A Lightweight Value-based Software Architecture Evaluation

Chang-Ki Kim*, Dan-Hyung Lee**, In-Young Ko**, Jongmoon Baik**
*Electronics Telecommunications Research Institute(ETRI),
Gajong-dong,Daejeon, Korea
ckkim1@etri.re.kr
** Information Communications University(ICU),
Moonji-dong,Daejeon, Korea
danlee@icu.ac.kr, iko@icu.ac.kr, jbaik@icu.ac.kr

Abstract

Current software engineering practice is focused on value-neutral processes. Value-based architecting, one of value-based software engineering agendas, involves the further consideration of the system objectives associated with different stakeholder values in selecting an optimal architectural alternative. There are several value-based architectural evaluation techniques and Cost Benefit Analysis Method (CBAM) is a widely used, established technique based on Return on Investment (ROI). The weaknesses of the existing techniques are uncertainties from several subjective errors and the heavyweight process, which requires many steps and participation of stakeholders. This paper proposes a lightweight value-based architecture evaluation technique, called LiVASAE, using Analytic Hierarchy Process (AHP), which can support a multi-criteria decision-making process. The proposed technique can help overcome the major weakness of the existing techniques such as the uncertainties caused by subjective decision making and heavy-weight process for architecture evaluations. LiVASAE provides a way to measure the uncertainty level using AHP's Consistency Rate (CR) and It also provides three simplified evaluation steps. In addition, the LiVASAE presents a framework for decision makers to make technical decisions associated with business goals (or values) such as cost, time-to-market, and integration with legacy system.

1. Introduction

It is one of the most important issues in value-based software development to evaluate software architecture alternatives with respect to their corresponding QA(Quality Attribute)s. However, there is a tendency that people make a decision intuitively on software architecture selection with regard to a range of QAs, relying heavily on their previous experience. In order to avoid the intuitive decision making, Software Engineering Institute (SEI) at Carnegie Mellon University developed and refined value-based systematic architecture evaluation techniques such as Architecture Trade-off Analysis Method (ATAM) and Cost Benefit Analysis Method (CBAM).

ATAM helps to find optimal architectural decisions in a software project and associates them with technical QAs [5]. The CBAM is built on the ATAM to model the costs and the benefits of architectural design decisions and is a means of optimizing such decisions using a quantitative approach[4]. Because the CBAM is widely used, the LiVASAE adopt it by target reference for comparison. Although CBAM provides a good guideline to select optimal architecture based on economic values, there are some well-known weaknesses such as uncertainties from several subjective errors and the heavyweight process, which have requires many steps and participation of stakeholders. The architecture evaluation in the CBAM is basically based on stakeholders' consensus on a selection of optimal architecture alternatives or strategies after many steps. Therefore, the CBAM requires active discussion and clarification to make consensus amongst the stakeholders. Since the CBAM requires frequent and tedious participation of stakeholders, it is a regarded as a heavyweight process, which makes it difficult to be used in practice. To evaluate architecture alternatives effectively cost benefit analysis, the participation of stakeholders in the decision making process must be minimized. In addition, it is necessary to quantify the subjective data obtained from stakeholders in the voting process to reduce uncertainties.

In this paper, we propose a lightweight value-based software architecture evaluation (LiVASAE) using AHP (Analytic Hierarchy Process), one of Multi-Criteria Decision Analysis (MCDA) techniques, which is widely used to rank a finite set of alternatives in terms of a finite number of de-

0-7695-2909-7/07 $25.00 © 2007 IEEE
DOI 10.1109/SNPD.2007.507

cision criteria. It helps to avoid the impractical weaknesses of the CBAM and requires a much simpler steps with cost benefit analysis considering technical and business QAs simultaneously.

2. The LiVASAE Technique

There have been several researches and case studies that make use of AHP to evaluate software architectures. Mikael et. al. [3] and Tariq et. al. [6] proposed quantitative approaches to support the comparison of Architecture Strategies (ASs) using AHP. These methods provide structured ways of eliciting stakeholders' preferences for technical QAs. They help gain quantified understanding of the benefits and liabilities for different ASs. However they did not consider business QAs such as cost and benefit, time to market and integration with legacy system. The LiVASAE provides a systematic method to consider two QA classes concurrently for architecture evaluation.

