Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

Enhancing Software Architecture Review Process via Knowledge Management

Sofia Sherman
Software Architecture Lab, Department of Management Information Systems, University of Haifa, shermans@mis.haifa.ac.il

Irit Hadar
Software Architecture Lab, Department of Management Information Systems, University of Haifa, hadari@mis.haifa.ac.il

Meira Levy
Software Architecture Lab, Department of Management Information Systems, University of Haifa, lmeira@mis.haifa.ac.il

Enhancing Software Architecture Review Process via Knowledge Management

Sofia Sherman
Software Architecture Lab.
Department of Management Information Systems
University of Haifa
shermans@mis.haifa.ac.il

Irit Hadar
Software Architecture Lab.
Department of Management Information Systems
University of Haifa
hadari@mis.haifa.ac.il

Meira Levy
Software Architecture Lab.
Department of Management Information Systems
University of Haifa
lmeira@mis.haifa.ac.il

ABSTRACT

Software architecture is considered to have a significant influence on the final software product's quality. A critical phase in ensuring and validating the quality of a suggested architecture is architecture review, conducted by experienced architects. While many evaluation methods have been researched thus far in the context of architecture review, little attention has been given to the review process and to the knowledge-related aspects embedded within it. In this paper we explore and analyze the architecture review process based on literature as well as empirical evidence obtained in a case study conducted in a large software development firm. For the aim of enhancing the review process to a systematic and scalable process, as well as expending its value to future architecture in addition to the reviewed artifacts, this research investigates the knowledge aspects of the review process and suggests a conceptual solution for enhancing the review process and embedding knowledge management within it.

Keywords Knowledge management, software architecture, architecture review, process

INTRODUCTION

There is a growing awareness nowadays of software architecture (SA) review as a critical phase in the software architecture process. Cook (2007), Babar, et.al (2004) and others claim that architecture review significantly improves architecture, leading to delivering higher quality systems to customers in lower costs, thus contributing to customer and product. "*The objective* [of software architecture review] *is to identify potential issues with a proposed architecture, prior to the construction phase, to determine its architectural feasibility and to evaluate its ability to meet its quality requirements.*" (Cook, 2007, p. 1). In addition, the architects gain knowledge and experience based on the feedback they receive on their work. Cook (2007) suggests that architects should treat their architecture review as a learning opportunity which, along with its contribution to the project, fosters the organizational architects' professional development. Another major type of stakeholders involved in the SA review process, are of course the architecture reviewers, who operate in different levels of architecture process, for example: pair review, manager review or board review. Here we address the architecture review board (ARB) following the definition by Maranzano, et al. (2005): *"The architecture review board is a standing board that oversees the review process and its effect on the organization"* (ibid, p.36).

SA review process is a very complicated and demanding process for both parties involved: (1) the architects, who deliver an architecture document, usually built according to templates, rules and other requirements dictated by the organization, and (2) reviewers, who review the document and the architecture it represents aiming to ensure it is of high quality and consistent with the objectives presented above. However, evidence from the field has shown that architecture documents to be reviewed

often include hundreds of pages that are difficult to read and understand in a reasonable period of time. Moreover, frequently, these documents include topics that are not always aligned with the reviewers' expertise, thus demanding even more time and effort as well as additional information and knowledge resources that are not always available.

The research presented in this paper is based on a case study that took place in a global and large software organization. The organization was interested in scaling up the review process beyond its current operation. The objective of this research is to identify difficulties in the organizational SA review process, from both architects and reviewers points of view, that may impede up scaling this process, and suggest recommendations that will help overcoming these difficulties.

The paper is organized as follows: the next section presents relevant existing literature; next we describe the empirical study we conducted, its setting and findings; we present a conceptual solution and discuss its alignment with the empirical findings, and finally we conclude.

THEORETICAL BACKGROUND

Software architecture establishes major design decisions which determine the system's development, deployment and evolution (Babar, et.al, 2004). SA review is a critical phase in software architecture, aiming at ensuring and validating the quality of a suggested architecture. Literature review has revealed a wide range of evaluation methods and their benefits as well as shortcomings with regard to SA review. However, as it turns out little attention has been given to the review process in general, as well as to the knowledge-related aspects embedded within it. In what follows we briefly review architecture evaluation methods and their analyses as appears in literature on the one hand and knowledge management related to SA on the other hand. These will provide the theoretical background for our proposed solution to close the gap between architecture review and knowledge management while defining an SA review process taking into account the different required aspects.

