# Software Architecture Evaluation: A Systematic Mapping Study

Sofia Ouhbi
*TIC Lab, FIL, International University of Rabat, Rabat, Morocco*

Keywords: Systematic Mapping Study, Software Architecture, Evaluation, Assessment.

Abstract: Software architecture provides the big picture of software systems, hence the need to evaluate its quality before further software engineering work. In this paper, a systematic mapping study was performed to summarize the existing software architecture evaluation approaches in literature and to organize the selected studies according to six classification criteria: research types, empirical types, contribution types, software quality models, quality attributes and software architecture models. Publication channels and trends were also identified. 60 studies were selected from digital libraries.

## 1 INTRODUCTION

Software architecture (SA) started to emerge as a discipline during the mid nineties due to the increasing complexity of software systems which led to increased challenges for software industry (iso, 2011). SA is defined as "the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both" (Clements et al., 2002). The SA highlights early design decisions that will have a tremendous impact on all software engineering work that follows (Bass, 2007). There is therefore a need for SA evaluation (SAE) approaches to minimize the negative impact of low quality SA on software implementation.

This paper presents the results of a systematic mapping study which was performed to obtain an updated overview of the current approaches used in SAE research. Many reviews have been conducted in this area (Babar et al., 2004; Ionita et al., 2002; Maurya, 2010; Bass and Nord, 2012), but to the best of our knowledge, no systematic mapping study of SAE approaches has been published to date. Eight mapping questions (MQs) are answered in this study and the papers which were selected after the search process are classified according to six criteria: research types, empirical types, contribution types, software quality (SQ) models, quality attributes and SA models, in addition to the main publication channels and trends.

The structure of this paper is as follows: Section 2 presents the research methodology. Section 3 reports the results. Section 4 discusses the findings. The conclusions and future work are presented in Section 5.

## 2 RESEARCH METHODOLOGY

The systematic mapping study principal goal is to provide an overview of a research area, and identify the quantity and type of research and results available within it. A mapping process consists of three activities: the search for relevant publications, the definition of a classification scheme and the mapping of publications (Petersen et al., 2008). A mapping study differs from a systematic literature review (SLR) as the articles are not studied in sufficient detail.

### 2.1 Mapping Questions

This study aims to gain insight into the existing SAE approaches. The systematic mapping study therefore addresses eight MQs. The MQs with the rationale motivating the importance of these questions are presented in Table 1. The search strategy and paper selection criteria were defined on the basis of them.

### 2.2 Search Strategy

The papers were identified by consulting the following sources: IEEE Digital Library, ACM Digital Library, Science Direct and SpringLink. Google scholar was also used to seek literature in the field. The search was done in January 2018. The following search string was applied in the title, abstract and keywords of the papers investigated to reduce the search results. "Software architecture" **AND** (evaluat* **OR** measur* **OR** assess*) **AND** (technique* **OR** appro-

Ouhbi, S.
Software Architecture Evaluation: A Systematic Mapping Study.
In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)*, pages 447-454
ISBN: 978-989-758-300-1

Table 1: Mapping questions.

| ID | Mapping question | Rationale |
|---|---|---|
| MQ1 | Which publication channels are the main targets for SAE research? | To identify where SAE research can be found as well as the good targets for publication of future studies |
| MQ2 | How has the frequency of approaches related to SAE changed over time? | To identify the publication trends over time of SAE research |
| MQ3 | What are the research types of SAE studies? | To explore the different types of research reported in the literature concerning SAE |
| MQ4 | Are SAE studies empirically validated? | To discover whether research on SAE has been validated through empirical studies |
| MQ5 | What are the evaluation approaches that were reported in SA research? | To discover the existing SAE approaches reported in the existing SAE literature |
| MQ6 | Were SAE approaches reported in literature based on SQ model? | To discover if researchers take into consideration SQ models in SAE approaches design |
| MQ7 | Which quality attributes were used to evaluate SA? | To identify the quality attributes used to evaluate SA in literature |
| MQ8 | What are the models that were used in SAE literature? | To identify the models used in the SAE literature |

ach* **OR** method* **OR** model* **OR** framework* **OR** tool*).

## 2.3 Paper Selection Criteria

Each paper was retrieved by the author and the information about it was filed in an excel file. The first step after the application of the search string was to eliminate duplicate titles, and titles clearly not related to the review. The inclusion criteria were limited to the studies that address evaluation, measurement or assessment of the SA in overall or through quality attributes. The studies that met at least one of the following exclusion criteria (EC) were excluded:

**EC1** Papers that focus on software design.

**EC2** Papers whose subject was one or many quality characteristics which were not used for SAE.

In total, 217 papers were identified after the removal of duplicates. When the same paper appeared in more than one source, it was considered only once according to our search order. Thereafter, 158 studies were excluded based on the inclusion and exclusion criteria leaving for the final result 60 selected studies.

