Towards a Distributed Software Architecture Evaluation Process - A Preliminary Assessment

Muhammed Ali Babar[1,2], Barbara Kitchenham[1], Ian Gorton[1]

[1] National ICT Australia Ltd; Sydney 1466 NSW, Australia;
{malibaba, barbara.kitchenham, ian.gorton}@nicta.com.au
[2]School of Computer Science & Engineering, University of New South Wales, Sydney 2052 NSW, Australia

ABSTRACT
Scenario-based methods for evaluating software architecture require a large number of stakeholders to be collocated for evaluation sessions. Collocating stakeholders is often an expensive exercise. We have proposed a framework for distributed evaluation process. We present the proposed framework and initial results of a controlled experiment that we ran to assess the effectiveness of the proposed idea.

Categories and Subject Descriptors
K.6.3 [**Software Engineering**]: Software Management – Software process.

General Terms
Management, Experimentation.

Keywords
Architecture evaluation, process improvement.

1. PROBLEM DESCRIPTION
Most of the well-known software architecture evaluation methods are scenario-based [3], for example, Architecture Tradeoff Analysis Method (ATAM) [8] and Performance Assessment of Software Architecture (PASA) [19]. Scenario-based architecture evaluation methods involve several Face-to-Face (F2F) meeting-based activities. The requirement of holding meetings to perform different activities is partially created by the very nature of scenario-based approaches. Software Architecture (SA) is evaluated against the desired quality attributes, which are specified by generating quality-sensitive scenarios. These scenarios are also prioritized according to business goals [4]. Moreover, the effect of a particular quality attribute cannot be analyzed in isolation as quality attributes have positive or negative influences on each other. All these activities require group discussions and group decision making processes, which necessitate meetings. Some approaches even suggest that evaluation meetings be held away from the development sites to avoid potential interruptions or distractions [13].

ICSE'06, May 20-28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

However, co-locating stakeholders can be an expensive exercise, which may also cause scheduling difficulties [16]. Moreover, stakeholders may have to travel if they are geographically separated, which is highly likely case in the wake of increasing trend of global software development [16]. Furthermore, current approaches provide little support to address several issues that characterise F2F meetings, such as conformity pressures, cultural differences and so on [15]. Collectively, such issues may hinder the widespread adoption of SA evaluation practices. In an attempt to find appropriate techniques and technologies to address some of these issues, we studied improvements made in meeting-based activities in other closely related disciplines. Based on our findings, we suggest that groupware applications might provide a cost effective and efficient alternative to F2F meetings.

1.1 Related work and Motivation
Both Software Inspection and Requirements Engineering (RE) communities have argued that groupware applications are an effective mechanism to support geographically distributed teams. They have provided empirical evidence to demonstrates that computer-mediated processes can help minimise meeting costs, maximise asynchronous work and conserve several organisational resources [7, 9, 10]. For example, it has been shown that F2F meetings for software inspections incur substantial cost and lengthen the development process [16]. Studies have indicated that computer tools, including groupware, may improve inspections [17]. Groupware-supported inspection processes have been successfully evaluated as a promising way to minimise meeting costs, maximise asynchronous work and conserve organisational resources [10, 12]. The RE community has also reported a number of studies of successful experiences in adopting groupware systems to improve RE processes [7, 9].

Successful implementation of groupware systems in other software development activities provides a strong motivation to systematically evaluate the pros and cons of using groupware to support the architecture evaluation process in a distributed arrangement for geographically dispersed stakeholders.

2. SOLUTION AND ASSESSMENT
2.1 Groupware-Supported Process
We are mainly interested in finding and evaluating an effective and efficient way of involving physically distributed stakeholders in architecting processes without making them travel, while improving the overall process. We suggest that geographically dispersed stakeholders can effectively perform various activities of software architecture evaluation process, shown in Fig. 1, using Internet-based groupware systems.

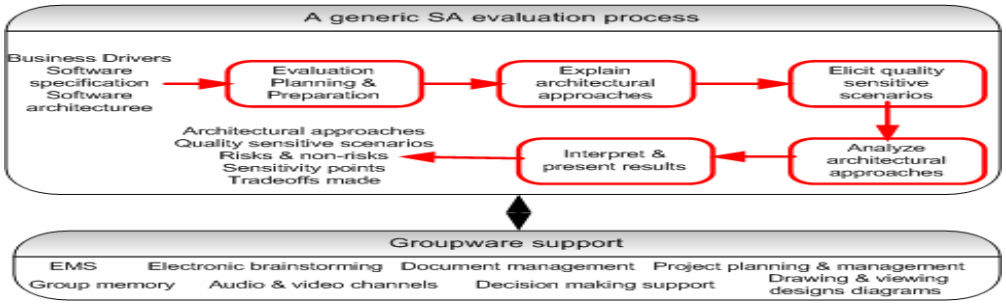

Figure 1: A groupware-supported software architecture evaluation process

Figure 1 shows a groupware-supported distributed software architecture evaluation process. It shows the five activities of a generic process and different tools that a groupware system needs to support the generic process. Each activity of the process can use one or more tools to receive the inputs or produce the outputs, or interact with the participants or other activities. In the following we briefly explain how groupware support can improve the process by providing several benefits. Our process is expected to help manage the review process in a disciplined manner.

