58	2	1	2	0								0	2				
212	58	2	1	2	0						0	2						

Quality Scores																		
213	50	2	1	1	0								0	2				
214	75	2	1	2	2						0	2						
215	75	2	2	1	2						0	2						
216	91	2	1	2	2								2	2				
217	66	2	1	1	2						0	2						
218	75	2	1	1	0										2	2	2	2
219	58	2	1	2	0								0	2				
220	75	2	1	2	2								0	2				
221	83	2	2	2	0								2	2				
222	58	2	1	2	0								0	2				
223	75	2	1	2	2								0	2				
224	66	2	2	2	0								0	2				
225	93	2	1	2	2										2	2	2	2
226	83	2	2	2	2								0	2				
227	91	2	1	2	2								2	2				
228	75	2	2	1	2								0	2				
229	66	2	2	2	2	2	0	0	2									
230	75	2	1	2	0						2	2						
231	58	2	1	2	0						0	2						
232	75	2	1	2	2								0	2				
233	66	2	1	1	2						0	2						
234	50	2	1	1	0								0	2				
235	75	2	1	2	2								0	2				
236	68	2	1	2	0										2	2	2	0
237	68	2	1	2	0										2	2	2	0
238	50	2	1	1	0								0	2				

APPENDIX B: EXTERNAL REVIEWER COMMENTS

External Reviewer's Comments				
Comment No.	Comment	Context	Explanation	Solution/Status
Comment [m1]	Quality of the reported empirical research? I expect you'd define and use some quality criteria…It's important that there is sufficient expertise in assessing the quality of the published literature, otherwise, the extracted data would have validity threats of its own.	Research question section	Quality criteria have already been defined in the protocol document and 3 researcher will participate in meetings regarding study selection, quality assessment and data extraction	Done
Comment [m2]	Too generic question…this can be one of the main aims of the study…then that aim should be broken down for concrete questions…may be if you identify the types of data to be extracted. it will become clear how such an abstract level questions will be addressed.	Research question itself	Although this is the main aim of the study but it is research question as well. It is kept too generic as the aim is to cove the whole SA area. Types of data has already been defined and explained in data extraction section	Done
Comment [m3]	There is no empirical SA literature – perhaps, you are	Research question	Grammatical/syntax error	Fixed

External Reviewer's Comments				
	talking about empirical studies reported on different technologies (i.e., methods, processes, and tools) SA.	itself		
Comment [m4]	There ca n be several aspect of assessing the strength of evidence.. What do you mean by the source of evidence?	Research question section	Source of evidence will tell source from where and how the data was obtained(by adopting which empirical method(s)); for instance the data gathered from a case study by observation or the data gathered from a case study by using interview questionnaire that is a more systematic and traceable source of evidence	Done
Comment [m5]	If this structure is being used according to the PICO principle…then PICO doesn't necessarily apply to all kinds of situations.	Search strategy	Guidelines of Kitchenham[13] was used to derive major terms and to form the search strings	Done
Comment [m6]	I am unable to understand the objective of structuring questions in this format…	Search strategy	It is just a method to explain the parts of RQs	Done
Comment [m7]	I think you meant…"Keywords",	Search strategy	Grammatical/syntax error	Fixed

External Reviewer's Comments				
	here..?			
Comment [m8]	Empirical – try to use the names of different empirical research methods, e.g.., case studies, surveys..	Search strategy	Already used names of different empirical methods in the search strings	Done
Comment [m9]	How are these keywords alternatives to the keyword like Empirical?	Search strategy	Case study, Experiment, Experience report and Survey are referred as empirical methods in literature that is why these terms was used as alternatives of empirical	Done
Comment [m10]	Have you already performed the pilot searches…then you should include the results in the protocol so it can be shown how your search strategy formulation progressed…	Search strategy	Pilot results are added in appendix section	Done
Comment [m11]	It is unlikely that you find the papers using keywords like SA in 70s or 80s even Parnas published work then…SA term started gaining attraction from 91…	Search strategy	Terms like System, Structure and Design are used along with SA to capture old literature when the term SA was not used	Done

External Reviewer's Comments				
Comment [m12}	How would you decide about the design and the architecture level studies? Any criteria to be used?	Study Selection Criteria	Confusion like these will be discussed and resolved in periodic meetings with research supervisor and co-supervisor	Done
Comment [m13]	But previously you indicated that papers from workshops will be included...	Study Selection Criteria	Grammatical/syntax error	Fixed
Comment [m14]	You may want to consult the quality assessment criteria used by Dyba and Dingsør for their study on empirical research of agile approaches....	Quality Instrument	Consulted the referred study and Study Quality Instrument was modified	Done
Comment [m15]	Provide full bibliographical details for all the references used in the protocol.	References	Grammatical/syntax error	Fixed
Comment [m16]	How many search strings you used for ACM? Complex search string worked for you?	Pilot Search Results	we didn't break down the expert search string. Breaking down the string in sub strings was required in older version of ACM	Done

APPENDIX C: LIST OF INCLUDED PRIMARY STUDIES

[S1] M. P. A. Grimán, L. Mendoza, F. Losavio, "Feature analysis for architectural evaluation methods," Journal of Systems and Software, vol. Volume 79, 2006.

[S2] P. A. Ahmad Waqas Kamal, "Modeling the variability of architectural patterns," presented at SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing.

[S3] F. Ahmed and L. F. Capretz, "The software product line architecture: An empirical investigation of key process activities," Information and Software Technology, vol. 50, pp. 1098-1113, 2008.

[S4] J. Aldrich, C. Chambers, and D. Notkin, "ArchJava: connecting software architecture to implementation," presented at Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on, 2002.

[S5] M. Ali Babar, L. Bass, and I. Gorton, "Factors influencing industrial practices of software architecture evaluation: An empirical investigation," presented at 3rd International Conference on Quality of Software Architectures, QoSA 2007, July 11, 2007 - July 13, 2007, Medford, MA, United states, 2007.

[S6] A. F. G. Ambra Molesini, Christina von Flach Garcia Chavez, Thais Vasconcelos Batista, "On the Quantitative Analysis of Architecture Stability in Aspectual Decompositions," in WICSA '08: Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), 2008.

[S7] A. G. Ambra Molesini, Christina von Flach Garcia Chavez, Thais Vasconcelos Batista, "Stability assessment of aspect-oriented software architectures: A quantitative study," Journal of Systems and Software, vol. Volume 83.

[S8] V. Ambriola, P. Ciancarini, and A. Corradini, "Declarative specification of the architecture of a software development environment," Software - Practice and Experience, vol. 25, pp. 143-174, 1995.

[S9] M. N. Anargyros Tsadimas, Dimosthenis Anagnostopoulos, "Evaluating software architecture in a model-based approach for enterprise information system design," presented at SHARK '10: Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge.

[S10] M. N. Anargyros Tsadimas, Dimosthenis Anagnostopoulos, "Handling Non-functional Requirements in Information System Architecture Design," in ICSEA '09: Proceedings of the 2009 Fourth International Conference on Software Engineering Advances, 2009.

[S11] B. Anders, J. Fellmann, M. Lindvall, and I. Rus, "Experimenting with Software Architecture Flexibility Using an Implementation of the Tactical Separation Assisted Flight Environment," presented at Software Engineering Workshop, 2005. 29th Annual IEEE/NASA, 2005.

[S12] M. H. T. Antony Tang, Jun Han, Hans Vliet, "Design Reasoning Improves Software Design Quality," presented at QoSA '08: Proceedings of the 4th International Conference on Quality of Software-Architectures: Models and Architectures, 2008.

[S13] C. H. Arie van Deursen1, Rainer Koschke3, Leon Moonen4, Claudio Riva5, "Symphony: View-Driven Software Architecture Reconstruction," Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S14] H. S. o. Arjan de Roo, Mehmet Aks¸it, "An Architectural Style for Optimizing System Qualities in Adaptive Embedded Systems using Multi-Objective Optimization," in IEEE/IFIP WICSA/ECSA, 2009.

[S15] A. Arsanjani, "Empowering the business analyst for on demand computing," IBM Systems Journal, vol. 44, pp. 67-80, 2005.

[S16] J. M. F. Atif Mashkoor, "Deriving Software Architectures for CRUD Applications: The FPL Tower Interface Case Study," 2007.

[S17] J. Axelsson, "Evolutionary Architecting of Embedded Automotive Product Lines:

An Industrial Case Study," in IEEE/IFIP WICSA/ECSA, 2009.

[S18] M. A. Babar, "An exploratory study of architectural practices and challenges in using agile software development approaches," presented at Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on, 2009.

[S19] M. A. Babar and S. Biffl, "Eliciting better quality architecture evaluation scenarios: A controlled experiment on top-down vs. bottom-up," presented at ISCE'06 - 5th ACM-IEEE International Symposium on Empirical Software Engineering, September 21, 2006 - September 22, 2006, Rio de Janeiro, Brazil, 2006.

[S20] M. A. Babar, B. Kitchenham, and I. Gorton, "Towards a distributed software architecture evaluation process - A preliminary assessment," presented at 28th International Conference on Software Engineering 2006, ICSE '06, May 20, 2006 - May 28, 2006, Shanghai, China, 2006.

[S21] M. A. Babar, B. Kitchenham, and R. Jeffery, "Comparing distributed and face-to-face meetings for software architecture evaluation: A controlled experiment," Van Godewijckstraat 30, Dordrecht, 3311 GZ, Netherlands, 2008.

[S22] M. A. Babar, B. Kitchenham, and M. Piyush, "The value of architecturally significant information extracted from patterns for architecture evaluation: a controlled experiment," presented at Software Engineering Conference, 2006. Australian, 2006.