## 2.4 Data Extraction Strategy

The publication source and channel of the papers selected respond to **MQ1**, while the publication year responds to **MQ2**. A research type (**MQ3**) can be classified in the following categories (Ouhbi et al., 2015): (1) evaluation research: existing SAE approaches are implemented in practice and an evaluation of them is conducted; (2) solution proposal: an SAE solution is proposed. This solution may be a new SAE approach or a significant extension of an existing approach. The potential benefits and the applicability of the solution could be shown with an empirical study or a good argumentation; or (3) other, e.g. experience paper, review. The empirical type of the selected study can be classified for **MQ4** as a (Ouhbi et al., 2013): (1) case study: an empirical inquiry that investigates an SAE approach within its real-life context; (2) survey: a method for collecting quantitative information concerning an SAE approach, e.g. a questionnaire; (3) experiment: an empirical method applied under controlled conditions to evaluate a SAE approach; or (4) theory: non-empirical research approaches or theoretical evaluation of an SAE approach. An approach (**MQ5**) can be classified as (Ouhbi et al., 2014): process, method, tool-based technique, model, framework, data mining technique, or other, e.g. guidelines.

A SQ model (**MQ6**) can be classified as (Ortega et al., 2003): McCall model (Company et al., 1977), Boehm model (Boehm et al., 1978), Dromey model (Dromey, 1996), ISO/IEC 9126 standard (ISO/IEC-9126-1, 2001), ISO/IEC 25010 standard (ISO, 2011), or other. A quality attribute (**MQ7**) can be classified into one of the internal and external quality characteristics proposed by ISO/IEC 25010: Functional suitability, reliability, usability, performance efficiency, maintainability, portability, compatibility, security or other. A SA model (**MQ8**) can be classified as (Vogel et al., 2011): UML, 4+1 view model, an architectural description language (ADL), or other.

# 3 RESULTS

This section describes results presented in Table 2.

Table 2: Classification Acronyms: SP: Solution proposal, ER: Evaluation research, EP: Experience paper, M: Maintainability, S: Security, Pe: Performance, Po: Portability, U: Usability, C: Compatibility, R: Reliability, F: Functional suitability.