The evaluation planning and preparation activity is aimed at preparing the roadmap of the evaluation process, getting the inputs ready, setting the evaluation criteria for the outputs and promoting the review. Groupware support is expected to automate several tasks by helping with identifying and checking the availability of the evaluators and stakeholders based on predefined criteria, suggesting evaluation methods, assigning the tasks and notifying the participants. Groupware support enables a manager to optimize use of the available resources [11].

The groupware-supported process is expected to benefit the architecture presentation activity in several ways. For example, the need for having lengthy architecture presentation meetings is minimized as such presentation and subsequent discussion can easily be conducted using different collaborative tools (e.g. drawing tools, EMs, chat rooms) and stakeholders can see different views of the architecture on their screens and raise their concerns in synchronous or asynchronous discussion sessions. Moreover, our suggested process is supported by a knowledge repository to store and retrieve the design decisions, known patterns and their effects on the desired quality attributes [1]. Online availability of such knowledge can help stakeholders to fully comprehend the architectural approaches being used and raise their concerns with informed arguments.

Scenario elicitation and prioritization is the most expensive and time consuming activity in the evaluation process. That is why we have begun our empirical investigation to assess the effectiveness of groupware support with scenario development activity. Groupware systems provide different tools for brainstorming, organizing, and prioritizing quality sensitive scenarios in distributed arrangement [14]. Our process uses the prepared group method of developing scenarios [5] along with both synchronous and asynchronous communication modes. For example, stakeholders develop and prioritize individual scenarios using a tool. All the stakeholders see each others' scenarios, make comments and discuss each scenario using a discussion board or chat room. Then they use an EMS to integrate their scenarios and brainstorm new ones [15]. Existing approaches have not paid any consideration to social and political issues. Thus, we believe that a groupware-supported process will also address a number of socio-technical issues (i.e. egoism, unfair floor control, and deficiency in a spoken language) [13].

The groupware-supported process can greatly facilitate the task of analyzing architectural approaches. Having performed this activity in physical meetings, our conclusion is that this activity is the least suitable for a face-to-face meeting of large number of stakeholders. We have found that more than a meeting, this activity needs an effective and efficient mechanism of sharing information (with human or computers), finding and evaluating design alternatives, identifying risks and non-risks, making tradeoffs, and storing and retrieving design rationale. The groupware-supported process will also provide a design decision repository to facilitate the decision making process by presenting architectural knowledge in a readily useable format [1].

The proposed process is also expected to greatly improve the task of interpreting the evaluation findings and preparing reports by providing online document management and group memory tools. The evaluation team can search through the annotated information to discuss and clarify ambiguities and debatable points. Groupware support can enable stakeholders to see the review report and presentation online. Findings can be electronically disseminated. Post-review feedback tasks are improved by getting electronic forms filled and directly entered into a database.

A long term benefit of the proposed approach is the possibility to evaluate the performance of the evaluation process with a wide variety of data (such as scenarios, architectural approaches, sensitivity points) along with its available meta-data, which can be processed to develop different metrics. One of our main goals is to design and validate some mechanism of quantifying the benefits of performing evaluations. We believe that capturing and analyzing the data used or generated during SA evaluation is the first step towards that goal.

2.2 Assessment Plan

We decided to use controlled experiments to evaluate the effectiveness of the groupware-supported distributed SA evaluation process and to identify the potential issues associated with it. Instead of evaluating the whole SA evaluation process in a distributed environment, we decided to begin our empirical investigations with the most expensive and time consuming activity of the SA evaluation: developing quality sensitive scenarios. Thus, our controlled experiments are designed to compare the performance of Co-located Groups (CG) and Distributed Groups (DG) based on the quality of the artifacts and scenario profiles, developed.

Null hypothesis: There is no difference between the quality of scenario profiles developed in face-to-face meetings and distributed meetings.

Alternative hypothesis: The quality of scenarios developed in face-to-face meeting is at least 10 units better than the quality of scenarios developed in distributed meetings.

Experiment design – We used an AB/BA cross-over design [18] for our pilot study and a large scale experiment. This is a balanced design in which each experimental unit (in this case, group of three participants) develops change scenario profiles for two systems in different meeting arrangements. Both the order in which the meeting arrangements takes place and type of group interaction, face-to-face and distributed, are assigned randomly. Table 1 shows the experimental design and group assignments.