Software Architecture Evaluation Methods

SA evaluation is a relatively new area of research. Although various methods for SA evaluation have been developed, they are still being validated and are considered complementary techniques to scenario–based methods, which is one of the most mature software architecture evaluation methods' type (Babar et al., 2004). These methods include, for example: Software Architecture Analysis Method (Kazman, et al., 1994), Architecture Tradeoff Analysis Method (Kazman, et. al., 1998), and Architecture-Level Maintainability Analysis (Lassing et al., 2002). Scenario-based methods are executed as a collaborative exercise that involves a number of stakeholders collocating in face-to-face meeting while performing various activities, such as defining and refining business drivers, generating quality sensitive scenarios and mapping them on proposed architecture (Babar et al., 2005). Rosso (2006) describes lessons learned from a case study of using scenario-based assessment techniques that include: SA documentation improvement, enhanced communication, scenarios collection and assessment focus, problem discovering during the whole software lifecycle and improvements for the software product family. Rosso (2006), Babar et al. (2004), and Inoita, et al. (2002) present comparison between different SA evaluation methods, when applying to the quality attributes they assess, architecture documentation they include, approaches they use, stage in the development lifecycle they appear in, and more.

Babar and Gorton (2004), based on the main SA evaluation approaches, identified five common activities which can make up a generic scenario-based SA evaluation process: evaluation planning and preparation for setting evaluation goals, selecting stakeholders and deciding on an evaluation team; explaining architectural approaches by architects' presentation of the architecture being reviewed; eliciting and developing quality attributes sensitive scenarios for examining the developed system; analyzing architectural approaches with respect to scenarios developed earlier; and interpreting and presenting results as a summary of the review process.

Babar et al. (2004) claim that there is no single method that can be considered as equally good for all types of assessment objectives, as different methods are optimized to achieve different evaluation goals. Some of the methods provide suitable guidance on required architectural description and views while most of them provide techniques for quality attribute characterization and scenario generation and evaluation They define the mutual goal of most evaluation methods as "*evaluation of the potential of the designed architecture to facilitate or inhibit the achievement of the required quality attributes"* (Babar et al., 2004, p. 312).

Software Architecture Evaluation Process

As described above, the literature does not provide a single evaluation method appropriate to all types of review objectives. Moreover, Babar et al. (2004) note that human aspects of the SA review process have been given little attention and that no tool exists for supporting the evaluation process. Babar et al. (2005) show that scenario based evaluation is an expensive and

time consuming process. Additionally, the geographically distributed stakeholders may encounter problems in organizing and participating in face-to-face meetings. In an attempt to overcome these difficulties, Babar et al. (2005) propose the concept of distributed architectural evaluation using internet-based collaborative technologies. They describe a pilot study which shows that the majority of reviewers who participated in the research preferred face-to-face meetings for the reasons such as: use of body language, natural and conventional meetings, and efficient collaboration due to not having typing problems and time lag.

Maranzano et al. (2005) propose five basic principles of architecture review: (1) A clearly defined problem statement drives the system architecture. (2) The architect must explain the reasons for all decisions and document choices (3) Independent experts conduct reviews. (4) The review team conducts the review openly and invites project members to attend. This approach develops the trust between reviewers and project members that is necessary for successful review. (5) Companies conduct reviews for the project's benefit. Additionally, Maranzano et al. propose several basic principles for building review process, such as:

(1) The architecture review process needs to gain acceptance and be viewed as a helpful mechanism rather than a management "stick". It is vital that the reviewers provide objective findings.
(2) The board should monitor trends that appear across reviews and fix systematic issues (establishing a training program or investing in a commercially available training program
(3) Commit to keeping the review results confidential to the project, its management, and the board unless stakeholders are willing to distribute them more widely.
(4) Establish a reviewer database that contains reviewers' names, organization, and technical expertise.

Former research acknowledged the important value of reviewers' input for improving SA and producing quality products (Babar, et.al, 2004; Cook, 2007). However, they do not address the value in leveraging the knowledge constructed and captured during the SA review process, for improving future systems' architecture development. While software architecture is considered as a major component for achieving software product quality, we still lack techniques for capturing, representing and maintaining knowledge about software architectures (de Boer et al., 2007). In particular, organizational knowledge gained in previous software architecture experience is a critical source within the architecture review process (Maranzano et al., 2005, Babar et al., 2009). In this regard, we introduce a review process that includes capturing and handling reviewers' comments as a knowledge resource for future software architecture processes within the firm.