To do this, the LiVASAE technique defines a modified decision matrix, which requires two elements, the relative importance of various QAs (the weight of QAs), and the relative supports of each AS on QAs (the performance value of QAs). Figure 1 depicts the conceptual process of computing these two elements. Both elements are calculated by a pair-wise comparison method in AHP. Reference [7] describes the AHP and pair-wise comparison in more detail.

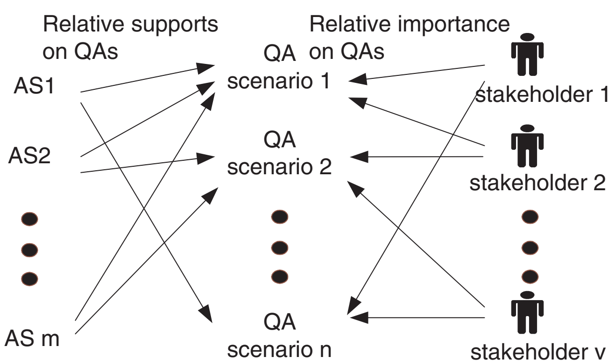

Figure 1. Conceptual process of computing elements value of decision matrix.

The new decision matrix is shown in Figure 2. It can be used for cost benefit analysis. Business QAs parts are extended to the existing decision matrix, which supports only technical QAs. QA_1,...,QA_{n1} represent technical QAs that are prioritized by ATAM's output, while QA_{n1+1},...,QA_n (i.e., the remaining ones) are the business QAs (where $1 < n_1 < n$). Alternative ASs, $AS_1 \sim AS_m$ are also obtained from ATAM's output.

In order to compute the relative importance and support of the decision matrix, the weight(w_j) of QAs and the performance value(a_{ij})of QAs, we use the pairwise comparison method because it includes much redundancy of comparison so that the result is less sensitive to judgmental errors that are commonly caused by the techniques using absolute assignments [2][7].

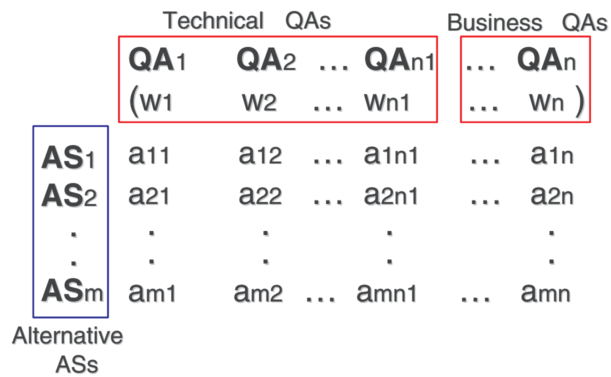

Figure 2. A modification of decision matrix.

We need to calculate the preference value of each AS based on return on investment (ROI). To get the preference value, we use the multiplicative AHP that is known for more consistent than other MCDA methods when benefit and cost criteria are compared together [1]. According to [1], the preference ROI value of each AS with the modified decision matrix can be calculated as follows.

$$P_{i,ROI} = \frac{\prod_{j=1}^{n_1}(a_{ij})^{w_j}}{\prod_{j=n1+1}^{n}(a_{ij})^{w_j}} \quad (1)$$

3. Evaluation and Analysis

In order to evaluate the effectiveness of using the LiVASAE technique described in this paper, we conducted an experiment in real practice. The experiment is done to give a more comprehensive presentation of how the technique can be effectively used based on the performance analysis on the LiVASAE and CBAM applied to the same project.

We applied both of the techniques to a one-year development project, a database management tool for a car navigation system. The major functional requirements of the target system include inspection, cleansing, and analysis of the point of interest (POI) data. On the other hand, both the LiVASAE and the CBAM must follow the ATAM. In other words, both techniques start with the output of the ATAM. Among the outputs of the ATAM, the ASs and the technical QAs are used as the inputs of the LiVASAE and the CBAM. The target system has four architectural strategies, AS1, AS2, AS3, and AS4. Figure 3 shows the major technical QAs obtained from the outputs of the ATAM. Each QA has a detailed scenario. For example, the scenario of the progress tracking (PT) is about "A user can recognize how much time left before the completion of a function ".