| Ref. & Year | Channel | Res. T. | Emp. Res. | Contr. T. | SQ Model | Quality attributes | Model |
|---|---|---|---|---|---|---|---|
| (Abowd et al., 1997) | Other | SP | Theory | Framework | - | F., Pe., other. | - |
| (Barbacci et al., 1997) | Other | SP | Theory | Method | Boehm, Other | R., other | - |
| (Dueñas et al., 1998) | Workshop | SP | Theory | Model | ISO/IEC 9126 | All | ADL, UML |
| (Bergey et al., 1999) | Other | SP | Theory | Process |  | Others. | - |
| (Kazman et al., 2000) | Other | SP | Theory | Method | - | Pe., S., others | - |
| (Van Gurp and Bosch, 2000) | Conference | SP | Case study | DataMining | McCall | M.,R., U., Pe., others | - |
| (Bosch and Bengtsson, 2001) | Conference | SP | Case study | Method | - | M. | - |
| (Ionita et al., 2002) | Workshop | Other | Theory | Method | - | Po., F.,M., U., Pe., others | UML, 4+1 view |
| (Choi and Yeom, 2002) | Conference | SP | Other | Method | - | Pe., R., others | All |
| (Tvedt et al., 2002b) | Conference | SP | Case study | Process | - | S., R., M., others | - |
| (Tvedt et al., 2002a) | Conference | SP | Case study | Process | - | M. | - |
| (Clements, 2002) | Other | ER | Case study | Method | - | Pe., S., R., M., Po., others | - |
| (Lindvall et al., 2003) | Journal | ER | Case study | Process | - | M. | - |
| (Clements et al., 2003) | Other | SP | Theory | Method | - | Pe., R., S., Po., F., others | - |
| (Folmer et al., 2003) | Workshop | SP | Theory | Method | ISO/IEC 9126 | U. | 4+1 view |
| (Barbacci et al., 2003) | Other | ER | Case study | Method | - | Pe., S., R., M., Po., others | 4+1 view |
| (Bahsoon and Emmerich, 2003) | Conference | Other | Theory | Method | - | Others | ADL |
| (Babar et al., 2004) | Conference | Other | Theory | Framework | McCall | M., others | ADL |
| (Zhu et al., 2004) | Conference | SP | Other | Process | McCall, ISO/IEC 9126 | - | - |
| (Subramanian and Chung, 2004) | Conference | SP | Theory | Framework | - | Other | - |
| (Bashroush et al., 2004) | Conference | SP | Theory | Tool | - | - | ADLARS |
| (Maheshwari and Teoh, 2005) | Journal | SP | Theory | Tool | - | S., other | - |
| (Liu and Wang, 2005) | Conference | SP | Theory | Method | ISO/IEC 9126 | Other | - |
| (Gorton and Zhu, 2005) | Conference | EP | Experiment | Tool | - | Others | UML |
| (Graaf et al., 2005) | Conference | ER | Case study | Method | - | M. | UML, 4+1 view |
| (Mattsson et al., 2006) | Conference | Other | Theory | Method | Other | Pe., M., Po., other | - |
| (Babar et al., 2006) | Conference | SP | Experiment | Framework | - | - | - |
| (Mårtensson, 2006) | Other | SP | Case study | Method | - | Pe., M., Po., other | UML |
| (Babar et al., 2007) | Conference | ER | Other | Other | - | M., U., Pe., S. | - |
| (Kim et al., 2007) | Conference | SP | Experiment | Other | - | Pe., R., U., other | - |
| (Jin-hua, 2007) | Journal | SP | Theory | Method | - | others | UML |
| (Svahnberg and Mårtensson, 2007) | Journal | EP | Experiment | Method | ISO/IEC 9126 | - | - |
| (Babar et al., 2008) | Journal | ER | Experiment | Framework | - | R., M., Pe., other | - |
| (Erfanian and Shams Aliee, 2008) | Conference | SP | Case study | Method | - | Pe., S., U., others | UML |
| (Salger et al., 2008) | Conference | SP | Case study | Framework | ISO/IEC 9126 | U., R. | - |
| (Roy and Graham, 2008) | Other | Other | Theory | Method | ISO/IEC 9126, McCall, Boehm | All | All |
| (Ali Babar, 2008) | Conference | ER | Experiment | Framework | - | S. | - |
| (Lee et al., 2009) | Journal | SP | Case study | Method | - | Pe., U., others | - |
| (Duszynski et al., 2009) | Conference | SP | Theory | Tool | - | - | - |
| (Babar, 2009) | Conference | SP | Experiment | Framework | - | M., others | - |
| (Salger, 2009) | Workshop | EP | Theory | Framework | - | - | - |
| (Bergey, 2009) | Other | SP | Other | Plan | - | Pe., S., others | - |
| (Alkussayer and Allen, 2010) | Conference | SP | Theory | Framework | - | S. | - |
| (Maurya, 2010) | Journal | Other | Theory | Method | - | M.,Pe., others | - |
| (Reijonen et al., 2010) | Conference | ER | Experiment | Method | ISO/IEC 9126 | All | - |
| (Martens et al., 2011) | Journal | ER | Experiment | Method | - | Pe. | UML |
| (Shanmugapriya and Suresh, 2012) | Journal | Other | Theory | Method | - | - | 4+1 view |
| (Bass and Nord, 2012) | Conference | Other | Theory | Framework | - | - | - |
| (Akinnuwesi et al., 2012) | Journal | SP | Theory | DataMining | - | Pe., others | - |
| (Sharafi, 2012) | Journal | SP | Other | Method | - | R., Pe., other | UML |
| (Bouwers, 2013) | Other | SP | Survey | Model | ISO/IEC 25010 | M. | UML |
| (Akinnuwesi et al., 2013) | Journal | SP | Experiment | DataMining | - | Pe. | - |
| (Zalewski and Kijas, 2013) | Journal | SP | Case study | Method | ISO/IEC 9126 | M., U., Pe., R., S., others | - |
| (González-Huerta et al., 2013) | Conference | Other | Experiment | Method | ISO/IEC 25000 | - | - |
| (Knodel and Naab, 2014) | Conference | EP | Case study | Method | - | Pe. | UML |
| (Santos et al., 2014) | Workshop | Other | Theory | Method | - | Pe., R., C., others | ADLARS |
| (Patidar and Suman, 2015) | Conference | Other | Theory | Method | - | - | - |
| (Gonzalez-Huerta et al., 2015) | Journal | ER | Experiment | Model | ISO/IEC 25000 | R., Pe. | UML |
| (Upadhyay, 2016) | Journal | SP | Other | Framework | - | M., others | - |
| (Abrahão and Insfran, 2017) | Conference | ER | Experiment | Method | - | R., Pe. | - |

## 3.1 MQ1. Publication Channels

48.3% of the selected papers were presented at conferences, 25% were published in journals, 11.7% were published as technical reports, 8.3% appeared in workshops, 3.3% are books and 3.3% are PhD thesis. 10% of the selected papers were published by Software Engineering Institute as technical reports.

## 3.2 MQ2. Publication Trend

Fig. 1 presents the number of articles published per year from 1997 to 2017.

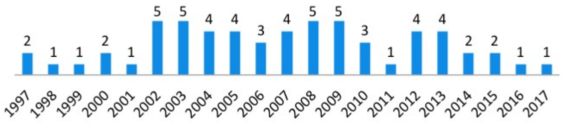

Figure 1: Number of articles published per year.

