

Empirical Evidence in Software Architecture: A Systematic Literature Review Protocol

Nadia Qureshi, Naveed Ikram, Muneera Bano, Muhammad Usman

International Islamic University Islamabad, Pakistan

nadia_iiu@yahoo.com, {naveed.ikram, muneera, m.usman} [@iiu.edu.pk](mailto:iiu.edu.pk)

Abstract—Software Architecture (SA) plays important role in software development as it acts as a skeleton and the whole development revolves around it. As the SA as a discipline is maturing, large number of empirically supported studies are being reported in SA. There is a need to systematically aggregate, analyze and synthesize evidence based studies in SA. We plan to systematically investigate evidence-based SA studies to see and report state of the art in evidence based SA reported research. This paper aims at providing a brief description of systematic literature review (SLR) protocol to describe a process for synthesizing the empirically supported work in the area of SA. Protocol for this review has already been developed and its implementation is in progress. Expected outcome of this review will be state-of-the-art of empirical work in the field of software architecture, strength and effectiveness of empirical work, best practices and future research directions.

Keywords—*systematic literature review; software architecture state-of-the-art.*

I. INTRODUCTION

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 [1], then this concept was enhanced and structure of software was emphasized [2] [3]. 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. 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. Academia and industry both are well aware of the importance of software architecture that is why there exists lots of empirical 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, to identify gaps, scope for further research and quality of the work. This is the reason to undertake this systematic literature review. This document provides an outline of the protocol for this systematic literature review and it is developed based on the guidelines of Kitchenham [4].

The remaining portion of the paper consists of following sections; Section II describes motivation and background, Section III explains the outline of research methodology and Section IV concludes the paper along with future work.

II. BACKGROUND

The main motive to undertake this systematic review is to identify all empirical research related to software architecture, aggregate the empirical studies and summaries the evidence for future use. Focus of this SLR is limited to aggregate empirically supported literature i.e. literature based upon some evidence (case study, experiment, experience report and lesson learned etc such studies are also called evidence-based studies). Evidence-based literature is more valuable than literature based upon authors' personal opinion. 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. The studies, those reported state-of-the-art in software architecture, did not performed qualitative and quantitative evaluation of empirical data at a time.

The concept of software architecture as a separate discipline started to emerge in 1990 [5] [6] and developed later on [7]. Since then, the key research areas of software architecture and its future directions have been identified time to time by conducting informal literature reviews and surveys [5] [8] [9]. 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 [10] [11]. The chronological history of the software architecture field, its innovative methods, tools, techniques, software architecture community, papers, books and conferences has already been aggregated [5].

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 literature. Evaluating empirical evidence is equally important for academia and software industry, as systematically gathering and summarizing empirical evidence will help researchers in future and practitioners will also get quantified measures to make informed decisions [12]. There is much work that points towards the need to systematically gather empirical evidence in software Engineering [13] [14]; so conducting the research using a systematic and unbiased methodology is

necessary. Mapping study [15] and many systematic literature reviews exist in the field of software architecture [16] [17] [18]. 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.

III. SYSTEMATIC LITERATURE REVIEW PROTOCOL

Systematic Literature Review is a form of secondary study and it is an established methodology used for the identification, analysis and interpretation of available literature relevant to the research question [4]. Systematic literature Review does not result in novel ideas, it is a fair and repeatable methodology for evaluation of the existing evidence [4]. There are three steps of a systematic literature review process i.e., planning, conducting and reporting [4]. This section will explain the outline of systematic literature review planning phase. Systematic review protocol is the outcome of this 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 [4].

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 the needs 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 main motive behind this research question is to find out the current state of the software architecture field in terms of existing empirical studies and to extract some future guidelines from the existing empirical literature. 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?

Objective behind this question is to find out the effectiveness and 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 [4]. "Assessing the frequency or rate of project development factor such as the adoption of a technology or the frequency of project success or failure" And "identify and/or scope future research activities". So the questions of this SLR will

be assessing the future research scope by evaluating and aggregating the available literature.