Table 1. Experimental design and group assignments

Treatments / Material	Face-to-face arrangement	Distributed arrangement
Zwiki system	A = 25 groups of 3 members	B = 25 groups of 3 members
InspectAnyWhere	B	A

The independent variable manipulated by this study is type of meeting arrangement (group interaction), with two treatments, F2F meeting and distributed meeting. **The Dependent variable** is quality of the scenario profiles developed.

Participants – Based on the results of power analysis of the pilot study data, 50 experiment units (groups of 3 subjects) were considered a sufficient sample size. Our study was part of a course in which 159 students were enrolled. **Training** – there were two lectures on developing change scenarios, architecture evaluation, quality attribute. The participants also received training on using a collaborative tool used to support the distributed meetings. **Material** - the participants used SRS for two different applications, a web-based content management system called *Zwiki* and a web-based software inspection support tool that we call *InspectAnyWhere*. We prepared simplified versions of SRS and descriptions of these systems along with some screen shots to provide a clear picture of the systems.

Collaborative Tool - the members of DG were required to brainstorm and structure their scenario profiles in a distributed arrangement using a web-based groupware system. We selected a generic collaborative application, *LiveNet*, based on its features and availability for research purposes [6]. **Measuring quality of scenario profiles** - in order to evaluate the effectiveness of a distributed arrangement compared with face-to-face, we need to compare the quality of the scenario profiles, developed by both DGs and CGs. We used a method of ranking scenario profiles to measure their quality. In order to use this method, the actual profiles for each group must be re-coded into a standard format for analysis. The quality of each of the recoded profiles is evaluated by comparison with a "reference profile" constructed from all the recoded profiles. The method is described in [5].

Procedure – The participants were randomly assigned to groups of three. The groups were randomly assigned to distributed and F2F meeting arrangements. Participants received the SRS for the Zwiki system and were asked to developed change scenarios individually for 15 minutes. When 15 minutes elapsed, the participants were asked to submit their individual scenario profiles and get a print of their scenario profiles as well. All the participants were asked to join their respective groups and develop group scenarios for 45 minutes. Once 45 minutes elapsed, the group scenario profiles were collected.

After a recess period (one week in our experiment), the process started again using the SRS for the *InspectAnyWhere*. The only difference in the process was the groups which had worked in a F2F arrangement for content management system were asked to work in a distributed arrangement and groups which had worked in the distributed arrangement were asked to work in the F2F arrangement. The amount of the time allowed for individual and group tasks was the same. After performing the required tasks, the participants filled a post-session questionnaire to provide both their demographics and their subjective experiences.

Data collection - three sets of data were collected; the individual scenario profiles, group scenario profiles, and questionnaires. Although our results are based on the comparison of group scenario profiles, we needed both individual as well as group profiles to develop the reference profiles. Each participant also filled out a questionnaire, which was designed to collect information on the participants' attitude towards face-to-face verses distributed SA evaluation meetings.

Validity threats - We considered four main threats to internal validity in the design of this experiment: selection effects, maturation effects, instrumentation effects, and presentation effects. We have considered three threats to external validity: participant representativeness, instrumentation representativeness and process representativeness. For details, see [2].

2.3 Preliminary Results

We gathered 150 scenarios from 216 profiles for the Zwiki system and 152 scenarios from 207 profiles for the InspectAnyWhere system. We developed two reference profiles, one for each system, to rank the scenario profiles. The process of developing a reference profile to measure the quality of the scenarios has been extensively documented in [5]. Each scenario profile was scored by comparing it with the reference profile. There were 43 groups, which participated in both sequences of the study (AB/BA). However, some groups experienced changes in terms of group membership because some participants could not join the same group for the second sequence due to scheduling problems, some members dropped out and new members were assigned to those groups, or a few participants did not take part in the first sequence but they turned up for the second sequence (all kind of changes are considered as dropouts for analysis purpose). There were only 32 groups, which did not experience any change in their membership over the whole experiment. Out of these 32 groups, 18 groups developed scenarios in distributed arrangement first and F2F second, while 14 groups developed scenarios in F2F arrangement first and distributed second.

Since we had unequal sample size because of drop outs and changes in the group membership, we performed statistical analysis of the results using the treatment group means and adjusted within-groups variance (see [18], Section 3.6). Table 2 presents the results of statistical analysis of the data. The result of

the analysis indicates that based on the 32 observations the mean effect of the treatment (i.e. the mean difference between the quality of scenario profiles from DGs and CGs) after adjustment for the period effect is 209.55 with a standard error of 36.88. This value is significantly different from zero (p=0.032, 95% confidence interval 1.34.24 to 284.87). This result suggests that the groupware supported distributed meeting is superior than a F2F meeting in terms of the quality of the scenario profiles developed in each type of meeting arrangement.