[S23] M. A. Babar, B. Kitchenham, L. Zhu, I. Gorton, and R. Jeffery, "An empirical study of groupware support for distributed software architecture evaluation process," Journal of Systems and Software, vol. 79, pp. 912-925, 2006.

[S24] M. A. Babar, B. Kitchenham, L. Zhu, and R. Jeffery, "An exploratory study of groupware support for distributed software architecture evaluation process," presented at Software Engineering Conference, 2004. 11th Asia-Pacific, 2004.

[S25] M. A. Babar, A. Northway, I. Gorton, P. Heuer, and N. Thong, "Introducing Tool Support for Managing Architectural Knowledge: An Experience Report," presented at Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the, 2008.

[S26] R. Bahsoon and W. Emmerich, "An Economics-Driven Approach for Valuing Scalability in Distributed Architectures," presented at Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference on, 2008.

[S27] S. Balsamo, M. Marzolla, A. Di Marco, and P. Inverardi, "Experimenting different software architectures performance techniques: A case study," presented at Proceedings of the Fourth International Workshop on Software and Performance, WOSP'04, January 14, 2004 - January 16, 2004, Redwood Shores, CA, United states, 2004.

[S28] K. S. Barber, T. Graser, and J. Holt, "Enabling iterative software architecture derivation using early non-functional property evaluation," presented at Automated Software Engineering, 2002. Proceedings. ASE 2002. 17th IEEE International Conference on, 2002.

[S29] K. S. Barber, J. Holt, and G. Baker, "Performance evaluation of domain reference architectures," presented at 14th International Conference on Software Engineering and Knowledge Engineering, SEKE '02, July 15, 2002 - July 19, 2002, Ischia, Italy, 2002.

[S30] J. E. Bardram, H. B. Christensen, and K. M. Hansen, "Architectural prototyping: an approach for grounding architectural design and learning," presented at Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on, 2004.

[S31] A. Bertolino, H. Muccini, and A. Polini, "Architectural verification of black-box component-based systems," presented at 3rd International Workshop on Rapid Integration of Software Engineering Techniques, RISE 2006, September 13, 2006 - September 15, 2006, Geneva, Switzerland, 2007.

[S32] B. Biel and V. Gruhn, "Analyzing the Architectural Support of Usability," presented at Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on.

[S33] R. A. Bittencourt and D. Guerrero, "Comparison of Graph Clustering Algorithms for Recovering Software Architecture Module Views," presented at Software Maintenance and Reengineering, 2009. CSMR '09. 13th European Conference on, 2009.

[S34] L. B. Bonnie E. John, Elspeth Golden, Pia Stoll, "A responsibility-based pattern language for usability-supporting architectural patterns," presented at EICS '09: Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems, 2009.

[S35] N. Boucké, D. Weyns, and T. Holvoet, "Composition of architectural models: Empirical analysis and language support," Journal of Systems and Software, vol. 83, pp. 2108-2127.

[S36] E. Bouwers, J. Visser, and A. van Deursen, "Criteria for the evaluation of implemented architectures," presented at Software Maintenance, 2009. ICSM 2009. IEEE International Conference on, 2009.

[S37] I. T. Bowman, R. C. Holt, and N. V. Brewster, "Linux as a case study: its extracted software architecture," presented at Software Engineering, 1999. Proceedings of the 1999 International Conference on, 1999.

[S38] H. P. Breivold, I. Crnkovic, R. Land, and S. Larsson, "Using dependency model to support software architecture evolution," presented at Automated Software Engineering - Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM International Conference on, 2008.

[S39] S. M. Brian S. Mitchell and M. Traverso, "Using Interconnection Style Rules to Infer Software Architecture Relations," in Lecture Notes in Computer Science,Genetic and Evolutionary Computation – GECCO 2004, vol. Volume 3103, 2004, pp. Pages 1375-1387.

[S40] P. H. S. Brito, R. De Lemos, C. M. F. Rubira, and E. Martins, "Architecting fault tolerance with exception handling: Verification and validation," Journal of Computer Science and Technology, vol. 24, pp. 212-237, 2009.

[S41] J. A. D.-P. J. P. C. M. B. A. S. M. R. Campo2, "Assisting the Synchronization of UCM-based Architectural Documentation with Implementation" in IEEE/IFIP WICSA/ECSA, 2009.

[S42] F. M. Carmen Zannier, "Social Factors Relevant to Capturing Design Decisions," in SHARK-ADI'07, 2007.

[S43] M. C. Carmen Zannier, Frank Maurer, "A model of design decision making based on empirical results of interviews with software designers," in Information and Software Technology, vol. Volume 49. 2007.

[S44] B. J. W. a. J. C. Carver, "Characterizing Software Architecture Changes: An Initial Study

" presented at First International Symposium on Empirical Software Engineering and Measurement

2007.

[S45] M. Castaldi, P. Inverardi, and S. Afsharian, "A case study in performance, modifiability and extensibility analysis of a telecommunication system software architecture," presented at Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on, 2002.

[S46] S. Chang-ai, J. Zhou, C. Jiannong, J. Maozhong, L. Chao, and S. Yanfang, "ReArchJBs: a tool for automated software architecture recovery of JavaBeans-based applications," presented at Software Engineering Conference, 2005. Proceedings. 2005 Australian, 2005.

[S47] H. Changjun, J. Feng, and Z. Chongchong, "An Architectural Quality Assessment for Domain-Specific Software," presented at Computer Science and Software Engineering, 2008 International Conference on, 2008.

[S48] C. L. Chen, S. Danhua, and D. E. Perry, "An Exploratory Case Study Using CBSP and Archium," presented at Sharing and Reusing Architectural Knowledge - Architecture, Rationale, and Design Intent, 2007. SHARK/ADI '07: ICSE Workshops 2007. Second Workshop on, 2007.

[S49] H.-M. Chen, R. Kazman, and O. Perry, "From software architecture analysis to service engineering: An empirical study of methodology development for enterprise SOA implementation," IEEE Transactions on Services Computing, vol. 3, pp. 145-160.

[S50] A. M. Christian Del Rosso, "Assessing the Architectonics of Large, Software-Intensive Systems Using a Knowledge-Based Approach," in WICSA '07: Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture, 2007.

[S51] L. O. B. Christoph Stoermer1, "MAP - Mining Architectures for Product Line Evaluations," 2001.

[S52] A. R. Christoph Stoermer, Liam O'Brien, Chris Verhoef, "Model-centric software architecture reconstruction," in Software—Practice & Experience, vol. Volume 36, 2006.

[S53] L. O. B. Christoph Stoermer, Chris Verhoef, "Moving Towards Quality Attribute Driven Software Architecture Reconstruction," presented at WCRE '03: Proceedings of the 10th Working Conference on Reverse Engineering, 2003.

[S54] P. S. Claudio Riva1, Tarja Systä2, Jianli Xu1, "UML-based Reverse Engineering and Model Analysis Approaches for Software Architecture Maintenance," presented at Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04), 2004.

[S55] V. Clerc, "Towards Architectural Knowledge Management Practices for Global Software Development," 2008.

[S56] F. Corradini, P. Inverardi, and A. L. Wolf, "On relating functional specifications to architectural specifications: A case study," Science of Computer Programming, vol. 59, pp. 171-208, 2006.

[S57] V. Cortellessa, H. Singh, B. Cukic, E. Gunel, and V. Bharadwaj, "Early reliability assessment of UML based software models," presented at Proceedings of the Third International Workshop on Software and Performance WOSP2002, July 24, 2002 - July 26, 2002, Rome, Italy, 2002.

[S58] F. Cuadrado, B. Garcia, J. C. Dueas, and H. A. Parada, "A Case Study on Software Evolution towards Service-Oriented Architecture," presented at Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on, 2008.

[S59] P. I. Daniele Compare, Patrizio Pelliccione and A. Sebastiani, "Integrating Model-Checking Architectural Analysis and Validation in a Real Software Life-Cycle," in Lecture Notes in Computer Science, FME 2003: Formal Methods, vol. Volume 2805, 2003, pp. Pages 114-132.

[S60] E. M. Dashofy, A. Van Der Hoek, and R. N. Taylor, "A comprehensive approach for the development of modular software architecture description languages," ACM Transactions on Software Engineering and Methodology, vol. 14, pp. 199-245, 2005.

[S61] R. C. Davide Falessi1, Giovanni Cantone3, "A Value-Based Approach for Documenting Design Decisions Rationale: A Replicated Experiment," in SHARK'08, 2008.

[S62] M. A. B. Davide Falessi, Giovanni Cantone, Philippe Kruchten, "Applying empirical software engineering to software architecture: challenges and lessons learned," in Empirical Software Engineering, vol. Volume 15.

[S63] V. C. de Paula, G. R. B. Justo, and P. R. F. Cunha, "Specifying and verifying reconfigurable software architectures," presented at Software Engineering for Parallel and Distributed Systems, 2000. Proceedings. International Symposium on, 2000.

[S64] P. N. Deepti Srivastava, "Architectural support for mode-driven fault tolerance in distributed applications," presented at WADS '05: Proceedings of the 2005 workshop on Architecting dependable systems, 2005.

[S65] C. Del Rosso, "Continuous evolution through software architecture evaluation: A case study," Journal of Software Maintenance and Evolution, vol. 18, pp. 351-383, 2006.

[S66] D. B. Edgardo Luzcando, Anthony Faiola, "Evaluating Usability-Supporting Architecture Patterns: Reactions from Usability Professionals," presented at Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends, 2009.