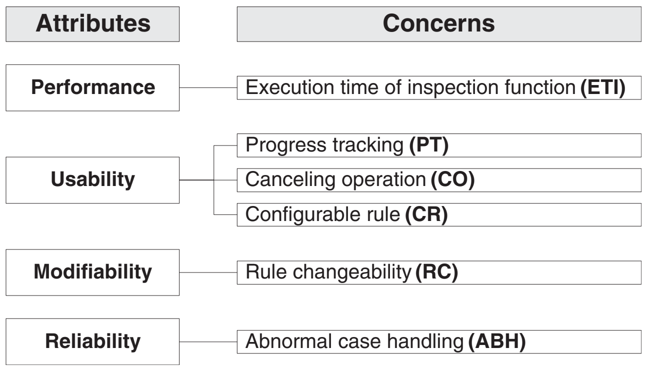

Figure 3. Collated quality attribute scenarios.

With these ASs and QAs, we had fourteen short interviews with the stakeholders to get the essential data for a new decision matrix. We conducted the experiment by using the LiVASAE technique as below For the comparison with the output of the CBAM we used only one business QA.

Step 1: Calculation of Weights(w_j) for Technical QAs.
To calculate the weight value of each QA, a data collection questionnaire was distributed to thirteen stakeholders from three different groups, customers(SH1), developers(SH2) and researchers(SH3). Then, they provided us with their relative preferences on QAs. Based on the collected data, the weight values were calculated by performing pairwise comparisons.

Each group's preference on each QA is the mean of individuals' relative preferences on the QA. The weight value for each QA is determined by taking the mean value of three group's preference on the QA as shown in the last column of Table 1. The process of calculating the weight values of business QAs is not explained here in this paper since the LiVASAE uses only one business QA, cost.

Step 2: Calculation of Performance Value(a_{ij}) for ASs.
In the second step, Each AS's relative support (Performance Value) for each of the technical QAs were collected from the group of developers. The relative support value for each QA is the mean value of individual developers' data of relative support that was already obtained from pair-wise comparisons. Table 2 shows the ASs' relative support values for technical QAs. For example, AS1 seems to be the best alternative that supports ETI. The performance value of business QA, cost, is the normalized value of the same implementation cost as used in the CBAM.

Table 1. Technical QAs' relative importance weights

Technical QAs	Stakeholders			weights of QAs
	SH1	SH2	SH3	
ETI	0.32	0.49	0.50	0.44
PT	0.11	0.18	0.16	0.15
CO	0.06	0.12	0.13	0.10
CR	0.13	0.06	0.11	0.10
RC	0.36	0.10	0.06	0.18
ABH	0.03	0.04	0.04	0.04

Table 2. ASs' relative support value with respect to technical QA

ASs	Technical QAs					
	ETI	PT	CO	CR	RC	ABH
AS1	0.38	0.10	0.16	0.25	0.25	0.10
AS2	0.14	0.25	0.26	0.25	0.25	0.36
AS3	0.22	0.26	0.19	0.25	0.25	0.11
AS4	0.26	0.39	0.38	0.25	0.25	0.43

Step 3: Calculation of ROI from Decision Matrix.
The final step of the LiVASAE technique is to calculate the ROIs of ASs using the multiplicative AHP method with a modified decision matrix as shown in Table 3. The decision matrix includes importance weights of QAs and relative support values calculated in the preceding steps. The ROI preference value P_i can be calculated by the formula (1) of the previous chapter. The calculated ROIs are ranked as shown in the Table 4.

As we can see, the analysis result of LiVASAE is consistent with CBAM's result even if LiVASAE uses a different way to calculate ROI values. The Consistency Ratio (CR) of QA weights defines the accuracy of the pair-wise comparisons. The LiVASAE's CR is 0.077 (< 0.1) which is accpetable.