Knowledge Management and Software Architecture

Knowledge management (KM) is becoming a key management factor for enhancing decision making processes, shortening time, saving rework, and improving quality of products. KM mainly targets knowledge intensive processes that are critical to the organizational performance, such as SA, which embody knowledge and serve as vehicles for communication among stakeholders. In particular, SA captures early design decisions and it is a transferable abstraction of the system which enables further reuse in other software systems (Babar et al., 2009). Thus, managing architecture knowledge is becoming an important and inherent part of the software development processes.

Several researches foster collaboration between software architects for sharing architecture knowledge (e.g., Farenhorst et al., 2007; Babar et al., 2009). However, we didn't find any research that focuses on the architecture review process. Former research indicated that architecture reviews are a critical part of the architecture process, aiming at identifying project problems before they become costly to fix, providing information to various stakeholders of the project for better decision making, and improving the product's quality (Maranzano et al., 2005). Their research also claimed that KM should be embodied in architecture review for capturing best practices and socialized such practices across the organization (Maranzano et al., 2005). While their research fosters cross-organizational learning and transfer of lessons learned and practices, as well as the need to identify and share corporate assets and reusable components, it does not provide a framework for embedding the KM practices within the review process itself for facilitating collaboration and engagement among the various review participants, such as the reviewers and the architects. The research presented in this paper aims to bridge this gap by providing an environment and practical guidelines for how to embed KM practices throughout the review process for the adoption of already existing architecture knowledge in new SA development and review.

THE EMPIRICAL STUDY

Methodology and Settings

This exploratory study was conducted in a large software firm. Due to confidentiality reasons, the name and other recognizable characteristics of this firm will not be mentioned here.

The main objective of this study was to identify difficulties in the existing SA review process**.** When aiming to learn a phenomenon and identify its characteristics, rather than corroborating predetermined hypothesis, it is appropriate to use qualitative research methods and tools (Bassey, 1999). In light of the research objective, the methodology used in this study is based on the qualitative grounded theory approach (Strauss & Corbin, 1994).

The research was conducted in three phases. The first phase included documents analysis and open discussions which were qualitatively analyzed, and the main categories regarding difficulties within the SA review process emerged. At the second phase, based on the findings of the first phase, semi-structured interviews for architects and reviewers were developed. The interview for the reviewers included 23 questions which focused on the review process and its strengths and weaknesses, while the interview for the architects included 20 questions, which focused on the preparation of the architecture document towards the review process as well as on the review process. The structured sections of the interviews were based on the emerging categories whereas the open section was designed to give the participants an opportunity to raise additional issues. All the questions within the interviews were open questions that allowed the interviewees to express their perceptions about the review process. The interviews were constructed by the first author and refined and validated first by the two co-authors and then by two experienced architects, who also serve as reviewers, from the studied firm. Then the interviews were conducted with 5 architects and 3 ARB members. At the third phase of this research, following the analyzed findings, a conceptual solution model was developed.

Data collected were transcribed and inductively analyzed. The text of the documents and transcribed discussions was divided to segments. For each segment we determined whether it is related, directly or indirectly, to difficulties within the SA review process. Only the related segments were included in the next analysis step, for identifying categories and classifying segments accordingly. The data collection and analysis was conducted iteratively, until category saturation was achieved. Finally, the main categories were determined, as presented in the tables below, and all segments were classified accordingly.

The final validation was conducted by a review meeting in which we presented the indentified difficulties and the recommended solution. In this meeting, in addition to the research team, 13 participants from the investigated firm were present, 11 of which were experienced stakeholders of the review process (architects and ARB members, most experienced in both roles) and 2 representatives from the education division of the firm. The participants agreed with the difficulties and enhancement requirements identified with regard to the review process and approved the conceptual solution proposed.

Findings

The outcomes of the data collection and analysis included the identification of the current SA review process and the difficulties and enhancement requirements in the context of this process as perceived by architects and by ARB members. The current SA review process that was identified is presented on Figure 1.