[S67] I. G. Eduardo Figueiredo1, Safoora Shakil Khan1, Alessandro Garcia3,, A. P. Claudio Sant'Anna3, Ana Luisa Medeiros5, Lyrene Fernandes5, Thais Batista5,, and P. v. d. B. Rita Ribeiro4, Mehmet Aksit2, Steffen Zschaler1, and Ana Moreira4, "Detecting Architecture Instabilities with Concern Traces: An Exploratory Study," in IEEE/IFIP WICSA/ECSA, 2009.

[S68] W. Eixelsberger, M. Ogris, H. Gall, and B. Bellay, "Software architecture recovery of a program family," presented at Software Engineering, 1998. Proceedings of the 1998 International Conference on, 1998.

[S69] U. Eklund and C. M. Olsson, "A case study of the Architecture Business Cycle for an in-vehicle software architecture," presented at Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on, 2009.

[S70] E. P. M. d. S. Elisa Yumi Nakagawa, Kiyoshi de Brito Murata, Gabriel de Faria Andery, Leonardo Bitencourt Morelli, José Carlos Maldonado, "Software Architecture Relevance in Open Source Software Evolution: A Case Study," presented at COMPSAC '08: Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, 2008.

[S71] V.-P. Eloranta and K. Koskimies, "Using Domain Knowledge to Boost Software Architecture Evaluation," in Lecture Notes in Computer Science,Software Architecture, vol. Volume 6285, pp. Pages 319-326.

[S72] B. E. J. Elspeth Golden, Len Bass, "The value of a usability-supporting architectural pattern in software architecture design: a controlled experiment," presented at ICSE '05: Proceedings of the 27th international conference on Software engineering, 2005.

[S73] N. H. Erwan Jahier, Pascal Raymond, Xavier Nicollin, David Lesens, "Virtual execution of AADL models via a translation into synchronous programs," 2007.

[S74] T. Everitt, R. T. Tvedt, and J. D. Tvedt, "Validating and improving an existing software architectural evaluation process," presented at Software Maintenance, 2004. Proceedings. 20th IEEE International Conference on, 2004.

[S75] M. Feilkas, D. Ratiu, and E. Jurgens, "The loss of architectural knowledge during system evolution: An industrial case study," presented at Program Comprehension, 2009. ICPC '09. IEEE 17th International Conference on, 2009.

[S76] J. Feng, H. Changjun, and Z. Chongchong, "A Software Complexity Metric for SCA Specification," presented at Computer Science and Software Engineering, 2008 International Conference on, 2008.

-[S77] F. A. R. a. H. Y. Feras T. Dabous, "Using Software Architectures and Design Patterns for Developing Distributed Applications," presented at Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), 2004.

[S78] J. L. Fernandez, "An Architectural Style for Object Oriented Real-Time Systems," 1998.

[S79] R. Ferrari and N. H. Madhavji, "The Impact of Requirements Knowledge and Experience on Software Architecting: An Empirical Study," presented at Software Architecture, 2007. WICSA '07. The Working IEEE/IFIP Conference on, 2007.

[S80] R. Ferrari, N. H. Madhavji, and M. Wilding, "The impact of non-technical factors on Software Architecture," presented at Leadership and Management in Software Architecture, 2009. LMSA 2009. ICSE Workshop on, 2009.

[S81] E. Figueiredo, N. Cacho, C. Sant'Anna, M. Monteiro, U. Kulesza, A. Garcia, S. Soares, F. Ferrari, S. Khan, F. C. Filho, and F. Dantas, "Evolving software product lines with Aspects: An empirical study on design stability," presented at 2008 ACM/IEEE 30th International Conference on Software Engineering, ICSE 2008, May 10, 2008 - May 18, 2008, Leipzig, Germany, 2008.

[S82] E. Folmer and J. Bosch, "Experiences with software architecture analysis of usability," International Journal of Information Technology and Web Engineering, vol. 3, pp. 1-29, 2008.

[S83] H. S. Franck Chauvel, Xiang Ping Chen, Gang Huang, and Hong Mei, "Using QoS-Contracts to Drive Architecture-Centric Self-adaptation," in QoSA, 2010.

[S84] D. Ganesan, T. Keuler, and Y. Nishimura, "Architecture compliance checking at run-time," Information and Software Technology, vol. 51, pp. 1586-1600, 2009.

[S85] J. R. E. Girish Suryanarayana, Scott A. Hendrickson, Richard N. Taylor, "PACE: An Architectural Style for Trust Management in Decentralized Applications," presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S86] E. Golden, B. E. John, and L. Bass, "Quality vs. quantity: comparing evaluation methods in a usability-focused software architecture modification task," presented at Empirical Software Engineering, 2005. 2005 International Symposium on, 2005.

[S87] A. W. Goran Mustapic, Christer Norström, Ivica Crnkovic, and J. F. Kristian Sandström, Johan Andersson, "Real World Influences on Software Architecture - Interviews with Industrial System Experts" presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S88] C. A. Gurr, "Software architectures in industrial systems: An empirical study," International Software Architecture Workshop, Proceedings, ISAW, pp. 57-60, 1998.

[S89] W. Heaven, D. Sykes, J. Magee, and J. Kramer, "A case study in goal-driven architectural adaptation," presented at Software Engineering for Self-Adaptive Systems, Tiergartenstrasse 17, Heidelberg, D-69121, Germany, 2009.

[S90] W. Heijstek and M. R. V. Chaudron, "The Impact of Model Driven Development on the Software Architecture Process," presented at Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on.

[S91] R. W. Heiko Koziolek, Jens Doppelhamer, "Evolving Industrial Software Architectures into a Software Product Line: A Case Study," presented at Proceedings of the 5th International Conference on the Quality of Software Architectures: Architectures for Adaptive Software Systems, 2009.

[S92] K. M. H. Henrik Bærbak Christensen, "An empirical investigation of architectural prototyping," Journal of Systems and Software, vol. Volume 83.

[S93] S. Herold, "Checking architectural compliance in component-based systems," presented at 25th Annual ACM Symposium on Applied Computing, SAC 2010, March 22, 2010 - March 26, 2010, Sierre, Switzerland.

[S94] X. Hongzhen and Z. Guosun, "Specification and verification of dynamic evolution of software architectures," Journal of Systems Architecture, vol. 56, pp. 523-533.

[S95] Y. Huiqun, H. Xudong, D. Yi, and M. Lian, "A formal method for analyzing software architecture models in SAM," presented at Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International, 2002.

[S96] H. G. a. M. Hussein**, "Software Reconfiguration Patterns for Dynamic Evolution of Software Architectures," presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S97] S. A. a. E. Insfran, "Early Usability Evaluation in Model Driven Architecture Environments," presented at Proceedings of the Sixth International Conference on Quality Software (QSIC'06), 2006.

[S98] P. W. Ipek Ozkaya1, Jakob Axelsson3, "Architecture Knowledge Management during System Evolution – Observations from Practitioners," in SHARK'10,, 2010.

[S99] C. K. a. V. Issarny, "SPIN-ning Software Architectures: A Method for Exploring Complex Systems," 2001.

[S100] V. Issarny and A. Zarras, "Software Architecture and Dependability," in Lecture Notes in Computer Science, Formal Methods for Software Architectures, vol. Volume 2804, 2003, pp. Pages 259-285.

[S101] H. M. Ivano Malavolta, Patrizio Pelliccione, Damien Tamburri, "Providing Architectural Languages and Tools Interoperability through Model Transformation Technologies," in IEEE Transactions on Software Engineering, vol. Volume 36.

[S102] J. D. J. Chattratichat, Y. Guo, S. Hedvall, M. Köhler, and et al., "A software architecture for deploying high performance solution on the Internet," in Lecture Notes in Computer Science,High-Performance Computing and Networking, vol. Volume 1401, 1998, pp. Pages 545-555.

[S103] J. M. J. Jenny Li, Joseph R. Horgan, "Automatic Simulation to Predict Software Architecture Reliability," presented at ISSRE '97: Proceedings of the Eighth International Symposium on Software Reliability Engineering, 1997.

[S104] Y. L. Jacky W. Keung, Kate Foster, Thong Nguyen, "A Statistical Method for Middleware System Architecture Evaluation," presented at ASWEC '10: Proceedings of the 2010 21st Australian Software Engineering Conference.

[S105] R. F. James A. Miller, Nazim H. Madhavji, "An exploratory study of architectural effects on requirements decisions," Journal of Systems and Software, vol. Volume 83, pp. 2441-2455, 2010.

[S106] A. Jansen, P. Avgeriou, and J. S. van der Ven, "Enriching software architecture documentation," Journal of Systems and Software, vol. 82, pp. 1232-1248, 2009.

[S107] M. Jazayeri, "On Architectural Stability and Evolution," in Lecture Notes in Computer Science,Reliable Software Technologies — Ada-Europe 2002, vol. Volume 2361, 2002, pp. Pages 397-420.

[S108] K. X. Jeffrey J. P. Tsai, "An empirical evaluation of deadlock detection in software architecture specifications," Annals of Software Engineering, vol. Volume 7, 1999.

[S109] K. X. Jeffrey J. P. Tsai, "A comparative study of formal verification techniques for software architecture specifications Annals of Software Engineering," vol. Volume 10, 2000.

[S110] D. M. Jens Knodel, Matthias Naab, Mikael Lindvall, "Static Evaluation of Software Architectures," presented at CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering, 2006.

[S111] D. M. Jens Knodel, Matthias Naab, "Understanding Software Architectures by Visualization – An Experiment with Graphical Elements" presented at Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06), 2006.

[S112] W. Jing, Y. Shi, Y. Ni, and Z. Linlin, "Architectural design of the Online auction system with AOSAD," presented at IEEE International Conference on e-Business Engineering, ICEBE'08, October 22, 2008 - October 24, 2008, Xi'an, China, 2008.