For the performance analysis between LiVASAE and CBAM, cost-value approach was used. It prioritizes ASs according to their relative benefit value and cost. Figure 4 shows the cost-value diagram of two technique after unit

Table 3. Decision matrix for ROI calculation

	ETI	PT	CO	CR	RC	ABH	COST
	w_1 0.44	w_2 0.15	w_3 0.10	w_4 0.10	w_5 0.18	w_6 0.04	w_7 1.00
AS1	0.38	0.10	0.16	0.25	0.25	0.10	0.31
AS2	0.14	0.25	0.26	0.25	0.25	0.36	0.22
AS3	0.22	0.26	0.19	0.25	0.25	0.11	0.28
AS4	0.26	0.39	0.38	0.25	0.25	0.43	0.19

Table 4. ROI values and ranking order

	ROI value(P_i)	Strategy rank
AS1	0.78	4
AS2	0.90	2
AS3	0.80	3
AS4	1.54	1

adjustment for a practical comparison.

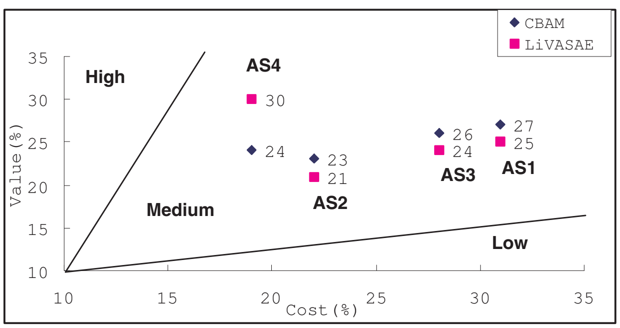

Figure 4. Cost-value diagram

In Figure 4, x-axis is the normalized cost of the each AS and y-axis is the normalized benefit of the each AS. There are three distinct areas in the cost-value diagrams : high ratio(the ratio exceeding 2), medium ratio (the ratio between 0.5 and 2), and low ratio(ratio lower than .5).

As shown in Figure 4, all the ASs fall into medium ratio category. The overall performance between LiVASAE and CBAM is similar. With the same cost, there are a few additional benefits in the CBAM. Overall order of ASs in terms of cost benefit analysis is the same in CBAM (AS4, AS2, AS3 and AS1 order). Even if LiVASAE technique has the same performance it could provide a simplied steps instead of many steps requiring participation of stakeholders.

4. Conclusion

We have introduced a lightweight value-based architecture evaluation (LiVASAE) technique to overcome the weakness of the existing software architecture evaluation techniques, heavyweight process steps and uncertainties. It performs systematic cost benefit analysis using AHP. It also adopts a pair-wise comparison from AHP to reduce uncertainties that affect decision making on software architecture alternatives. Technical QAs and business QAs are separately considered and multiplicative AHP is used to calculate final ROIs.

We performed an experiment in real practice to verify the effectiveness of the LiVASAE technique via cost-value analysis over the CBAM. Even though both techniques show similar results, the LiVASAE provides a way to measure the uncertainty level using AHP's CR and it also provides three simplied evaluation steps. The LiVASAE also provides a framework for decision makers to make architecture evaluation associated with the business QA values as well as technical QA values. More research that supports more than two business QAs is required.

References

[1] K. E.Triantaphyllou. The impact of aggregating benefit and cost criteria in four mcda methods. *IEEE Transactions on Engineering Management*.

[2] K. J.Karlsson. A cost-value approach for prioritizing requirements. *IEEE Software Magazine*, October.

[3] C. M.svhnberg and L.Lungberg. A quality-driven decision-support method for identifying software architecture candidates. *International Journal of Software Engineering*.

[4] J. R. Kazman and M. Klein. Making architecture design decisions :an economic approach. *CMU/SEI-2002-TR-035, Software Engineering Institute, Carnegie Mellon University*.

[5] M. K. R. Kazman and P. Clements. Atam:a method for architecture evaluation. *CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University*.

[6] F. T.Al-Naeem, M.Ali Babar and B. Benatallah. A quality-driven systmatic approach for architecting distributed software applications. *International Conference of Software Engineering*, May.

[7] T.L.Satty. *The Analytic Hierarchy Process*. McGraw-Hill.