A. Search Strategy

- Identify Major Search Terms

Search criteria used to construct major search terms is as follows

- Derive major terms from Research Questions;*
- Software architecture, empirical*
- Find alternative spellings and synonyms of major terms;*
- Software OR System*
- Architecture OR Structure OR Design*
- Empirical OR Industrial OR Case study OR Experiment OR Experience Report OR Lesson learned*
- Use Boolean Operators 'AND' and 'OR' to concatenate search terms if these operators are allowed to be used in the search database strings. Use 'OR' operator to concatenate synonyms of the search terms while use 'AND' to concatenate major search terms.*
- ((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))*

- Resources to be Searched: Springerlink, IEEE Explore, ACM Digital library, ScienceDirect, EI Compendex

- Search Constraints

Search is limited to published studies related to research questions. Search will be applied on conferences papers, journal articles, and workshop papers. This review will consider the work in English and since 1972 (After Parnas work on software structure and decomposition [2] [3]).

B. Publication Selection

- Inclusion Criteria

Research articles based on empirical evidence related to software architecture will be included with either professionals or students as subjects of investigation. Only one instance will be included if multiple studies report same empirical results.

- Exclusion Criteria

Editorials, prefaces, discussions, comments, summaries of tutorials, workshop brief, panels and duplicate studies will be excluded. Studies with insufficient focus on software architecture or with absence of empirical data will be excluded

- Selecting Primary Studies

Search Strings will be applied on the databases and obtained references will be archived in a Reference library. Duplicates will be removed. In the first phase titles of studies will be assessed upon inclusion exclusion criteria. In the next phase the abstracts will be reviewed and after that full text of the selected studies will be

assessed upon inclusion exclusion criteria. As the inclusion/exclusion criteria is multiphase so the results of each screening phase will be maintained in separate libraries. The papers that are not clearly relevant or irrelevant will be included or excluded in discussion meeting with secondary researcher/research supervisor.

C. Publication 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; a 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, experiment and experience report. These criteria are based upon SLR guidelines [4], along with revised set of items adopted from various checklists that have already been used [19] [20] [21] [22] [23]. This checklist generation was a mutual group effort and we are using it in other similar studies [27] as well. The detailed checklist is in Table I.

TABLE I. QUALITY CHECKLIST ADOPTED FROM [4] [19] [20] [21] [22] [23]

Quality Checklist	
<i>Generic</i>	
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
<i>Survey</i>	
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
Have "drop outs" introduced biasness on result limitation?	YES/NO/NOT APPLICABLE
<i>Experiment</i>	
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

Quality Checklist	
<i>Generic</i>	
Are the variables used in the study adequately measured (i.e. are the variables likely to be valid and reliable)?	YES/NO
<i>Case Study</i>	
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
<i>Experience Report</i>	
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

Some of the checklist items will be graded on yes/no and few with partially. Scores will also be assigned according the grades, 1 for yes, 0 for No and 0.5 for partially. The total sum of the scores will be used for the quality assessment of studies.

D. Data Extraction Strategy

Data-Extraction will be performed by using extraction forms. Each paper selected for data extraction will be assigned a unique ID. A general form will obtain generic data about the studies like title of the study, author(s) name, year of publication, journal/conference name etc. 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 is adapted from [24]. The extracted data for research questions is as:

For RQ1 data extraction form will extract following information:

- Software Architecture area (Software Architecture design, Software Architecture Documentation and Specification, Software Architecture Analysis, Software Architecture Evolution, Software Architecture Knowledge Management etc)
- Research output (New Tool/Technique/Process, Modification of Tool/Technique/Process, Usage Experience of Tool/Technique/Process, Software Architecture issues and Challenges).
- Subjects of investigation (Academia, Industry, Mixed)
- Country (involved in research)
- Conference/ Journal
- Year of Publication

For RQ2 the extracted information is as follows:

- Type of evidence (case study, experiment, experience report etc)
- Data collection method (interview, questionnaire etc)
- Type of research: Data about type of research will be extracted based upon an existing classification of research [25] [26] i.e. validation research, evaluation research, solution proposal, philosophical papers, opinion papers and experience papers.

E. Data Synthesis Strategy

Data Extracted from selected literature will be analyzed using quantitative and qualitative synthesis methods. 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 and to draw certain patterns. The expected outcomes will contain information like:

- Publication chronology of included studies
- Distribution of included studies in publication channels along with most cited studies
- List of best practices
- Percentage of studies for different research and evidence types
- Analysis of evidence type versus participant type
- Analysis of evidence type versus SA area type.
- Analysis of evidence type versus type of research
- And more complex analysis comprising more than two parameters.