[S113] W. Jiwen, G. Chenghao, and L. Fengyu, "Self-healing based software architecture modeling and analysis through a case study," presented at Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE, 2005.

[S114] M. C. Johan Muskens, Christian Lange, "Investigations in Applying Metrics to Multi-View Architecture Models," presented at EUROMICRO '04: Proceedings of the 30th EUROMICRO Conference (EUROMICRO'04), 2004.

[S115] J. P. Jukka Kaartinen, Kai Koskimies, "A Pattern-Driven Process Model for Quality-Centered Software Architecture Design--A Case Study on Usability-Centered Design," presented at ASWEC '07: Proceedings of the 2007 Australian Software Engineering Conference, 2007.

[S116] S. Kalyanasundaram, K. Ponnambalam, and A. Singh, "Metrics for software architecture: a case study in the telecommunication domain," presented at Proceedings of the 1998 11th Canadian Conference on Electrical and Computer Engineering, CCECE. Part 2 (of 2), May 24, 1998 - May 28, 1998, Toronto, Can, 1998.

[S117] H. K. Kangtae Kim, Sundeok Kim, Gihun Chang, "A case study on SW product line architecture evaluation: experience in the consumer electronics domain," in Third International Conference on Software Engineering Advances, 2008.

[S118] R. Kazman, G. Abowd, L. Bass, and P. Clements, "Scenario-based analysis of software architecture," Software, IEEE, vol. 13, pp. 47-55, 1996.

[S119] R. Kazman, H. P. In, and H.-M. Chen, "From requirements negotiation to software architecture decisions," Information and Software Technology, vol. 47, pp. 511-520, 2005.

[S120] D. Kim and S. Park, "Designing dynamic software architecture for home service robot software," presented at International Conference on Embedded and Ubiquitous Computing, EUC 2006, August 1, 2006 - August 4, 2006, Seoul, Korea, Republic of, 2006.

[S121] J. Kim, S. Park, and V. Sugumaran, "DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines," Journal of Systems and Software, vol. 81, pp. 37-55, 2008.

[S122] T. Kim, I.-Y. Ko, S.-W. Kang, and D.-H. Lee, "Extending ATAM to assess product line architecture," presented at 2008 IEEE 8th International Conference on Computer and Information Technology, CIT 2008, July 8, 2008 - July 11, 2008, Sydney, NSW, Australia, 2008.

[S123] T. Kishi, N. Noda, and T. Katayama, "Architectural design for evolution by analyzing requirements on quality attributes," presented at Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific, 2001.

[S124] C. M. S. Klaus Krogmann, Sabine Buckl, Michael Kuperberg, Anne Martens, Florian Matthes, "Improved Feedback for Architectural Performance Prediction Using Software Cartography Visualizations," presented at QoSA '09: Proceedings of the 5th International Conference on the Quality of Software Architectures: Architectures for Adaptive Software Systems, 2009.

[S125] A. S. Klusener, R. Lämmel, and C. Verhoef, "Architectural modifications to deployed software," Science of Computer Programming, vol. 54, pp. 143-211, 2005.

[S126] B. Lague, C. Leduc, A. Le Bon, E. Merlo, and M. Dagenais, "An analysis framework for understanding layered software architectures," presented at Program Comprehension, 1998. IWPC '98. Proceedings., 6th International Workshop on, 1998.

[S127] W. Lam, "Creating reusable architectures: initial experience report," in SIGSOFT Software Engineering Notes, vol. Volume 22, 1997.

[S128] E. Landre, H. Wesenberg, and H. Rnneberg, "Architectural improvement by use of strategic level domain-driven design," presented at 21st Annual ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages, and Applications, OOPSLA 2006, October 22, 2006 - October 26, 2006, Portland, OR, United states, 2006.

[S129] N. Lassing, D. Rijsenbrij, and H. van Vliet, "Towards a broader view on software architecture analysis of flexibility," presented at Software Engineering Conference, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific, 1999.

[S130] N. Lassing, D. Rijsenbrij, and H. Van Vliet, "Viewpoints on modifiability," International Journal of Software Engineering and Knowledge Engineering, vol. 11, pp. 453-478, 2001.

[S131] Y. Lau, W. Zhao, X. Peng, Y. Chen, and Z. Jiang, "A unified formal model for supporting Aspect-Oriented Dynamic Software architecture," presented at 2nd International Conference on Convergent Information Technology, ICCIT 07, November 21, 2007 - November 23, 2007, Gyongju, Korea, Republic of, 2007.

[S132] J. Lee, S. Ha, K. C. Kang, Y. Choo, Y. Hong, and H. Hwang, "Quality requirement elicitation for the architecture evaluation of process computer systems," presented at 8th Asia Pacific Software Engineering Conference APSEC'2001, December 4, 2001 - December 7, 2001, Macao, China, 2001.

[S133] J. Lee, S. Kang, and C.-K. Kim, "Software architecture evaluation methods based on cost benefit analysis and quantitative decision making," Empirical Software Engineering, vol. 14, pp. 453-475, 2009.

[S134] N. M. Lei Dingl, "Focus_A Light-Weight, Incremental Approach to Software Architecture R-ecovery and Evolution," 2001.

[S135] J. J. Li and J. R. Horgan, "Applying formal description techniques to software architectural design," Computer Communications, vol. 23, pp. 1169-1178, 2000.

[S136] P. Liang, A. Jansen, and P. Avgeriou, "Sharing architecture knowledge through models: Quality and cost," Knowledge Engineering Review, vol. 24, pp. 225-244, 2009.

[S137] M. Lindvall, W. C. Stratton, D. E. Sibol, C. Ackermann, W. M. Reid, D. Ganesan, D. McComas, M. Bartholomew, and S. Godfrey, "Connecting research and practice: an experience report on research infusion with software architecture visualization and evaluation," pp. 1-23.

[S138] C. López, V. Codocedo, H. Astudillo, and L. M. Cysneiros, "Bridging the gap between software architecture rationale formalisms and actual architecture documents: An ontology-driven approach," Science of Computer Programming, vol. In Press, Corrected Proof.

[S139] C. López, P. Inostroza, L. M. Cysneiros, and H. Astudillo, "Visualization and comparison of architecture rationale with semantic web technologies," Journal of Systems and Software, vol. 82, pp. 1198-1210, 2009.

[S140] F. Luders, I. Crnkovic, and P. Runeson, "Adopting a component-based software architecture for an industrial control system - A case study," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3778 LNCS, pp. 232-248, 2005.

[S141] C.-H. Lung and K. Kalaichelvan, "Approach to quantitative software architecture sensitivity analysis," International Journal of Software Engineering and Knowledge Engineering, vol. 10, pp. 97-114, 2000.

[S142] C.-H. Lung, P. Rajeswaran, S. Sivadas, and T. Sivabalasingam, "Experience of building an architecture-based generator using GenVoca for distributed systems," Science of Computer Programming, vol. 75, pp. 672-688.

[S143] W. Luo, H. Ghenniwa, W. Shen, and Z. Rao, "Capture Dynamic Aspects of Software Architecture for Distributed Self-Adaptive Environments," presented at Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on, 2006.

[S144] A. Maccari, "Experiences in assessing product family software architecture for evolution," presented at Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on, 2002.

[S145] S. Malek, M. Mikic-Rakic, and N. Medvidovic, "A style-aware architectural middleware for resource-constrained, distributed systems," Software Engineering, IEEE Transactions on, vol. 31, pp. 256-272, 2005.

[S146] S. Malek, C. Seo, S. Ravula, B. Petrus, and N. Medvidovic, "Providing middleware-level facilities to support architecture-based development of software systems in pervasive environments," presented at 4th International Workshop on Middleware for Pervasive and Ad-hoc Computing, MPAC 2006, held at the ACM/IFIP/USENIX 7th International Middleware Conference, November 27, 2006 - December 1, 2006, Melbourne, Australia, 2006.

[S147] A. G. María Pérez, Francisca Losavio, "Simulation-Based Architectural Evaluation for Collaborative Systems," presented at SCCC '02: Proceedings of the XII International Conference of the Chilean Computer Science Society, 2002.

[S148] J. H. Markku Hakala1, Kai Koskimies1 Jukka Paakki2, Antti Viljamaa2, Jukka Viljamaa2, "Annotating Reusable Software Architectures with Specialization Patterns," 2001.

[S149] F. Martensson, H. Grahn, and M. Mattsson, "An approach for performance evaluation of software architectures using prototyping," presented at Proceedings of the Seventh IASTED International Conference on Software Engineering and Applications, November 3, 2003 - November 5, 2003, Marina del Rey, CA, United states, 2003.

[S150] T. L. Marwan Abi-Antoun Talia Selitsky1, "Developer Refinement of Runtime Architectural Structure," in SHARK'10, 2010.

[S151] M. Matinlassi, "Evaluating the Portability and Maintainability of Software Product Family

Architecture: Terminal Software Case Study" presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S152] M. Matinlassi, E. Niemela, and L. Dobrica, "Quality-driven architecture design and quality analysis method: A revolutionary initiation approach to a product line architecture," VTT Publications, 2002.

[S153] A. Mattsson1, B. Lundell2, B. Lings2, and B. Fitzgerald3, "Experiences from representing software architecture in a large industrial project using model driven development," in SHARK-ADI'07, 2007.

[S154] A. Mattsson, "Automatic enforcement of architectural design rules," presented at 32nd ACM/IEEE International Conference on Software Engineering, ICSE 2010, May 1, 2010 - May 8, 2010, Cape Town, South africa.

[S155] D. S. McCrickard and G. D. Abowd, "Assessing the impact of changes at the architectural level: a case study on graphical debuggers," presented at Software Maintenance 1996, Proceedings., International Conference on, 1996.