## 3.3 MQ3. Research Types

Fig. 2 shows the research type of the selected papers. Around 57% of the selected papers were solution proposal studies, 18% of the selected papers were under-

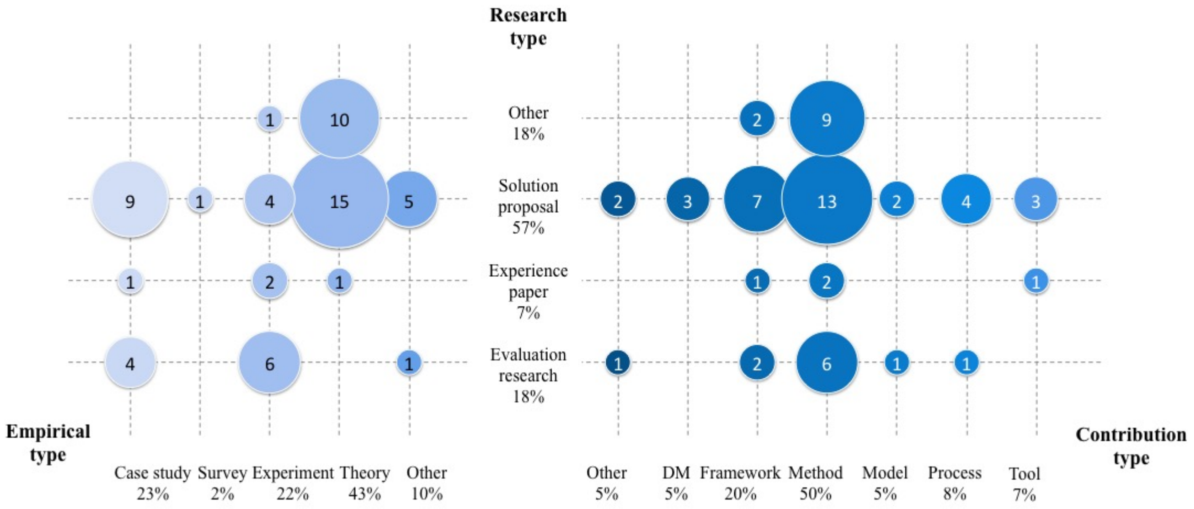

Figure 2: RQ3, RQ4 and RQ5 results.

taken to evaluate SAE existing approaches, 7% were reporting the authors' experience with SAE and the remaining papers were classified as others. Among the other types that we have identified: 8 reviews, 2 position papers (Bahsoon and Emmerich, 2003; Santos et al., 2014) and one replication study (González-Huerta et al., 2013). This result shows that the main concern of researchers in the SAE domain is to propose and develop approaches to enhance SAE. Fig. 2 shows also that 44% of solution proposals were not empirically validated and that 38% of the suggested solutions are methods.

### 3.4 MQ4. Empirically Types

Fig. 2 shows if the selected studies were empirically validated and presents the empirical types used in the validation of SAE approaches. A percentage of 43% of the selected studies were not evaluated empirically. 23% of the selected papers undertook case studies to evaluate SAE approaches and 22% were evaluated with experiments. One paper (Bouwers, 2013) has used a survey and another paper (Babar et al., 2007) used focus group to evaluate SAE approaches while the remaining papers used illustration examples to demonstrate the applicability of their approaches.

### 3.5 MQ5. Contribution Types

Fig. 2 presents the SAE approaches extracted from the selected papers. The approaches most frequently reported are methods (50% of the selected papers) followed by frameworks (20%). Processes, tool-based techniques, models, and data mining techniques were also identified in the selected studies. Other techniques in this study were also identified, such as a AHP technique (Kim et al., 2007).

### 3.6 MQ6. SQ Models

The results shown in Fig. 3 reveal that around 69% of SAE papers do not cite any well-known SQ model. The principal model cited in the selected studies was the ISO/IEC 9126 standard. McCall model and Boehm model were also cited. Note that only one paper (Bouwers, 2013) has cited ISO/IEC 25010 but it has been insinuated in two papers (Gonzalez-Huerta et al., 2015; González-Huerta et al., 2013) as they have mentioned ISO/IEC 25000. Some papers cited different models and standards that helped them in the design of SAE techniques, such as: IEEE 610.12-1990 cited by (Mattsson et al., 2006) and IEEE 1061 cited by (Barbacci et al., 1997).

### 3.7 MQ7. Quality Attributes

15% of SAE papers did not mention any quality attribute. Fig. 4 shows how often a quality attribute has been mentioned in SAE literature. It is worth mentioning that some papers dealt with the evaluation of SA through only a unique quality attribute. Table 3 presents these characteristics.

### 3.8 MQ8. SA Description Models

Around 65% of the studies selected didn't specify any SA model. Fig. 5 presents how often the SA models have been reported in the remaining 21 studies.