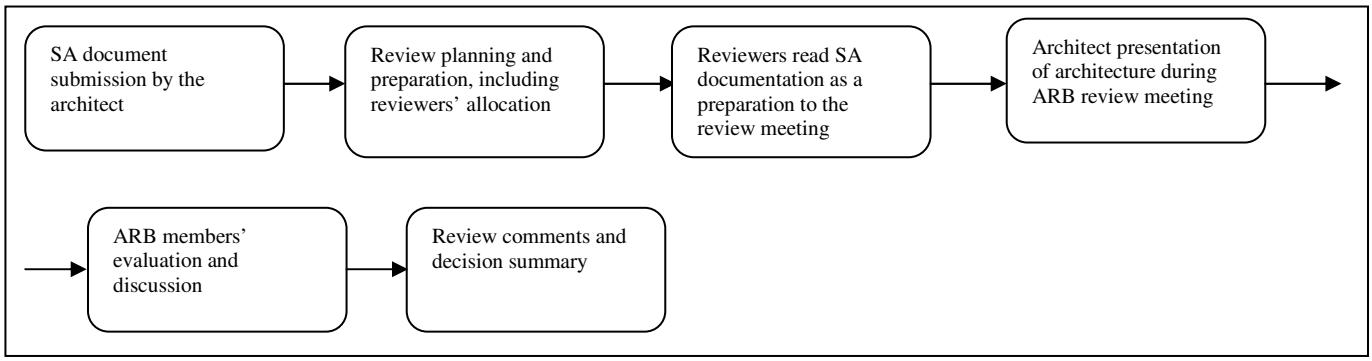

Figure 1: The current review process

The main weaknesses of the SA review process that were reported by the research participants were categorized and are presented in the two tables below. The main two categories that emerged from the data analysis and are common to both architects and reviewers relate to collaboration and resources. One of the main problems caused by insufficient collaboration between stakeholders is the dis-attachment between the review process and the architecture process as well as among

different review processes. Knowledge reuse, preservation and accessibility are vital for good, effective and efficient review process for making use of already constructed and proven knowledge and reducing time and other resources invested.

Table 1 summarize the categories that emerged from the ARB members' point of view and Table 2 presents the categorizes that emerged from the architects' point of view, and relate additionally to pre and post review stages, including SA development and document preparation before the review and a correction if necessary after the review process.

Category	**Problem Definition**	**Evidence from the field - examples**
Collaboration	There is no formally supported collaboration process and tools between the reviewers and architects.	*"There is no tool to support it"* *"I have no time for contacting the architects and discuss the architecture with them."*
Resources: Time	Lack of time to read and understand the entire SA document.	*"I don't know how to do meaningful review of >300 pages of architecture document"*
Resources: Material	When performing review, reviewers often need to access different relevant documents related to the same or other projects within the firm. Although this information usually exists, it is not easy and immediate to retrieve.	*"Various hardware/platform/technology details suggest looking at the requirements document. But I don't know where I can find ii. It would be very helpful to have a link to the relevant revision of such document. "*
Expertise	Reviewers tend to concentrate in their review on parameters relevant to their interests and expertise, and skip those that are not familiar to them.	*"In my opinion, there is a lot of information I have no way to comment on such as low-level engineering estimates."* *"… in some cases the document does not contain information potentially interesting to me."*

Table 1: ARB data analysis

Category	Problem Definition	Evidence from the field
Collaboration	Architects are often not completely familiar with reviewers' requirements, and do not know how to adjust the architecture document to the review.	*"Once I posed a question to one of the reviewers via email, but never got a response; he was probably too busy to answer."*
Resources: review outcomes	The reviewers' comments regarding the architecture documents do not contribute to the next review processes.	*"It would have helped significantly if I knew what the reviewers expect of me."* *"I would have spent much less time on the SA document preparation if I knew exactly what the review's objectives are"*
Education	There are no guidelines or a formal educational procedure for writing an architecture document. However there is a template architects must use when preparing the document. Thus there is self-education process during the architecture document writing; using the template leads the architects to rethink, recheck and improve the architecture.	*"It takes a lot of time to write an architecture document, especially when writing it for the firsts times"* *"I wish someone had taught me how to write the architecture document"* *"There are parts in the architecture document that I don't understand the*

		purpose of, so I ignore them" *"Using the template helps me to think about possibilities I was not aware off"*
Skills	Architects do not always have presentation skills required for the review meeting.	*"I don't know how to deliver high level message "* *"If there were guidelines for presenting the architecture, focusing the message etc, it would have been very helpful "*