[S156] M. S. Mehwish Riaz and H. Naqvi, "Architectural Decay during Continuous Software Evolution and Impact of 'Design for Change' on Software Architecture," in Communications in Computer and Information Science, 1, Advances in Software Engineering, vol. Volume 59, pp. Pages 119-126.

[S157] C. Mei-Hwa, T. Mei-Huei, and W. Wen-Li, "Effect of architecture configuration on software reliability and performance estimation," presented at Application-Specific Software Engineering Technology, 1998. ASSET-98. Proceedings. 1998 IEEE Workshop on, 1998.

[S158] K. P. Michael Goedicke and U. Zdun, "Domain-Specific Runtime Variability in Product Line Architectures," in Lecture Notes in Computer Science,Object-Oriented Information Systems, vol. Volume 2425, 2002, pp. Pages 213-218.

[S159] C. W. Mikael Svahnberg, "An Investigation of a Method for Identifying a Software Architecture Candidate with Respect to Quality Attributes," in Empirical Software Engineering, vol. Volume 10: Kluwer Academic Publishers Hingham, MA, USA 2005, pp. 149-181.

[S160] T. G. ı. Mircea Lungu and Michele Lanzal, "Package Patterns for Visual Architecture Recovery" Proceedings of the Conference on Software Maintenance and Reengineering (CSMR'06), 2006.

[S161] B. K. Muhammad Ali Babarl, "The Impact of Group Size on Software Architecture Evaluation: A Controlled Experiment," in First International Symposium on Empirical Software Engineering and Measurement, 2007.

[S162] M. R. Muhammad Sulayman, Husnain Naqvi and M. A. Aman, "Using Area-Team to Evaluate the Software Architecture for an Online Banking System: A Case Study," in Communications in Computer and Information Science, 1, Advances in Software Engineering, vol. Volume 59, pp. Pages 111-118.

[S163] S. Nakajima and T. Tamai, "Behavioural analysis of the enterprise javaBeans™ component architecture," in Lecture Notes in Computer Science, Model Checking Software, vol. Volume 2057, 2001, pp. Pages 163-182.

[S164] T. Nakamura and V. R. Basili, "Metrics of software architecture changes based on structural distance," presented at 11th IEEE International Software Metrics Symposium, METRICS 2005, September 19, 2005 - September 22, 2005, Como, Italy, 2005.

[S165] R. S. S. a. C. J. Neill, "Characterizing Essential and Incidental Complexity in Software Architectures" in IEEE/IFIP WICSA/ECSA, 2009.

[S166] E. Niemelä and A. Immonen, "Capturing quality requirements of product family architecture," Information and Software Technology, vol. 49, pp. 1107-1120, 2007.

[S167] L. O'Brien, D. Smith, and G. Lewis, "Supporting Migration to Services using Software Architecture Reconstruction," presented at Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on, 2005.

[S168] P. Ojala, "Developing Value Assessment for SW Architecture," presented at Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference on, 2008.

[S169] F. G. Olumofin and V. B. Misic, "A holistic architecture assessment method for software product lines," Information and Software Technology, vol. 49, pp. 309-323, 2007.

[S170] H. B. Onaiza Maqbool, "Hierarchical Clustering for Software Architecture Recovery," in IEEE Transactions on Software Engineering, vol. Volume 33, 2007.

[S171] K. K. Outi Raiha, Erkki Makinen, "Scenario-Based Genetic Synthesis of Software Architecture," presented at ICSEA '09: Proceedings of the 2009 Fourth International Conference on Software Engineering Advances, 2009.

[S172] P. Ovaska, M. Rossi, and P. Marttiin, "Architecture as a coordination tool in multi-site software development," Software Process Improvement and Practice, vol. 8, pp. 233-247, 2003.

[S173] D. B. a. S. Overhage, "Is BPMN Really First Choice in Joint Architecture Development? An Empirical Study on the Usability of BPMN and UML Activity Diagrams for Business Users," in QoSA, 2010.

[S174] C. M. P. Ciancarini, "A Catalog of Architectural Styles for Mobility," 1998.

[S175] R. F. P. Tonella, G. Antoniol, E. Merlo, "Augmenting Pattern-Based Architectural Recovery with Flow Analysis: Mosaic - a Case Study," presented at WCRE '96: Proceedings of the 3rd Working Conference on Reverse Engineering (WCRE '96), 1996.

[S176] S. R. Paris Avgeriou and M. Skordalakis, "Building Quality into Learning Management Systems - An Architecture-Centric Approach," in Lecture Notes in Computer Science,Advanced Conceptual Modeling Techniques, vol. Volume 2784, 2003, pp. Pages 312-324.

[S177] J. J. Pasi Alatalo, Jari Karvonen, Ari Keronen and P. Kuvajal, "Mobile Application Architectures," in Lecture Notes in Computer Science,Product Focused Software Process Improvement, vol. Volume 2559, 2002, pp. Pages 572-586.

[S178] L. O. B. Paul Brebner, Jon Gray, "Performance Modeling Evolving Enterprise Service Oriented Architectures" in IEEE/IFIP WICSA/ECSA, 2009.

[S179] M. J. H. a. D. E. P. Paul S Grisham, "Architecture and Design Intent: An Experience Report," in Second Workshop on SHAring and Reusing architectural

Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI'07), 2007.

[S180] J. Pérez, N. Ali, J. A. CarsI´, I. Ramos, B. Álvarez, P. Sanchez, and J. A. Pastor, "Integrating aspects in software architectures: PRISMA applied to robotic tele-operated systems," Information and Software Technology, vol. 50, pp. 969-990, 2008.

[S181] M. D. Petty, F. D. McKenzie, and X. Qingwen, "Using a software architecture description language to model the architecture and run-time performance of a federate," presented at Distributed Simulation and Real-Time Applications, 2002. Proceedings. Sixth IEEE International Workshop on, 2002.

[S182] D. Port and H. LiGuo, "Strategic architectural flexibility," presented at Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on, 2003.

[S183] A. Postma, "A method for module architecture verification and its application on a large component-based system," Information and Software Technology, vol. 45, pp. 171-194, 2003.

[S184] A. Purhonen, "Performance Optimization of Embedded Software Architecture – A Case Study," presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S185] W. Qing. H. Weihua, and D. Wen, "A semantic and adaptive middleware architecture for pervasive computing systems," Journal of Software, vol. 4, pp. 1061-1068. 2009.

[S186] T. Rafla, P. N. Robillard, and M. Desmarais, "Investigating the impact of usability on software architecture through scenarios: A case study on Web systems," Journal of Systems and Software, vol. 79. pp. 415-426. 2006.

[S187] C. N. Raghvinder Sangwan, Matthew Bass, Zakaria El Houda, "Integrating a software architecture-centric method into object-oriented analysis and design," Journal of Systems and Software, vol. Volume 81, 2008.

[S188] J. A. M. Remo Ferrari, Nazim H. Madhavji, "A controlled experiment to assess the impact of system architectures on new system requirements," in Requirements Engineering, vol. Volume 15.

[S189] H. Reza, "A methodology for architectural design of concurrent and distributed software systems," Journal of Supercomputing, vol. 37, pp. 227-248, 2006.

[S190] H. v. V. Rik Farenhorst, "Understanding How to Support Architects in Sharing Knowledge," in SHARK'09, 2009.

[S191] C. Riva, "Reverse architecting: an industrial experience report," presented at Reverse Engineering, 2000. Proceedings. Seventh Working Conference on, 2000.

[S192] B. S. Roshanak Roshandel*, Nenad Medvidovic*, David Garlan†, Dehua Zhang†, "Understanding Tradeoffs among Different Architectural Modeling Approaches" presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04, 2004.

[S193] T. Rötschke, "Re-engineering a Medical Imaging System Using Graph Transformations," in Lecture Notes in Computer Science. vol. Volume 3062, 2004, pp. Pages 185-201.

[S194] B. Roy and T. C. N. Graham, "An Iterative Framework for Software Architecture Recovery: An Experience Report," in Lecture Notes in Computer Science,Software Architecture, vol. Volume 5292, 2008, pp. Pages 210-224.

[S195] J. P. S. Kim, Y.-B. Kwon, "An Embedded OCR Software Architecture for Enhancing Portability," presented at ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition -, 2007.

[S196] K. S. a. N. D. a. H. Safyallah, "An Orchestrated Multi-view Software Architecture Reconstruction Environment" presented at Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06), 2006.

[S197] L. D. Sam Michiels, Wouter Joosen, Pierre Verbaeten, "The DiPS+ Software Architecture for Self-healing Protocol Stacks," presented at Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[S198] M. B. Simonetta Balsamo and M. Simeoni, "Performance Evaluation at the Software Architecture Level," in Lecture Notes in Computer Science,Formal Methods for Software Architectures, vol. Volume 2804, 2003, pp. Pages 207-258.

[S199] K. Smolander, "What is included in software architecture? A case study in three software organizations," presented at Engineering of Computer-Based Systems, 2002. Proceedings. Ninth Annual IEEE International Conference and Workshop on the, 2002.

[S200] D. Soni, R. L. Nord, and C. Hofmeister, "Software Architecture in Industrial Applications," presented at Software Engineering, 1995. ICSE 1995. 17th International Conference on, 1995.

[S201] C. Stoermer, L. O'Brien, and C. Verhoef, "Architectural views through collapsing strategies," presented at Program Comprehension, 2004. Proceedings. 12th IEEE International Workshop on, 2004.

[S202] C. Stringfellow, C. D. Amory, D. Potnuri, A. Andrews, and M. Georg, "Comparison of software architecture reverse engineering methods," Information and Software Technology, vol. 48, pp. 484-497, 2006.