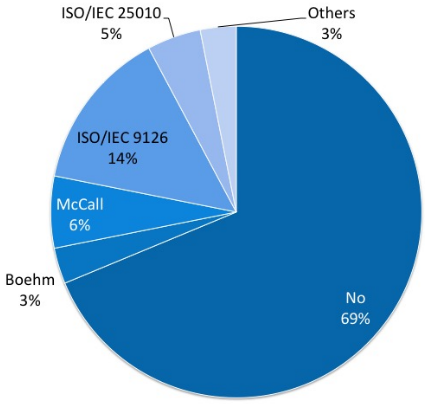

Figure 3: SQ models in SEA literature.

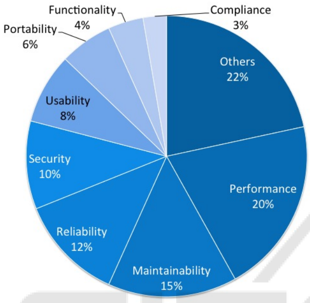

Figure 4: Quality attributes in SEA literature.

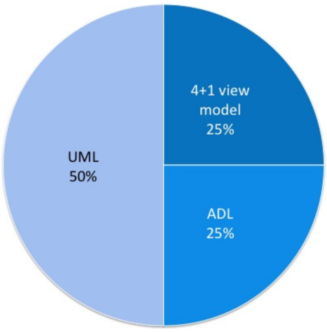

Figure 5: SA models reported in 21 studies.

# 4 DISCUSSION

The interest on SAE began after the publication of technical reports by the Software Engineering Institutes in 1997. This interest was at it most during the last decade where many researchers have based their research on the outcomes of these technical reports mainly on the Architecture Tradeoff Analysis Method (ATAM). However, this interest started to fade since 2014, which indicates that there is a need for novel SAE techniques, particularly with the emerging new technologies such as the IoT and the Big Data (Krco et al., 2014; Gorton and Klein, 2015). The majority of the papers were evaluated using case studies, it is easier to evaluate SA of existing systems rather than developing a system only for the purpose to evaluate its architecture. In fact, SA requires an early software engineering activity which is the specification of stakeholders needs, also known as requirements engineering (Ouhbi et al., 2013). This step is critical to identify the quality attributes that will influence SA design and description. Working with case study reduce the effort required to specify requirements and quality attributes. Around half of SAE selected studies present methods to evaluate SA. The majority of these methods are based on ATAM and few researchers (Clements, 2002; Svahnberg and Mårtensson, 2007; Graaf et al., 2005) based their methods on the Software architecture analysis method (SAAM) (Kazman et al., 1994). SAAM is a method for analyzing the properties of SA and not for SAE, for this reason the study by (Kazman et al., 1994) was not included in this mapping study.

Few researchers based their solutions on SQ models. ISO/IEC 9126 standard was the most used due to the fact that it is the most well-known SQ model during the last decade before it was replaced by ISO/IEC 25010 in 2011. Recent studies have used the ISO/IEC 25010 to analyze SQ requirements and to specify quality attributes. Implementing quality attributes makes it is easier for the software architect to evaluate the quality of SA (Witt et al., 1993). The most quality attribute cited in SAE literature is Performance. This could be explained by the fact that this attribute affects runtime behavior and overall user experience. The main SA model used to describe and evaluate SA is UML, due to the fact that UML is a standardized and popular modeling language know by the software development community. 4+1 view model and ADLs were also cited in few papers, more precisely ADLARS in (Bashroush et al., 2004; Santos et al., 2014) which is a relational ADL for software

Table 3: Papers which have focus on one quality attributes.

| Quality attrib. | Ref. | Total |
|---|---|---|
| Maintainability | (Bosch and Bengtsson, 2001) (Bouwers, 2013) (Graaf et al., 2005) (Lindvall et al., 2003) (Tvedt et al., 2002a) | 5 |
| Performance | (Akinnuwesi et al., 2013) (Knodel and Naab, 2014) (Martens et al., 2011) (Reijonen et al., 2010) | 4 |
| Security | (Ali Babar, 2008) (Alkussayer and Allen, 2010) | 2 |
| Usability | (Folmer et al., 2003) | 1 |
| Adaptability | (Liu and Wang, 2005) | 1 |
| Changeability | (Subramanian and Chung, 2004) | 1 |

families (Brown et al., 2003). All these models are related. In fact, UML is considered as an ADL as it serves to describe SA. Moreover, 4+1 view model uses UML diagrams to describe the logical, process, development, physical and scenario views of a SA.