Table 2: Architect data analysis

THE CONCEPTUAL SOLUTION

Based on the data analysis and categorization presented above, the conceptual solution model was developed and is presented in Figure 2 below followed by an explanation. The main benefits of the proposed solution are:

- Reviewers' profiling: in order to enhance the review process to be more efficient and focused, we propose to profile the reviewers and establish a 'reviewer profile' for each reviewer, according to their expertise. The 'reviewer profile' will guide the allocation of architecture to be reviewed to the appropriate reviewers, so that each of the reviewers involved in the process will focus and take responsibility for the parts of the document in the area of their expertise.
- Knowledge reuse: every outcome from the review process will be stored within the 'recommendation database'. Outcomes will include discussions between reviewers and architects, and among reviewers, during the process of the SA development, review preparation, discussion and decisions, and so on. All that information will be accessible for the architects and the reviewers. The architects will be able access this database in order to reuse knowledge for architecture development and learn about reviewers' requirements and comments regarding the SA document. The reviewers will be able to access previous reviews' outcomes in order to see issues that were previously raised regarding the same product/customer/architect and so on.

An additional issue this solution covers is online collaboration. A study regarding distributed architectural internet-based evaluation (Babar et al., 2005) showed reviewers' lack of satisfaction regarding online decision making processes. However, our solution does not offer to manage an online review but rather to use a collaborative platform in order to interact merely as a preparation towards the review meeting.

Figure 2 describes the SA review process and its surrounding environment. Note, we suggest including the architecture development activity as related to, although of course not included in, the review process.

In the context of the conceptual solution presented above, these are the main usage scenarios:

1. During SA development and document preparation, architects will be able to use the collaboration system in order to communicate with each other, retrieve knowledge for reuse during their current architecture document preparation including developments history with regard to previous architecture documents they, as well as other architects, submitted. They will also be able to access previous review comments, discussions and decisions.
2. After the SA document is ready to be submitted for review, the collaboration system will allocate the review responsibilities to the different ARB members according to their profiles. The profiles may be changed individually by each reviewer.
3. During the review period, reviewers may communicate with each other using the collaboration system. They will be able to look for history of previous reviews for the same or other products, and read comments and discussions among reviewers regarding other relevant SA reviews. Using the system for discussion between reviewers regarding the reviewed document will contribute to handling knowledge history by documenting and storing it in the recommendation database.

4. Eventually, the data regarding each review, including the SA document, comments, discussions and decisions will be stored in the recommendation database and for future use by both architects and reviewers.

