The Impact of Group Size on Software Architecture Evaluation: A Controlled Experiment

Muhammad Ali Babar[1], Barbara Kitchenham[2]
[1]*Lero, University of Limerick, Ireland,* [2]*University of Keele*
malibaba@lero.ie, Barbara@cs.keele.ac.uk

Abstract

Background*: An important element in scenario-based architecture evaluation is the development of scenario profiles by stakeholders working in groups. In practice groups can vary in size from 2 to 20 people. Currently, there is no empirical evidence about the impact of group size on the scenario development activity.*
Goal*: Our experimental goal was to investigate the impact of group size on the quality of scenario profiles developed by different sizes of groups.*
Experimental design*: We had 165 subjects, who were randomly assigned to 10 groups of size 3, 13 groups of size 5, and 10 groups of size 7. Participants were asked to develop scenario profiles. After the experiment each participant completed a questionnaire aimed at identifying their opinion of the group activity.*
Result: *The average quality score for group scenario profiles for 3 person groups was 362.4, for groups of 5 person groups was 534.23 and for 7 person groups was. 444.5. The quality of scenario profiles for groups of size 5 was significantly greater than the quality of scenario profiles for groups of size 3 (p=0.025), but there was no difference between the size 3 and size 7 groups. However, participants in groups of size 3 had a significantly better opinion of the group activity outcome and their personal interaction with their group than participants in groups of size 5 or 7.*
Conclusion: *Our results suggest that the quality of the output from a group does not increase linearly with group size. However, individual participants prefer small groups. This means there is a trade-off between group output quality and the personal experience of group members.*

Keywords: Architecture evaluation, controlled experiments, scenario development, quality attributes.

1. Introduction

Software Architecture (SA) evaluation is a relatively new technique that aims to improve the quality of software intensive systems [1, 2]. The main objective of architecture evaluation is to address quality requirements at the software architecture level [1]. There are various techniques to assess the potential of the chosen architecture to deliver a system capable of satisfying desired quality requirements and identify risks. Most of the well-known approaches are scenario-based [3] such as Architecture Tradeoff Analysis Method (ATAM) [4], Software Architecture Analysis Method (SAAM) [5] and Architecture-Level Maintainability Analysis (ALMA) [6].

Scenario-based software architecture evaluation involves a number of stakeholders working together in groups. In practice, group size can vary from two to 20 stakeholders [7]. However, currently there is no empirical evidence concerning the impact of group size on group performance. As part of our overall research program investigating mechanisms for improving the effectiveness and efficiency of software architecture evaluation, we undertook an exploratory experiment to investigate the impact of group size on group performance for developing quality sensitive scenarios during software architecture evaluation.

The paper is organized as follows. In the next section, we briefly review the software architecture evaluation process. We describe experiment details in section 3. We present the results of our experiment in Section 4. We discuss our results in Section 5 and present our conclusions in Section 6.

2. Background and Motivation

In this section, we briefly describe the software architecture evaluation process and roles of scenarios and stakeholders, which provide the motivation for the study reported in this paper.

2.1 Software architecture evaluation

A quality attribute is a non-functional requirement of a software system such as reliability, modifiability, performance, and usability. According to [8], software quality is the degree to which software possesses a

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.38

desired combination of attributes. The quality attributes of large software intensive systems are largely determined by the system's software architecture [1]. Since software architecture plays a vital role in achieving system wide quality attributes, it is very important to evaluate a system's architecture with regard to desired quality requirements as early as possible. The principle objective of software architecture evaluation is to assess the potential of the chosen architecture to deliver a system capable of fulfilling required quality requirements and to identify any potential risks [9]. Additionally, it is quicker and less expensive to detect and fix design errors during the initial stages of the software development.

Several methods and techniques have been applied to ensure that the quality concerns are addressed at the architecture level. Scenario-based software architecture evaluation methods such as ATAM, SAAM, and ALMA, are considered relatively mature and established as they have been widely applied and rigorously validated in various domains [3].

2.2 Scenario Profiles

Scenarios have been used for a long time in research and practice of many disciplines (military and business strategy, decision making, etc). A scenario is a textual specification of a quality attribute required of a system [1]. The software engineering community uses scenarios in user-interface engineering, requirements elicitation, performance modeling, and more recently in software architecture evaluation [10].