[S203] N. Subramanian and L. Chung, "Relationship between the whole of software architecture and its parts: an NFR perspective," presented at Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on, 2005.

[S204] A. A. Subrina Anjum Tonu, Ladan Tahvildari, "Evaluating Architectural Stability Using a Metric-Based Approach" presented at Proceedings of the Conference on Software Maintenance and Reengineering (CSMR'06), 2006.

[S205] A. Sutcliffe. "The socio-economics of software architecture," Automated Software Engineering, vol. 15, pp. 343-363, 2008.

[S206] J. v. K. Sylvia Stuurman, "On-line change mechanisms: the software architectural level," presented at SIGSOFT '98/FSE-6: Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering, 1998.

[S207] A. Tang, M. A. Babar, I. Gorton, and J. Han, "A survey of architecture design rationale," Journal of Systems and Software, vol. 79, pp. 1792-1804, 2006.

[S208] A. Tang and H. v. Vliet, "Software Architecture Design Reasoning," in Software Architecture Knowledge Management, 2009, pp. Pages 155-174,Part 3.

[S209] E. M. D. A. v. d. H. R. N. Taylor, "An Infrastructure for the Rapid Development of XM L-based Architecture Description Languages," ICSE'OZ,, 2002.

[S210] A. A. Terekhov, "Dealing with architectural issues: a case study," in SIGSOFT Software Engineering Notes, vol. Volume 29, 2004.

[S211] E. V. M. Tien N. Nguyen, John T. Boyland, and Cheng Thao, "Architectural Software Configuration Management in Molhado." presented at Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04), 2004.

[S212] M. Tivoli and P. Inverardi, "Failure-free coordinators synthesis for component-based architectures," Science of Computer Programming, vol. 71, pp. 181-212, 2008.

[S213] R. T. Tvedt, P. Costa, and M. Lindvall, "Does the code match the design? A process for architecture evaluation." presented at Software Maintenance. 2002. Proceedings. International Conference on, 2002.

[S214] R. T. Tvedt, M. Lindvall, and P. Costa, "A process for software architecture evaluation using metrics," presented at Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard/IEEE, 2002.

[S215] H. Unphon, "Making Use of Architecture throughout the Software Life Cycle - How the Build Hierarchy can Facilitate Product Line Development " SHAR K'09, 2009.

[S216] H. Unphon and Y. Dittrich, "Software architecture awareness in long-term software product evolution," Journal of Systems and Software, vol. 83, pp. 2211-2226.

[S217] T. J. van Beek and T. Tomiyama, "Connecting Views in Mechatronic Systems Design, a Function Modeling Approach," presented at Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on, 2008.

[S218] P. L. Viktor Clerc, Hans van Vliet, "The Usefulness of Architectural Knowledge Management Practices in GSD" in Fourth IEEE International Conference on Global Software Engineering, 2009.

[S219] W.-L. Wang, D. Pan, and M.-H. Chen, "Architecture-based software reliability modeling," Journal of Systems and Software, vol. 79, pp. 132-146, 2006.

[S220] W.-L. Wang, M.-H. Tang, and M.-H. Chen, "Software architecture analysis - a case study," presented at Proceedings of the 1999 23rd Annual International Computer Software and Applications Conference (COMPSAC '99), October 27, 1999 - October 29, 1999, Phoenix, AZ, USA, 1999.

[S221] K. M. H. Weishan Zhang, Joao Fernandes, "Towards OpenWorld Software Architectures with Semantic Architectural Styles, Components and Connectors," presented at ICECCS '09: Proceedings of the 2009 14th IEEE International Conference on Engineering of Complex Computer Systems, 2009.

[S222] T. Werner, C. Vetter, and E. Svensson, "Integrating heterogeneous IT systems: a standards based approach using SOAP," presented at Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on, 2005.

[S223] D. Weyns. "A Pattern Language for Multi-Agent Systems" in IEEE/IFIP WICSA/ECSA, 2009.

[S224] D. Weyns and T. Holvoet, "A reference architecture for situated multiagent systems," presented at 3rd International Workshop on Environments for Multi-Agent Systems, E4MAS 2006, May 8, 2006 - May 8, 2006, Hakodate, Japan, 2007.

[S225] D. Winkler, S. Biffl, and M. Ali Babar, "An empirical investigation of scenarios gained and lost in architecture evaluation meetings," presented at 2nd International Symposium on Empirical Software Engineering and Measurement, ESEM 2008, October 9, 2008 - October 10, 2008, Kaiserslautern, Germany, 2008.

[S226] B. Xu, X. Yang, Y. Shen, S. Li, and A. Ma, "A role-based SOA architecture for community support systems," presented at 2008 International Symposium on Collaborative Technologies and Systems, CTS'08, May 19, 2008 - May 23, 2008, Irvine, CA, United states, 2008.

[S227] Z. Xulin and Z. Ying, "A Business Process Driven Approach for Generating Software Architecture," presented at Quality Software (QSIC), 2010 10th International Conference on.

[S228] Z. D. Yujian Fu1, Xudong He3, "A Method for Realizing Software Architecture Design," presented at Proceedings of the Sixth International Conference on Quality Software (QSIC'06), 2006.

[S229] P. L. Yuxin Wang, He Guo, Han Li, Xin Chen, "Improved Hierarchical Clustering Algorithm for Software Architecture Recovery," presented at ICICCI '10: Proceedings of the 2010 International Conference on Intelligent Computing and Cognitive Informatics.

[S230] G. Zayaraz, S. Vijayalakshmi, and V. Vijayalakshmi, "Evaluation of software architectures using multicriteria fuzzy decision making technique," presented at Intelligent Agent & Multi-Agent Systems, 2009. IAMA 2009. International Conference on, 2009.

[S231] H. Zhang and K. Ben, "Architectural design of the health watch system with an integrated aspect-oriented modeling approach," presented at 2010 International Conference on Computer Design and Applications, ICCDA 2010, June 25, 2010 - June 27, 2010, Qinhuangdao, Hebei, China.

APPENDIX D: INITIAL VERSION OF PROTOCOL

Title: Protocol of Systematic Literature Review of Empirical Evidence in Software Architecture

- Internal advisor

Dr. Naveed Ikram, Associate Professor at International Islamic University Islamabad, Pakistan.

naveed.ikram@iiu.edu.pk

- External Reviewer

Dr. M. A. Babar, Associate Professor at IT University of Copenhagen, Denmark.

malibaba@itu.dk

Appendix D

Systematic Literature Review of Empirical Evidence in Software Architecture

1. Preamble

Software Architecture acts as the skeleton for the software development. It is the artifact that needs to be created early during the software development and then the whole development process revolves around this skeleton, keeping into account the constraints and facilities implied by the software architecture. Few decades back there was nothing like architecture but increasing complexity of software and software quality needs urged the practitioners to opt modularity and ultimately it turned into the form which is now called software architecture. It is the software architecture that is responsible for incorporating quality in software by accommodating quality attributes and functional requirements. Moreover software architecture must have to accommodate the continuous changing requirements so it should be flexible enough to evolve. Academia and industry both are well aware of the importance of software architecture that is why there exists lots of literature on various sub areas of software architecture. 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. This is the reason why I undertook this systematic literature review. My inspiration for such kind of work was SLR conducted in the field of GSD by (Šmite et al. 2010). This document provides an outline of the protocol for my SLR and it is developed based on the guidelines of (Kitchenham. 2007).

2. Background

The main motive to undertake this systematic review is to identify gaps and commonalities in empirical research related to software architecture and summaries the existing empirical evidence to form a stepping-stone for future research and for 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.

Garlan in (1995) conducted a study and listed the active research areas of software architecture and pointed out future directions (Garlan. 1995).

Shaw in (2001) focused the progress of software architecture and the level of maturity of the field. She elaborated the development of software architecture in terms of maturity model discussed in (Redwine and Riddle. 1985). Then she listed the types of questions and research paradigms used with an objective to assess the way research was designed and carried out in the field of software architecture (Shaw. 2001).

Kruchten et al. in (2006) conducted a study that captured the articles containing innovative and highlighted methods, tools and techniques. This study described history of software architecture chronologically listing the research papers, books, conferences and authors in the field of software architecture (Kruchten et al. 2006).

Shaw & Clement in (2006) conducted a comprehensive survey of the software architecture field. This work examined the software architecture's growth in the context of a technology maturation model with matching field's significant accomplishments to various stages of the model, with a perspective to find out where the field stands today (Shaw & Clement. 2006).

Chen et al. in (2010) examined the advancement and achievements made in the 10 years in the field of software architecture and discussed the future research directions as well. The research methodology used was literature survey (Chen et al. 2010).

The research methodology used in most of the studies is literature survey by selecting studies upon author's opinion. 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

(Falessi et al. 2010). There is much work that points towards the need to systematically gather empirical evidence in software Engineering (Dybå et al. 2005, Oates. 2003); so conducting the research using a systematic and unbiased methodology is very important. To the best of my knowledge no systematic literature review in the area of software architecture with a focus on empirical evidence exists.

3. 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 architecture 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.

RQ1: what is the state-of-the-art in empirical studies of software architecture?

The purpose of this question is to evaluate the status of the software architecture 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 architecture along with other relevant information in terms of quantity of the studies.

RQ2: what is the strength of empirical evidence reflected in empirical software architecture 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 my questions will be assessing the future research scope by evaluating and aggregating the available literature.

4. Structured Questions:

RQ1: What is the state-of-the-art in empirical studies of software architecture?

Population: software projects

Intervention: No

Comparison: No

Outcome: status of the software architecture field

RQ2: What is the strength of empirical evidence reflected in empirical software architecture literature?