## 5 CONCLUSION AND FUTURE WORK

The overall goal of this study is to conduct a thematic analysis and identify publication fora as regards SAE approaches. The findings of this systematic map have implications for researchers and practitioners who work in the SA domain, since this study will allow them to discover the existing SAE approaches and techniques in the literature. The presented empirical studies may also provide an overview of the efficiency of each approach. For future work, we intend to conduct an SLR of empirical evidence on SAE.

## REFERENCES

(2011). ISO/IEC 25010 standard. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

(2011). Iso/iec/ieee 42010: 2011-systems and software engineering–architecture description.

Abowd, G., Bass, L., Clements, P., Kazman, R., and Northrop, L. (1997). Recommended best industrial practice for software architecture evaluation. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Abrahão, S. and Insfran, E. (2017). Evaluating software architecture evaluation methods: An internal replication. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 144–153. ACM.

Akinnuwesi, B. A., Uzoka, F.-M. E., Olabiyisi, S. O., and Omidiora, E. O. (2012). A framework for user-centric model for evaluating the performance of distributed software system architecture. *Expert Systems with Applications*, 39(10):9323–9339.

Akinnuwesi, B. A., Uzoka, F.-M. E., and Osamiluyi, A. O. (2013). Neuro-fuzzy expert system for evaluating the performance of distributed software system architecture. *Expert Systems with Applications*, 40(9):3313–3327.

Ali Babar, M. (2008). Assessment of a framework for designing and evaluating security sensitive architecture. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, pages 156–165.

Alkussayer, A. and Allen, W. H. (2010). A scenario-based framework for the security evaluation of software architecture. In *3rd IEEE International Conference on Computer Science and Information Technology*, volume 5 of *ICCSIT*, pages 687–695. IEEE.

Babar, M. A. (2009). A framework for supporting the software architecture evaluation process in global software development. In *Fourth IEEE International Conference on Global Software Engineering*, pages 93–102. IEEE.

Babar, M. A., Bass, L., and Gorton, I. (2007). Factors influencing industrial practices of software architecture evaluation: an empirical investigation. In *Third International Conference on Quality of Software Architectures*, QoSA, pages 90–107. Springer.

Babar, M. A., Kitchenham, B., and Gorton, I. (2006). Towards a distributed software architecture evaluation process: a preliminary assessment. In *Proceedings of the 28th International Conference on Software Engineering*, pages 845–848. ACM.

Babar, M. A., Kitchenham, B., and Jeffery, R. (2008). Comparing distributed and face-to-face meetings for software architecture evaluation: A controlled experiment. *Empirical Software Engineering*, 13(1):39–62.

Babar, M. A., Zhu, L., and Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods. In *Australian Software Engineering Conference*, pages 309–318. IEEE.

Bahsoon, R. and Emmerich, W. (2003). Evaluating software architectures: Development stability and evolution. In *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, Tunis, Tunisia*, pages 47–56. IEEE Computer Society Press.

Barbacci, M., Clements, P. C., Lattanze, A., Northrop, L., and Wood, W. (2003). Using the architecture tradeoff analysis method (ATAM) to evaluate the software architecture for a product line of avionics systems: A case study. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Barbacci, M. R., Klein, M. H., and Weinstock, C. B. (1997). Principles for evaluating the quality attributes of a software architecture. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Bashroush, R., Spence, I., Kilpatrick, P., and Brown, J. (2004). Towards an automated evaluation process for software architectures. In *IASTED international conference on Software Engineering*.

Bass, L. (2007). *Software architecture in practice*. Pearson Education India.

Bass, L. and Nord, R. L. (2012). Understanding the context of architecture evaluation methods. In *Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA)*, pages 277–281. IEEE.

Bergey, J. K. (2009). A proactive means for incorporating a software architecture evaluation in a DoD system acquisition. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Bergey, J. K., Fisher, M. J., Jones, L. G., and Kazman, R. (1999). Software architecture evaluation with ATAM in the DoD system acquisition context. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Boehm, B. W., Brown, J. R., Kaspar, H., and Lipow, M. (1978). *Characteristics of software quality*. TRW Softw. Technol. North-Holland, Amsterdam.

Bosch, J. and Bengtsson, P. (2001). Assessing optimal software architecture maintainability. In *Fifth European Conference on Software Maintenance and Reengineering*, pages 168–175. IEEE.

Bouwers, E. M. (2013). *Metric-based Evaluation of Implemented Software Architectures*. PhD thesis, Electrical Engineering, Mathematics and Computer Science. Delft University of Technology.

Brown, T. J., Spence, I. T., and Kilpatrick, P. (2003). A relational architecture description language for software families. In *International Workshop on Software Product-Family Engineering*, pages 282–295. Springer.

Choi, H. and Yeom, K. (2002). An approach to software architecture evaluation with the 4+ 1 view model of architecture. In *Ninth Asia-Pacific Software Engineering Conference*, pages 286–293. IEEE.