The quantitative information will be depicted in the form of Bar graphs, Bubble plots etc. A thorough Qualitative analysis of the claims, future directions, recommendations and personal reflections will be performed to draw certain research patterns, future direction and existing gaps.

We are conducting similar studies in other disciplines as well like requirements engineering [27] and using almost the same extraction and synthesis strategies in all studies.

IV. CONCLUSION AND FUTURE WORK

Software Architecture (SA) is maturing into a discipline and has now a long history of research and development. It has its own workshops, conferences and special issues in journals. Large number of empirically supported studies has been published in SA. There lacks a study which presents state of the art of empirically supported work in overall SA discipline. This paper presents the plan for conducting such study i.e. a systematic literature review to present state of the art of empirically supported evidence based SA work. The study will help SA practitioners and researchers to find out mature practices and techniques, patterns/trends in research, gaps and future directions where more emphasis should be placed. The implementation of this systematic literature review protocol is under progress. Search strings returned 5617 results. Screening of the studies upon titles and

abstracts is complete. At present data extraction and quality ranking procedure is under progress.

TABLE II. PUBLICATIONS OBTAINED FROM VARIOUS PUBLICATION CHANNELS.

S. NO.	Publication distribution		
	Publication Channel	No. of studies	%age of studies
1	IEEE	1930	34.35%
2	ACM	1761	31.35%
3	ScienceDirect	259	4.61%
4	SpringerLink	308	5.5%
5	EI Compendex	1359	24.19%
	Sum	5617	100%

REFERENCES

- [1] E. W. Dijkstra, "The structure of the 'THE'-multiprogramming system," in *Proceedings of the first ACM symposium on Operating System Principles*, 1967, pp. 10.1-10.6.
- [2] 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.
- [3] D. L. Parnas, "Information distribution aspects of design methodology," *Methods*, vol. 4, no. 5, pp. 6-7.
- [4] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Engineering*, vol. 2, no. EBSE 2007-001, 2007.
- [5] 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.
- [6] 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.
- [7] D. E. Perry, "State of the Art: Software Architecture," in *International Conference on Software Engineering*, 1997, vol. 19, pp. 590-591.
- [8] D. Garlan, "Research directions in software architecture," *ACM Computing Surveys (CSUR)*, vol. 27, no. 2, pp. 257-261, 1995.
- [9] Y. Chen, X. Li, L. Yi, D. Liu, L. Tang, and H. Yang, "A ten-year survey of software architecture," in *IEEE International Conference on Software Engineering and Service Sciences (ICSESS)*, 2010, pp. 729-733.
- [10] M. Shaw, "The coming-of-age of software architecture research," in *Proceedings of the 23rd international conference on Software engineering*, 2001, p. 656.
- [11] M. Shaw and P. Clements, "The golden age of software architecture," *Software, IEEE*, vol. 23, no. 2, pp. 31-39, 2006.
- [12] 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.
- [13] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *Software, IEEE*, vol. 22, no. 1, pp. 58-65, 2005.
- [14] B. J. Oates, "Widening the scope of evidence gathering in software engineering," in *11th International Workshop on Software Technology and Engineering Practice*, 2003, pp.59-64.

- [15] 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.^h
- [16] R. Farenhorst and R. C. de Boer, "Knowledge management in software architecture: State of the art," *Software Architecture Knowledge Management: Theory and Practice*. Springer, Under submission.
- [17] 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.
- [18] H. P. Breivold, I. Crnkovic, and M. Larsson, "A systematic review of software architecture evolution research," *Information and Software Technology*, 2011, ISSN 0950-5849, 10.1016/j.infsof.2011.06.002. (<http://www.sciencedirect.com/science/article/pii/S0950584911001376>)
- [19] 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, available online http://www.bcs.org/upload/pdf/ewic_ea09_s3paper1.pdf
- [20] B. Kitchenham et al., "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, pp. 1-8.
- [21] 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.
- [22] D. Budgen and C. Zhang, "Preliminary reporting guidelines for experience papers," in *Proceedings of EASE*, 2009, vol. 2009, pp. 1-10.
- [23] T. Dyba and T. Dingsoyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9-10, pp. 833-859, 2008.
- [24] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, vol. 15, no. 1, pp. 91-118, 2010.
- [25] 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.
- [26] 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.
- [27] T. Ambreen, M. Usman, N. Ikram and M. Bano, "Software requirement engineering: A systematic literature review protocol", in *6th International Conference on Software Engineering Advances, 2011., in press.*