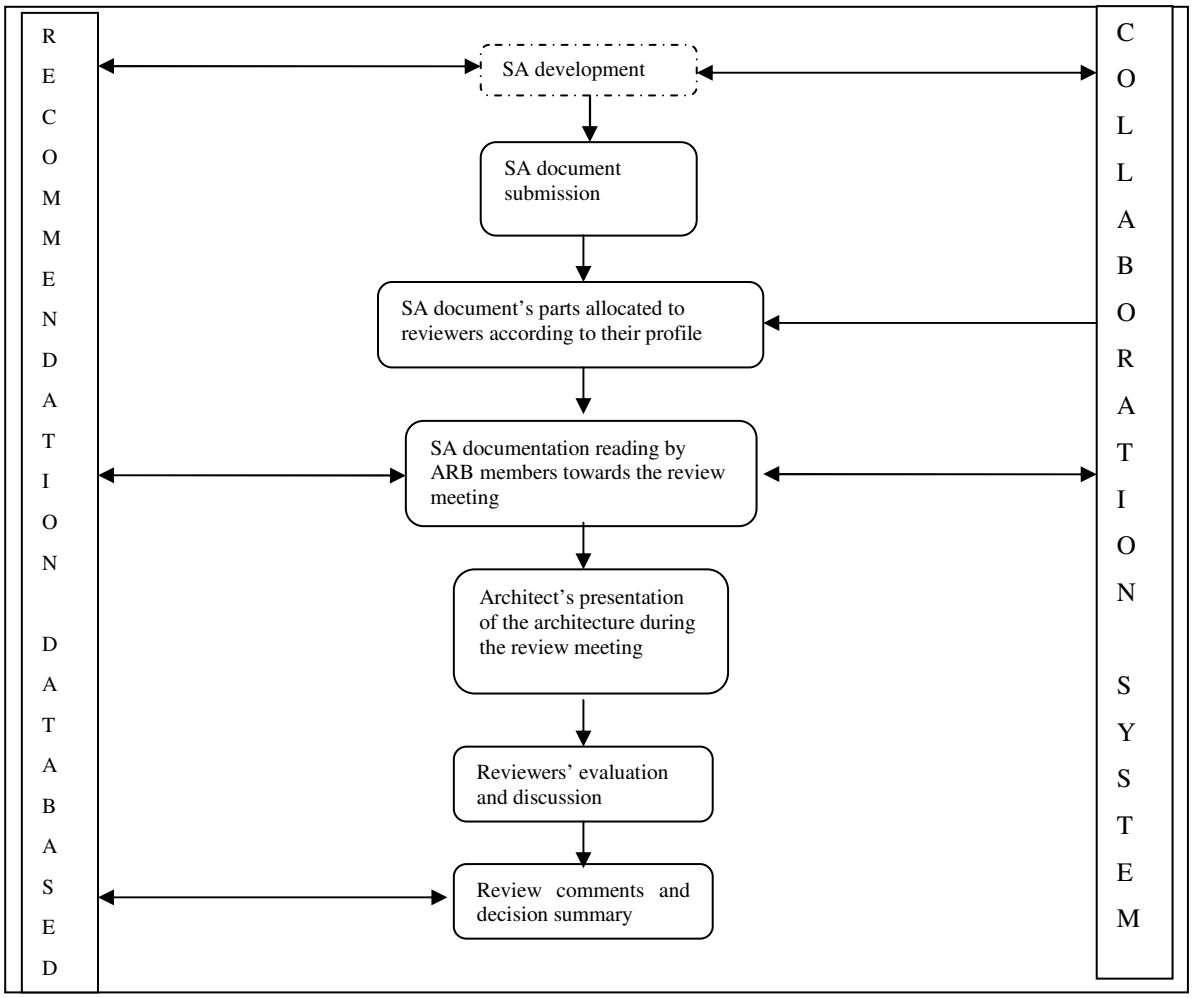

Figure 2: Suggested review process and environment

An additional, future solution we propose for both architects and reviewers will be developing a bidirectional checklist. The checklist will be developed according to the organizational and reviewers' requirements regarding the context of the SA document. Using the bidirectional checklist will strengthen the collaboration between architects and ARB members: it will support the architects during the preparation towards the review and eventually this checklist will be submitted with the SA document for review; and will include a short reference regarding all topics written by the architects in order to be used as a roadmap for reviewing the SA document by the ARB members.

DISCUSSION

The case study presented in this paper describes the difficulties and weaknesses of the existing review process, in particularly in light of the firm's aspiration to upscale it, and offers a conceptual solution for using a collaborative environment and knowledge management mechanisms. While the expected benefits were discussed above, some shortcomings and limitations need to be understood and carefully handled before implementing a solution based the principles suggested in this research.

The use of collaborative tools and knowledge sharing mechanisms depend on the willingness of the stakeholders involved to collaborate with each other. The main principle of the proposed solution is full collaboration on all levels and stages of the

SA development (architects, project management, ARB members). In order to promote such collaboration it should be defined as one of the basic principles for SA review within the firm. Maranzano et al. (2005), describing their five principles of the review process (see section 2), relate to the review as an open process. We believe that using a collaboration system during the review process will contribute to an open approach and will contribute to knowledge reuse due to transparency and collaboration between architects from different projects and products. Nevertheless, collaboration is not always achieved even in cases when its benefits seem to be clear. When implementing such a solution, careful investigation of the stakeholders' willingness to collaborate must be performed and, when needed, incentives must be provided in order to ensure the feasibility of the solution.

The proposed solution implicates that all reviews' data (results, conversations, decisions) will be open to use not only for ARB members involved in the review and project management but also for other project stakeholders including the architect reviewed as well as his fellow architects in the firm, that may learn from the review results as well. This approach contradicts the approach suggested by Maranzano et al. (2005) who claim that the review results should be confidential to the project, its management, and the board unless stakeholders are willing to distribute them more widely. While we understand the considerations supporting confidentiality, the knowledge that may be collected during the whole architecture process generally and on review process particularly is vital to the next SA developments and reviews and we believe would be very beneficial to all stakeholders in the long term as well as to the organization in general. As a possible solution to the problem described, we suggest using the review coordinator that additionally to the summary of the reviews results, will summarize all outcomes in a way that minimizes the identification and maximize knowledge reuse. The recommendation database also requires coordinator to maintain and add the data to the database. Today, at the firm we studied, there already is an allocated coordinator who summarizes reviews outcomes. In the proposed solution his responsibilities would include some additional KM activities derived from this solution.