Table 2. Result of statistical analysis

Cross-Over analysis	Coef.	Std. Err.	t	P>|t|	95% Conf. Interval
Period Effect	82.84	36.88	2.246	0.032	7.52,158.15
Treatment Effect	209.55	36.88	5.682	0.000	134.24, 284.87

We are continuing further analysis of our data for sensitivity analysis that is expected to enable us understand if there has been any bias introduced because of drop outs. For this purpose, we will analyze the data as if there were two parallel experiments instead of one cross-over experiment. We are also analyzing the questionnaire-based data to determine the satisfaction level of the participants with each type of meeting arrangement and to identify other socio-psychological issues of using groupware systems for supporting architecture evaluation process.

3. IMPLICATIONS AND OUTLOOK

Like any process of software development lifecycle, architecture evaluation process faces new challenges posed by the increasing trend of distributed software development. Current processes do not scale up to the needs of geographically dispersed stakeholders. Hence, new processes and support mechanism are necessary. We propose that groupware system can enable geographically dispersed stakeholders to participate in architecture evaluation process. The initial analysis of the data gathered through a controlled experiment supports our proposed solution. These results are consistent with the empirical findings of the use of groupware systems in other activities of software development, i.e. RE [7, 9] and software inspection [10].

Such positive results should encourage researchers and practitioners to identify and assess the changes required to incorporate groupware support for architecture evaluation process. Our work may also provide research directions to groupware systems developers to design and developed groupware systems aimed at architecture evaluation process. Our data analysis is continuing and final results are expected to identify socio-psychological aspects of using groupware systems to support architecture evaluation in distributed arrangement.

4. REFERENCES

[1] Ali-Babar, M., et al., Capturing and Using Software Architecture Knowledge for Architecture-Based Software Development. in *Proc. of the 5th International Conference on Quality Software*, (2005).

[2] Ali-Babar, M., et al. *Design and Evaluation of a Research Program for Distributed Software Architecture Evaluation*. Technical Report No. 0417, University of New South Wales, Sydney, 2004

[3] Ali-Babar, M., et al., A Framework for Classifying and Comparing Software Architecture Evaluation Methods. in *15th Australian Software Engineering Conference*, (Melbourne, Australia, 2004).

[4] Bass, L., et al. *Software Architecture in Practice*. Addison-Wesley, 2003.

[5] Bengtsson, P. and Bosch, J. An Experiment on Creating Scenario Profiles for Software Change. *Annals of Software Engineering, 9*. pp. 59-78.

[6] Biuk-Aghai, R.P. and Hawryszkiewyez, I.T., Analysis of Virtual Workspaces. in *Proceedings of the Database Applications in Non-Traditional Environments*, (Japan, 1999).

[7] Boeham, B., et al. Developing Groupware for Requirements Negotiation: Lessons Learned. *IEEE Software, 18* (3). pp. 46-55.

[8] Clements, P., et al. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley, 2002.

[9] Damian, D.E., et al. Using Different Communication Media in Requirements Negotiation. *IEEE Software, 17* (3). pp. 28-36.

[10] Genuchten, M.V., et al. Supporting Inspection with an Electronic Meeting System. *Journal of Management Information Systems, 14* (3). pp. 165-178.

[11] Griffith, T., et al. Virtualness and Knowledge in Teams: Managing the Love Triangle of Organisations, Individuals, and Information Technology. *MIS, Quarterly, 27* (2). pp. 265-287.

[12] Halling, M., et al., Tailoring a COTS group support system for software requirements inspection. in *Proc. of the 16th International Conference on Automated Software Engineering*, (2001).

[13] Kazman, R. and Bass, L. Making Architecture Reviews Work in the Real World. *IEEE Software, 19* (1). pp. 67-73.

[14] Nunamaker, J.F., et al. Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings. *Journal of Management Information Systems, 13* (3). pp. 163-207.

[15] Nunamaker, J.F., et al. Electronic Meeting Systems to Support Group Work. *Communication of the ACM, 34* (7). pp. 40-61.

[16] Perry, D.E., et al. Reducing inspection interval in large-scale software development. *IEEE Transactions of Software Engineering, 28* (7). pp. 695-705.

[17] Sauer, C., et al. The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research. *IEEE Transactions on Software Engineering, 26* (1). pp. 1-14.

[18] Senn, S. *Cross-Over Trials in Clinical Research*. John Wiley & Sons Ltd., 2002.

[19] Williams, L.G. and Smith, C.U., PASA: A Method for the Performance Assessment of Software Architecture. in *Proc. of the 3rd Workshop on Software Performance*, (Rome, Italy, 2002).