Population: software projects

Intervention: No

Comparison: No

Outcome: strength of empirical literature

5. Search Strategy

o Strategy for Search terms

The search strategy has been adapted from (Beecham et al. 2006). The steps for extracting search terms are as;

- Derive major search strings.
- Identify alternative spellings and synonyms for major terms.
- When database allows, use the Boolean OR to incorporate alternative spellings and synonyms.
- When database allows, use the Boolean AND to link the major terms.

Step1: Major Search Terms:

Software Architecture

Empirical

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

(Software OR System)

(Architecture OR Structure OR Design)

(Empirical OR Industrial OR Case study OR Experiment OR Experience Report OR Lesson learned)

In order to answer the stated research questions search strategy has already been defined before conducting the review. Research articles that are based on empirical evidence with either professional software developers or students are the main focus of this research. Software architecture will only be considered. The final search strings are selected on the basis of experience from the pilot search and consisted of the following terms:

A1—software architecture	B1—empirical
A2—software structure	B2—case study
A3—software design	B3—industrial
A4—system architecture	B4—experiment

A5—system structure	B5—experience
A6—system design	B6—lessons learned

The search string consists of a Boolean expression: (A1 OR A2 OR A3 OR A4 OR A5 OR A6) AND (B1 OR B2 OR B3 OR B4 OR B5 OR B6). The terms selected for search are quite general for both dimensions used in the search (A and B as shown above). It means that a high proportion of papers will be obtained that will be judged as not relevant, but it will limit the risk of not finding the majority of the relevant papers. For string expression I concatenated the term "software" and its synonym with the term "architecture" and its synonyms as without this concatenation search result bring hardware architecture and basic engineering architectural studies as well.

The search strategy contains the following decisions:

Items: Journal articles, workshop papers and conference papers.

Apply search on: Abstract

Language: The papers written in English.

Publication period: Since 1990

Initially we piloted my search with publication period since 1972 after NATO conference but we found that there is not so much empirical work before 1990 so we restricted my search since the year 1990 as before 1990 there were no such architectures as developing now a days.

6. **Resources to be searched**

Following data bases will be searched for the retrieval of studies:

Springerlink

IEEE Explore

ACM Digital library

ScienceDirect

EI Compendex

7. Study selection criteria

- Search Strings will be applied on the above mentioned databases and obtained references will be archived in a Reference Database.
- Duplicates will be removed.
- Then, the titles of studies will be assessed by using the inclusion criteria.
- In the next step the abstracts will be assessed upon inclusion criteria.
- As the inclusion/exclusion criteria is multiphase so the results of each screening phase will be maintained in separate libraries of database. (Like EndNote libraries).
- The papers that are not clearly relevant or irrelevant will be included/excluded in discussion meeting with research supervisor Dr. Naveed Ikram.

8. Study inclusion and Exclusion Criteria

The inclusion and exclusion scheme consists of multiple phases. The objective of this multiphase study selection process is to identify the articles relevant for the objectives of the systematic review. There are no solid boundaries of the software architecture field so I'll try to be specific in study selection; as software Design and software Architecture has blur boundaries in between that's why I'll exclude design level studies from the search literature. The search strings are quite broad and hence it is expected that all studies identified will not be included in the final phase. The template of inclusion/exclusion criteria has been adapted from (Šmite.2010). The detail of these phases is as:

Relevance Analysis Phase	Inclusion Criteria
Selection of studies based on the search	Only English
	Date of publication: 1990—Present
	Only Published work
	Contains the search strings
Screening: Exclusion upon titles	Not editorials, prefaces, discussions, comments, summaries of tutorials, workshops, panels or duplicates
Screening: Exclusion upon abstracts 1	Empirical background
Screening: Exclusion upon abstract 2	Subject: practitioners or students
Discussion Meeting with Research Supervisor	Main focus on software architecture
Relevance Analysis Phase: Exclusion upon full text	Presence of empirical data in the paper
	Originality of empirical evidence(only one inclusion for studies with the same results reported multiple times

	Sufficient focus on software architecture

9. Quality Instrument for assessing validity

Quality Instrument will be used as a support for data analysis and synthesis. This quality checklist will be used to assess the quality of the studies. The Quality instrument consists of 11 criteria. These criteria were used by (Dyba. 2008). These criteria consider three main characteristics that deal with quality of the study. The three characteristics are: rigor, credibility and relevance. The questions that need to be considered are:

- Does the paper present an empirical study?
- Are the aims of the research clearly stated?
- Is the context of the study adequately described?
- Is the research design appropriate to address the aims of the research?
- Is the recruitment strategy appropriate to the aims of the research?
- Is there a control group with which to compare treatments?
- Is the data collected in a way that addressed the research issue?
- Is the data analysis sufficiently rigorous?
- Does the involvement of researcher affect the results ("causing bias")?
- Is there a clear statement of findings?
- Does the study provide value for research or practice?

Each of the question of 11 criteria will be graded on yes/no/partly or '1/0/0.5' scale. The total sum of grades for the 11 criteria will be used to assess the quality of the studies. The detailed questions of these criteria are appended at the end of this document. (Appendix B)

10. Primary Search Documentation

- Search strings will be applied on the selected sources and obtained results will be saved in different folder on the basis of;
  - Database
  - Screening Phases (multiple phases of inclusion/exclusion criteria)
- These categorizations will be implemented by making folders and saving files in these folders.
- Then after assessment of the studies accepted papers will be copied to another folder.

11. Secondary Search Documentation

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

- Search strings will be applied on the selected sources and obtained results will be saved in different folder on the basis of;
  - Database
  - Screening Phases (multiple phases of inclusion/exclusion criteria)

- These categorizations will be implemented by making folders and saving files in these folders.
- Then after assessment of the studies accepted papers will be copied to another folder.

12. Data Extraction

- Data-Extraction will be performed by using extraction forms.
- The extracted data will be entered into the Access database developed.
- Each paper selected for data extraction will be assigned a unique ID
- A general form will obtain generic data about the study like:
    - Title
    - Author(s)
    - Journal / conference
    - Year of publication
    - Publisher
    - Volume
    - Issue
    - Key words
    - Abstract
    - Geographical area

- Date (of conference)
- Date of Review
- Name of Reviewer
- Type of Study [Empirical or Case Study]
- Quality assessment ranking
- Decision Status

Then the data extraction form will extract data specifically relevant to research questions. The data will be extracted with the help of a classification scheme. This classification scheme and data extraction form is adapted from (Šmite. 2010). This scheme captures data regarding;

- Relevance of the study
- Empirical background
- Software Architectural background
- Focus of the study
- Qualitative evaluation of the work

Data Extraction Classification Scheme

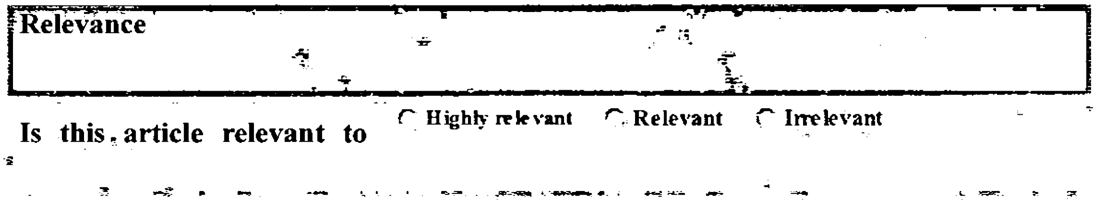
Extracted Data Type	Corresponding Section	Description of Extracted Data
Technical and methodological flaws of the study	Relevance	A study contains empirical evidence
		A study is relevant to software

		architecture
		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 Architectural Background	Architectural styles and Patterns
		Architectural Description and documentation
		Tools
		Software Architecture Evaluation and Analysis
		Architectural Knowledge Management
		Software Architecture Evolution
		Software Architecture Recovery
		Why used/proposed certain

		tool/technique/framework?
Central focus of the study and problem addressed	Study	Focus of the study(practice/tool/technique/ addressing certain problem)
		Evaluation of the study in terms of success/failure
		Application Domain
		Definitions in introduction section
Review of the key results	Qualitative Evaluation	Claims
		Personal Evaluations
		Recommendations

Based upon the above mentioned categorization scheme a sample data extraction form is designed. In this form some fields/questions are given with a set of known answers from the software architecture field. This is just a sample extraction form; certain fields or items of this form can be updated according to the needs if required after piloting the data extraction of few known papers of the software architecture containing empirical evidence.

Data Extraction Form



Relevance

Is this article relevant to ○ Highly relevant ○ Relevant ○ Irrelevant

Software Engineering?	
Is this article relevant to software architecture?	○ Highly relevant ○ Relevant ○ Irrelevant
Is this an empirical study?	○ Yes ○ No
Does this article repeat already reviewed article(s)?	○ Yes ○ No
Empirical Background	
Main Empirical Method	□ Survey □ Case Study □ Interviews □ Controlled Experiment □ Other
Sub-Method	□ Survey □ Case Study □ Interviews □ Archive Analysis □ Controlled Experiment □ Other...
Background	□ Laboratory □ Industry/Real world
Subjects of investigation	□ Students □ Practioners/Researchers
Empirical focus	□ Empirically based □ Empirically evaluated
Software Architectural Background	
Software Architecture styles or Patterns	○ AAL ○ Three-tier ○ Space based SA ○ Three-tier ○ Service Oriented Architecture ○ Three-tier

	○ Three-tier ○ Aspect Oriented Architecture ○ Shared nothing Architecture ○ Event Driven Architecture ○ Pipes and Filters ○ Other
Software Architectural Description/Documentation Language/ Framework	○ AAL ○ DAOP ○ ABC/ADL ○ EAST ○ ArchiMate ○ PiLar ○ ALI ○ xADL ○ XYZ/ADL ○ AC2/ADL ○ PI-ADL ○ Other
Software tool support	○ OSATE ○ DAOP-ADTool ○ DAOP ○ Metaedit ○ systemweaver ○ Other
Software Architecture Recovery	....
Software Architecture Analysis and Evaluation	○ SAAMCS ○ ESAAM ○ SAAMER ○ ATAM ○ SBAR ○ ALPSM ○ SAEM ○ ALRRA ○ SACAM ○ ALMA ○ SARAH ○ Other
Software Architectural Knowledge Management	....
Software Architecture Evolution	....