The conceptual solution suggested in this paper is based on a case study performed in a large, global, distributed, however single, firm. Further validation and generalization of this solution will be enabled by extending the research to additional software firms. As well, a practical implementation of this solution is required, and planned to be conducted next, in order to prove its feasibility and accurately measure its benefit. As this research, and the solution it proposes, is about building a knowledge base for reuse and organizational learning, it would take a long-term study to fully capture its value.

CONCLUSION

This paper explored the architecture review process based on literature and a case study conducted in a large software development firm. Its main objective was enhancing the review process to a systematic and scalable process, as well as expending its value to future architecture development and review in addition to the reviewed artifacts. To this end, this research investigated the difficulties and enhancement opportunities of the architecture review process, in particular in the context of the knowledge it produces and requires, and proposed a conceptual solution for enhancing the review process and embedding KM within it. Further research is required for expending the generalizability of the findings of this research and the proposed solution beyond the case study conducted, as well as for a practical validation of this conceptual solution.

REFERENCES

1. Babar, M. A., DingsØyr, T., Lago, P., and van Viliet, H. (2009). Software Architecture Knowledge Management - Theory and Practice. Springer.

2. Babar, M., Kitchenham, B., Zhu, L., Gorton, I., and Jeffery, R. (2005). An empirical study of groupware support for distributed software architecture evaluation process. *Journal of Systems and Software*, 79(7):912-925.

3. Babar, M. A., Zhu, L., and Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods. *Australian Software Engineering Conference*, 0:309-318.

4. Babar, M. A. and Gorton, I. (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. APSEC'04, 600-607.

5. Bassey, M. (1999). Case Study Research in Educational Settings, Chapter 7: Methods of Enquiry and the Conduct of Case Study Research. *UK: Open University Press*, 65-91.

6. Cook, D. (2007). Architecture evaluation and review practices. http://msdn.microsoft.com/en-us/library/bb896741.aspx Accessed, February 2010

7. de Boer, R. C. , Farenhorst, R., Lago, P., van Vliet, V., Clerc, V., and Jansen, A. (2007). Architectural Knowledge: Getting to the Core, in Overhage, S. et al. (Eds.): QoSA 2007, LNCS 4880, pp. 197–214, Springer-Verlag Berlin Heidelberg.

8. Farenhorst, R., Lago, P., van Vliet, H. (2007). EAGLE: Effective tool support for sharing architectural knowledge. Int. J. Cooper. Inform. Syst. 16(3/4), pp. 413–437.

9. Ionita, M., Hammer, D., 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*.

10. Kazman, R., Bass, L., Webb, M., and Abowd, G. (1994). Saam: a method for analyzing the properties of software architectures. In *ICSE '94: Proceedings of the 16th international conference on Software engineering*, pages 81-90, Los Alamitos, CA, USA. IEEE Computer Society Press.

11. Kazman, R., Klein, M., Brbacci, M., Longstaff, T., Lipson, H., and Carriere, J. (1998). The Architecture Tradeoff Analysis Method, *Proceedings of IEEE, ICECCS.*

12. Lassing, N., Bengtsson, P., Bosch, J. and Vliet, H.V. (2002). Experiences with ALMA: Architecture-Level Modifiability Analysis. *Journal of Systems and Software 61(1).* 47 – 57.

13. Maranzano, J. F., Sandra A., Rozsypal, S. A., Zimmerman, G. H., Warnken, G. W., Wirth, P. E. and Weiss, D. M. (2005). Architecture Reviews: Practice and Experience. IEEE SOFTWARE, Mach-April.

14. Rosso C.D. (2006) Continuous evaluation through software architecture evaluation: a case study. *Journal of Software Maintenance and Evolution: Research and Practice.* 18, 351-383.

15. Strauss, A. & Corbin, J. (1994). Grounded Theory Methodology: An Overview. In Denzin, N.K. & Lincoln, Y.S. (Eds.), Handbook of qualitative research, ch. 1, Thousand Oaks: Sage.