Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 91 (2016) 599 – 608

Procedia Computer Science

Information Technology and Quantitative Management (ITQM 2016)

SDMF: Systematic Decision-making Framework for Evaluation of Software Architecture

Nitin Upadhyay[a,]*

[a]Information Technology and Operations, Goa Institute of Management, Goa India

Abstract

The software architectural decisions are crucial and critical to the success of a software project life cycle. The set of relevant design decisions affects the quality of the software architecture. In this paper, a systematic decision-making framework is proposed by considering management and organizational factors and design goals/parameters that affect software architecture (SA) and integrating it with the technique for order preference by similarity to ideal solution (TOPSIS) to evaluate and select the quality software architecture. An illustrative case study is also mentioned to show the applicability of the proposed framework. The framework suggested in the paper should enable an architect and other key stakeholders of the software architecture to efficiently identify, evaluate and select the software architecture.

Peer-review under responsibility of the Organizing Committee of ITQM 2016

Keywords: Decision-making; software architecture; design decisions; pattern; software quality; TOPSIS

1. Introduction

Software Architecture (SA) plays a critical role in realizing the quality of the software product. The software product stakeholders are increasingly more concerned about the quality of the software product to satisfy various functional and non-functional requirements. The researchers, decision makers, managers, practitioners and product owners have identified that SA of the software system help in understanding and managing large and complex software systems [1-2] and in constraining the quality attributes [3]. A quality SA is important to achieve a high-quality software system, both regarding development and long-term maintainability. Since SA plays a critical role in realizing software quality attributes, it has become a paramount task to evaluate SA about desired quality requirements as early as possible in the software development life cycle. The SA evaluation deals with the problem of assessing and selecting the potential SA, from the pool of alternatives SA candidates, that is capable of realizing required quality requirements [4]. The detection and fixing of possible errors and faults later in the software development life cycle contribute to enormous risks and costs. Thus, an early evaluation of SA plays a significant role in understanding software quality and associated potential risks [5]. Researchers have developed many methods to evaluate quality related issues at the

* Corresponding author. Tel.: +91-083225366751; fax: +91-0832-2366710.
E-mail address: upadhyay.nitin@gmail.com

Peer-review under responsibility of the Organizing Committee of ITQM 2016
doi:10.1016/j.procs.2016.07.151

SA level. The conventional qualitative and quantitative SA evaluation techniques [1-2], [4] majorly focus on the limitations of the quality in SA. Most of the available methods focus mainly on assessment of single quality attribute [6]. For example, a modifiability analysis of an SA is achieved through Architecture-Level Modifiability Analysis (ALMA) [7]; Scenario based Architecture Analysis Method (SAAM) [8] is used to analyze the modifiability attribute of the SA quality.

To achieve business value and stakeholders' requirements satisfaction, it is of utmost important to evaluate SA considering multi-attribute quality analysis. Thus, there is dire need of a systematic SA evaluation framework that considers management and organizational factors and designs goals/parameters that drive the quality of the SA. Moreover, provide results quantitatively for the purpose of benchmarking and ranking of the software architectures.

The remainder of this paper is as follows: Section 2 provides the literature review of the related work. Section 3 describes the systematic decision-making framework for SA evaluation. Section 4 presents an illustrative case study that shows the utility of the proposed framework. Finally, Section 5 concludes the paper.

2. Literature Review

One of the most vital features of SA evaluation method is the number of quality attributes that a process can handle for the evaluation. Most of the methods available in the literature focus mainly on a single quality attribute analysis for SA assessment. For example, a modifiability analysis of an SA is achieved through Architecture-Level Modifiability Analysis (ALMA) [7]; modifiability quality attribute has also been analyzed through Scenario based Architecture Analysis Method (SAAM) [9]; Active Reviews for Intermediate Design (ARID) [10]; Cost-Benefit Analysis Method (CBAM) [11], [6] focuses on Costs, Benefits, and Schedule Implications; Scenario-based Architecture Level UsabiliTy Analysis (SALUTA) [12] focuses on Usability of the SA; SAAM for Complex Scenarios (SAAMCS) [13] addresses the Flexibility attribute; Extending SAAM by Integration in the Domain (ESAAMI) [14] target to analyze modifiability attribute; Aspectual Software Architecture Analysis Method (ASAAM) [15] focuses on modifiability analysis; maintainability analysis is achieved through Architecture-Level Prediction of Software Maintenance (ALPSM) [16]; performance evaluation of enterprise architecture [17]; efficiency evaluation by developing executable model [18] and serviceability analysis of service-oriented architecture [19].

There is a lack of work on the systematic evaluation of SA considering multiple quality attributes quantitatively. Currently, only two techniques Scenario-Based Architecture Reengineering (SBAR) [20] and Architecture Tradeoff Analysis Method (ATAM) [8] focus on multiple quality attributes analysis. However, SBAR focus on only development and operational related quality attributes analysis [20] and thus limit the overall assessment of the SA towards achieving business value. ATAM specifically focuses on utility tree and scenarios to make the quality analysis. In case the scenarios are not mapped precisely and consistently then the overall assessment becomes the far-reaching goal. None of the methods mentioned in the available literature consider management and organizational factors and design goals/parameters that drive the quality of the SA for the evaluation purpose.

One of the widely used decision methods is Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) which is the focus point of usage in varied disciplines [21-25]. This technique functions on the concept that the chosen alternative should have the shortest Euclidean distance from the positive ideal solution and farthest from the negative ideal solution. Generally, TOPSIS is a technique where decision maker has to perform the evaluation of the finite number of decision alternatives under the finite number of criteria. The underlying philosophy of TOPSIS is that the selected option remains at the shortest distance, in a geometrical sense, w.r.t the ideal solution and longest distance from the worst solution. The purpose of the analysis is to rank the alternatives in an order of preference.

3. Systematic Decision-making framework (SDMF)

The process of SA evaluation comes under multi-criteria decision-making problem and to attain such a process of SA evaluation a systematic decision-making framework is proposed. The framework considers management and organizational factors and design goals/parameters for software architecture evaluation and also utilizes TOPSIS technique to restructure the complex domains composed of diverse internal and external factors in the SA evaluation process. Figure 1 presents the framework.