Clements, P. (2002). *Evaluating Software Architecture, Methods and Case Studies*. Addison-Wesley Professional.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., and Little, R. (2002). *Documenting software architectures: views and beyond*. Pearson Education.

Clements, P., Kazman, R., and Klein, M. (2003). *Evaluating software architectures*. Tsinghua University Press Beijing.

Company, G. E., McCall, J. A., Richards, P. K., and Walters, G. F. (1977). *Factors in Software Quality: Final Report*. Information Systems Programs, General Electric Company.

Dromey, R. G. (1996). Cornering the Chimera. *IEEE Software*, 13(1):33–43.

Dueñas, J., de Oliveira, W., and de la Puente, J. (1998). A software architecture evaluation model. *Development and Evolution of Software Architectures for Product Families. 2nd International ESPRIT ARES Workshop*, pages 148–157.

Duszynski, S., Knodel, J., and Lindvall, M. (2009). SAVE: Software architecture visualization and evaluation. In *13th European Conference on Software Maintenance and Reengineering*, CSMR, pages 323–324. IEEE.

Erfanian, A. and Shams Aliee, F. (2008). An ontology-driven software architecture evaluation method. In *Proceedings of the 3rd International Workshop on Sharing and Reusing Architectural Knowledge*, pages 79–86. ACM.

Folmer, E., van Gurp, J., and Bosch, J. (2003). Scenario-based assessment of software architecture usability. In *ICSE Workshop on SE-HCI*, pages 61–68.

González-Huerta, J., Insfrán, E., and Abrahão, S. (2013). On the effectiveness, efficiency and perceived utility of architecture evaluation methods: A replication study. In *18th National Conference in Software Engineering and Databases, Madrid, Spain*, pages 427–440.

Gonzalez-Huerta, J., Insfran, E., Abrahão, S., and Scanniello, G. (2015). Validating a model-driven software architecture evaluation and improvement method: A family of experiments. *Information and Software Technology*, 57:405–429.

Gorton, I. and Klein, J. (2015). Distribution, data, deployment: Software architecture convergence in big data systems. *IEEE Software*, 32(3):78–85.

Gorton, I. and Zhu, L. (2005). Tool support for just-in-time architecture reconstruction and evaluation: an experience report. In *27th International Conference on Software Engineering*, ICSE, pages 514–523. IEEE.

Graaf, B., Van Dijk, H., and Van Deursen, A. (2005). Evaluating an embedded software reference architecture-industrial experience report. In *Ninth European Conference on Software Maintenance and Reengineering*, CSMR, pages 354–363. IEEE.

Ionita, M. T., Hammer, D. K., and Obbink, H. (2002). Scenario-based software architecture evaluation methods: An overview. In *Workshop on methods and techniques for software architecture review and assessment at the international conference on software engineering*, pages 19–24.

ISO/IEC-9126-1 (2001). Software engineering – Product quality – Part 1: Quality models.

Jin-hua, L. (2007). Uml based quantitative software architecture evaluation. *Journal of Chinese Computer Systems*, 6:017.

Kazman, R., Bass, L., Abowd, G., and Webb, M. (1994). SAAM: A method for analyzing the properties of software architectures. In *16th International Conference on Software Engineering*, ICSE, pages 81–90. IEEE.

Kazman, R., Klein, M., and Clements, P. (2000). Atam: Method for architecture evaluation. Technical report, Softwrae Engineering Institute. Carnegie-Mellon University.

Kim, C.-K., Lee, D.-H., Ko, I.-Y., and Baik, J. (2007). A lightweight value-based software architecture evaluation. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, volume 2 of *SNPD*, pages 646–649. IEEE.

Knodel, J. and Naab, M. (2014). Software architecture evaluation in practice: Retrospective on more than 50 architecture evaluations in industry. In *IEEE/IFIP Conference on Software Architecture*, WICSA, pages 115–124. IEEE.

Krco, S., Pokric, B., and Carrez, F. (2014). Designing IoT architecture(s): A european perspective. In *IEEE World Forum on Internet of Things (WF-IoT)*, pages 79–84. IEEE.

Lee, J., Kang, S., and Kim, C.-K. (2009). Software architecture evaluation methods based on cost benefit analysis and quantitative decision making. *Empirical Software Engineering*, 14(4):453–475.

Lindvall, M., Tvedt, R. T., and Costa, P. (2003). An empirically-based process for software architecture evaluation. *Empirical Software Engineering*, 8(1):83–108.

Liu, X. and Wang, Q. (2005). Study on application of a quantitative evaluation approach for software architecture adaptability. In *Fifth International Conference on Quality Software*, QSIC, pages 265–272. IEEE.

Maheshwari, P. and Teoh, A. (2005). Supporting atam with a collaborative web-based software architecture evaluation tool. *Science of Computer Programming*, 57(1):109–128.