Why used/proposed certain ....

tool/technique/framework?

Study

Focus of the Study	□ Software Architecture in general □ Single Practice(s)/tool/technique □ Others...
Success or failure?	□ Clear success story □ Success of practices described □ Clear failure story □ Failure of practices described □ Evidence of software architecture related problems □ Unclear □ Other...
Application Domain	□ Telecom □ Automotive □ Web □ Finance □ Automation □ Unclear □ Other...
Definitions in the introduction-like sections?	□ No □ Software Architecture □ System Design □ Software Structure □ System Architecture □ Software Design □ System Structure □ Other related definitions

Qualitative Evaluation

Claims	Narrative

Personal reflection	**Narrative**
Recommendations	**Narrative**

13. Data Analysis and Synthesis

Extracted data will be analyzed using quantitative and qualitative synthesis methods. The research areas in the field of software architecture will be identified along with gaps and future directions. The classification scheme used in data extraction will help here 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 architecture? Either researchers or practitioners who are most involved in software architecture research and in which specific sub area of software architecture? What's the source of empirical evidence etc? this information will be depicted in form of systematic maps like Bar graphs, Bubble plots etc.

14. Validation of Review Process

Protocol Evaluation

- The protocol will be initially evaluated by research supervisor Dr. Naveed Ikram.
- It would then be sent for external evaluation to independent reviewers.

- Protocol will be updated into final version after comments from external reviewers.

Pilot Testing

- Initially testing will be done for different sources to check the validity of search term results.
- Few papers will be selected on the basis of inclusion and exclusion criteria.
- Data will be extracted by using data extraction form so that it can be verified whether extracted data is appropriate for answering the research questions.

15. References

1. Šmite. D, Wohlin. C, Gorschek. T & Feldt. R, (2010). Empirical evidence in global software engineering: a systematic review. Empir Software Eng 15:91–118
2. Kitchenham B (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering, version 2.3. EBSE Technical Report
3. Garlan. D. (1995). Research directions in software architecture. ACM Computing Surveys, Vol. 27, No. 2
4. Shaw. M. (2001). Coming-0f-age of software architecture research. Institute of software research, CMU. Pittsburgh, Pa.

5. Redwine S. and Riddle W. (1985). Software technology maturation. Proceedings of 8th International Conference on Software Engineering, pp. 189-200.

6. Kruchten. P., Obbink. H. & Stafford. J. (2006). The past, present and future of software architecture. IEEE Software.

7. Shaw. M & Clements. P. (2006). The golden age of software architecture. IEEE Software.

8. Chen. Y., Li. X., Yi. L., Liu. D., Tang. L. & Yang. H. (2010). A ten-year survey of software architecture. College of computer science, BUT. Beijing, China.

9. Falessi D., Babar A., Cantone G. & Kruchten P. (2010). Applying empirical software engineering to software architecture: challenges and lessons learned. Empir Software Eng (2010) 15:250–276.

10. Dybå T, Kitchenham B, Jørgensen M (2005) Evidence-based software engineering for practitioners. IEEE Software 22(1):58–65.

11. Oates BJ (2003) Widening the scope of evidence gathering in software engineering. Proceedings of the Eleventh Annual International Workshop on Software Technology and Engineering Practice. IEEE Computer Society

12. Beecham. S, Baddoo. N, Hall. T, Robinson. H, Sharp. H. (2006). Protocol for a Systematic Literature Review of Motivation in Software Engineering.

13. Dyba T and Dingsoyr T. (2008). Empirical studies of agile software development: A systematic review. Inf. Softw. Technol.50, (9-10), pp. 833-859

Appendices

Appendix A: Search Queries

Note: There may be a chance of changes in the syntax of the query strings in accordance with different databases.

ACM Search Query

(Abstract:" software architecture" OR Abstract:"software structure" OR Abstract:"software design" OR Abstract:"system architecture" OR Abstract:"system structure" OR Abstract:"system design") and (Abstract:"case study" OR Abstract:"experience report" OR Abstract:"lesson learned" OR Abstract:"empirical" OR Abstract:"industrial" OR Abstract:"experiment")

Science Direct Search Query

pub-date > 1990 and abstract({Software Architecture} OR {Software structure} OR {Software design} OR {System Architecture} OR {System structure} OR {System design}) and abstract(empirical OR industrial OR experiment OR {case study} OR {experience report} OR {lesson learned}) [All Sources(Computer Science)]

IEEE Search Query

("Abstract":"software architecture" OR "Abstract":"software structure" OR "Abstract":"software design" OR "Abstract":"system architecture" OR "Abstract":"system structure" OR "Abstract":"system design") **AND**

("Abstract":"empirical" OR "Abstract":"case study" OR "Abstract":"experiment" OR "Abstract":"industrial" OR "Abstract":"experience report"" OR "Abstract":"lessons learned")

Compendex Search Query

({software Architecture} OR {Software structure} OR {software design} OR {system Architecture} OR {System structure} OR {system design})) AND (Empirical OR Industrial OR Experiment OR {Case study} OR {Experience Report} OR {Lessons learned})

English

From year 1990 – 2011

Appendix B: Quality Assessment Criteria

Screening questions

1. Is this a research paper containing empirical evidence? Consider: –Is the paper based on empirical research? –Does the study present empirical data?	○ Yes ○ No
2. Is there a clear statement of the aims of the research? Consider:	○ Yes

–Is there a rationale for why the study was undertaken? –Is the study's focus or main focus on Software Architecture? –Is there a clear statement of the study's primary outcome?	○ No
3. Is there an adequate description of the context in which the research was carried out? Consider whether the researcher has identified: –The industry in which products are used (e.g. banking, telecommunications, consumer goods, travel, etc) –The nature of the software development organization (e.g. in-house department or independent software supplier) –The skills and experience of software staff (e.g. with a language, a method, a tool, an application domain) –The type of software products used (e.g. a design tool)	○ Yes ○ No

Detailed questions:

Research design 4. Was the research design appropriate to address the aims of the	○ Yes

research? Consider: – Has the researcher justified the research design (e.g. have they discussed how and why they decided which methods to use)?	○ No ○ Partly
Sampling 5. Was the recruitment strategy appropriate to the aims of the research? Consider: –Has the researcher explained how the participants or cases were identified and selected? –Are the cases defined and described precisely? –Were the cases representative of a defined population? –Have the researchers explained why the participants or cases they selected were the most appropriate to provide access to the type of knowledge sought by the study? –Was the sample size sufficiently large?	○ Yes ○ No ○ Partly
Control group	

6. Was there a control group with which to compare treatments? Consider: –How were the controls selected? –Were they representative of a defined population? –Was there any thing special about the controls? –Was the no n-response high? Could non-respondents be different in any way?	○ Yes ○ No ○ Partly
Data collection 7. Was the data collected in a way that addressed the research issue? Consider: –Were all measures clearly defined (e.g. unit and counting rules)? –Is it clear how data was collected (e.g. semi-structured interviews, focus group etc.)? –Has the researcher justified the methods that were chosen? –Has the researcher made the methods explicit (e.g. is there an indication of how interviews were conducted, did they use an interview guide)?	○ Yes ○ No ○ Partly

–If the methods were modified during the study, has the researcher explained how and why? –Whether the form of the data is clear (e.g. tape recording, video material, notes etc.) –Whether quality control methods were used to ensure completeness and accuracy of data collection?	
Data analysis 8. Was the data analysis sufficiently rigorous? Consider: –Was there an in-depth description of the analysis process? –If thematic analysis was used, is it clear how the categories/ themes were derived from the data? –Has sufficient data been presented to support the findings? –To what extent has contradictory data been taken into account? –Whether quality control methods were used to verify the results?	○ Yes ○ No ○ Partly
Reflexivity (research partnership relations/recognition of	

researcher bias) 9. Has the relationship between researcher and participants been considered adequately? Consider: –Did the researcher critically examine their own role, potential bias and influence during the formulation of research questions, sample recruitment, data collection, and analysis and selection of data for presentation? –How the researcher responded to events during the study and whether they considered the implications of any changes in the research design?	○ Yes ○ No ○ Partly
Findings 10. Is there a clear statement of findings? Consider: –Are the findings explicit (e.g. magnitude of effect)? –Has an adequate discussion of the evidence, both for and against the researcher's arguments, been demonstrated? –Has the researcher discussed the credibility of their findings (e.g. triangulation, respondent validation, more than one analyst)?	○ Yes ○ No ○ Partly

–Are limitations of the study discussed explicitly? –Are the findings discussed in relation to the original research questions? –Are the conclusions justified by the results?	
Value of the research 11. Is the study of value for research or practice? Consider: –Does the researcher discuss the contribution the study makes to existing knowledge or understanding (e.g. do they consider the findings in relation to current practice or relevant research-based literature)? –Does the research identify new areas in which research is necessary? –Does the researcher discuss whether or how the findings can be transferred to other populations, or consider other ways in which the research can be used?	○ Yes ○ No ○ Partly