Martens, A., Koziolek, H., Prechelt, L., and Reussner, R. (2011). From monolithic to component-based performance evaluation of software architectures. *Empirical Software Engineering*, 16(5):587–622.

Mårtensson, F. (2006). *Software Architecture Quality Evaluation: Approaches in an Industrial Context*. PhD thesis, Blekinge Institute of Technology.

Mattsson, M., Grahn, H., and Mårtensson, F. (2006). Software architecture evaluation methods for performance, maintainability, testability, and portability. In *Second International Conference on the Quality of Software Architectures*.

Maurya, L. (2010). Comparison of software architecture evaluation methods for software quality attributes. *Journal of Global Research in Computer Science*, 1(4).

Ortega, M., Pérez, M., and Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. *Software Quality Control*, 11(3):219–242.

Ouhbi, S., Idri, A., Alemán, J. L. F., and Toval, A. (2014). Evaluating software product quality: A systematic mapping study. In *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA),*, pages 141–151. IEEE.

Ouhbi, S., Idri, A., Fernandez-Aleman, J. L., and Toval, A. (2013). Software quality requirements: A systematic mapping study. In *20th Asia-Pacific Software Engineering Conference*, APSEC, pages 231–238.

Ouhbi, S., Idri, A., Fernández-Alemán, J. L., and Toval, A. (2015). Predicting software product quality: a systematic mapping study. *Computación y Sistemas*, 19(3):547–562.

Patidar, A. and Suman, U. (2015). A survey on software architecture evaluation methods. In *2nd International Conference on Computing for Sustainable Global Development*, INDIACom, pages 967–972. IEEE.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, pages 71–80, Bari, Italy. Blekinge Institute of Technology.

Reijonen, V., Koskinen, J., and Haikala, I. (2010). Experiences from scenario-based architecture evaluations with atam. In *European Conference on Software Architecture*, pages 214–229. Springer.

Roy, B. and Graham, T. N. (2008). Methods for evaluating software architecture: A survey. Technical report, School of Computing. Queen's University at Kingston.

Salger, F. (2009). Software architecture evaluation in global software development projects. In *OTM '09 Proceedings of the Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems*, pages 391–400. Springer.

Salger, F., Bennicke, M., Engels, G., and Lewerentz, C. (2008). Comprehensive architecture evaluation and management in large software-systems. In *QoSA*, pages 205–219. Springer.

Santos, D. S., Oliveira, B., Guessi, M., Oquendo, F., Delamaro, M., and Nakagawa, E. Y. (2014). Towards the evaluation of system-of-systems software architectures. In *6th Workshop on Distributed Development of Software, Ecosystems and Systems-of-Systems (WDES co-located with CBSoft)*, pages 53–57.

Shanmugapriya, P. and Suresh, R. (2012). Software architecture evaluation methods-a survey. *International Journal of Computer Applications*, 49(16).

Sharafi, S. M. (2012). SHADD: A scenario-based approach to software architectural defects detection. *Advances in Engineering Software*, 45(1):341–348.

Subramanian, N. and Chung, L. (2004). Process-oriented metrics for software architecture changeability. In *Software Engineering Research and Practice*, pages 83–89.

Svahnberg, M. and Mårtensson, F. (2007). Six years of evaluating software architectures in student projects. *Journal of Systems and Software*, 80(11):1893–1901.

Tvedt, R. T., Costa, P., and Lindvall, M. (2002a). Does the code match the design? a process for architecture evaluation. In *International Conference on Software Maintenance*, pages 393–401. IEEE.

Tvedt, R. T., Lindvall, M., and Costa, P. (2002b). A process for software architecture evaluation using metrics. In *27th Annual NASA Goddard/IEEE Software Engineering Workshop*, pages 191–196. IEEE.

Upadhyay, N. (2016). SDMF: Systematic decision-making framework for evaluation of software architecture. *Procedia Computer Science*, 91:599–608.

Van Gurp, J. and Bosch, J. (2000). SAABNet: Managing qualitative knowledge in software architecture assessment. In *Seventh IEEE International Conference and Workshopon the Engineering of Computer Based Systems*, ECBS, pages 45–53. IEEE.

Vogel, O., Arnold, I., Chughtai, A., and Kehrer, T. (2011). *Software architecture: a comprehensive framework and guide for practitioners*. Springer Science & Business Media.

Witt, B. I., Baker, F. T., and Merritt, E. W. (1993). *Software architecture and design: principles, models, and methods*. John Wiley & Sons, Inc.

Zalewski, A. and Kijas, S. (2013). Beyond atam: Early architecture evaluation method for large-scale distributed systems. *Journal of Systems and Software*, 86(3):683–697.

Zhu, L., Babar, M. A., and Jeffery, R. (2004). Mining patterns to support software architecture evaluation. In *Fourth Working IEEE/IFIP Conference on Software Architecture*, WICSA, pages 25–34. IEEE.