Table 1: Performance scenario profile example

Quality Factor	Scenario description
Initialization	Must perform all initialization activities within 10 minutes.
Latency	Run simulations with no instantaneous lags greater than five seconds, no average lags greater than three seconds.
Capacity	Run-time simulation with debug enabled.
Latency	Finish data collection within 30 seconds of simulation termination.
Throughput	Collect data from three network sensors within 10 seconds.

Scenarios make it possible to evaluate most quality attributes, e.g., we can use scenarios that represent failure to examine availability and reliability, scenarios that represent change requests to analyze modifiability, or scenarios that represent security threats to analyze security. Moreover, scenarios are normally very concrete, enabling the system user to understand their detailed effect [11]. A set of scenarios is called a scenario profile as shown in Table 1

It is important to note that the use of the term 'scenarios' in software architecture evaluation is different to the term used in Object-Oriented design methods where the term "scenarios" generally refers to use-case scenarios, i.e., scenarios describing system's functions. Instead, quality attribute scenarios describe an action, or sequence of actions that might occur related to the system to be built using a particular architecture. The description of a quality attribute scenario includes a stimulus/response pair of which response part is usually measurable behavior such as x number of transactions in y time period. Hence, a change scenario describes a certain maintenance task or a change to be implemented [12].

2.3 Architecture Evaluation and Stakeholders

The role of stakeholders is vital in scenario-based software architecture evaluation methods as scenarios are mainly gathered from stakeholders; and stakeholders need to be satisfied with the proposed architectural solution. Clements et al. [2] describe the active participation of stakeholders in the architecture evaluation process as absolutely essential for a high-quality evaluation. Parnas [13] regards the presence of wrong people in the design review sessions as the one of the major problems with the conventional design review approaches.

One of the challenges of architecture evaluation process is to decide the appropriate size and formation of an evaluation team. Since there is no consensus on a suitable team size, architectural evaluation sessions may have varying sizes (2 to 20 or more participants) of teams [7]. Group size has also been researched for software inspection teams [14, 15]. Although, there is no consensus on an optimal size of inspection team, many researchers agree that the benefits of an additional inspector diminish with growing team size [14, 16]. We also think that larger architecture evaluation team size may not be justifiable in terms of output quality. However, to date, there has been no study on the impact of team size on any of the activity of the architecture evaluation process. We believe this is an important research topic which will help managers to optimize the resources allocated to the software architecture evaluation activities.

3. Experiment Description

This section discusses the objectives, research questions, experimental design and logistics of the empirical study reported in this paper.

3.1 Introduction

The main objective of this study was to gain some understanding of the impact of group size on the outcome of a software architecture evaluation exercise. Developing quality sensitive scenarios is the most expensive and time consuming activity of the software architecture evaluation process [1, 12]. Thus, this controlled experiment was aimed at understanding the impact of different sizes of groups (i.e., 3, 5, and 7 members) on the quality of scenario profiles.

Since there has been no previous research on the impact of group size on the quality of the output in the context of software architecture evaluation, this experiment can be considered an exploratory study aimed at finding answers to the following research questions:

1. Is there any difference in quality of scenario profiles created by different sizes of groups?
2. How does the size of a group affect the satisfaction of the participants with the process and the outcomes, and sense of personal contribution to the outcome?

3.2 Experiment design and task

The experiment design was a randomized design, which used the same experimental materials for all treatments and assigned the subjects randomly to groups of three different sizes (3, 5, and 7). The assignment of individuals to all three treatment groups was randomized using sort card randomization. Table 2 shows the experimental design.

Table 2: Experimental design and group assignments

Material \ Treatments	Group of 3	Group of 5	Group of 7
Lab booking system	10 groups	13 groups	10 groups

The experimental task was to generate quality-sensitive scenarios for characterising quality requirements of a particular system. The participants of this experiment were asked to construct software change scenarios for a lab booking system. They were given software requirements specifications (SRS) of the system, which was well-known to them. Because scenario profiles for architecture evaluation need to be concrete, we decided to select only one quality attribute, modifiability, for developing scenario profiles. However, we believe that the experiment design allows the results to be applicable to selected scenario profiles for other quality attributes [12].

This research used a two-stage process of developing scenario profiles. According to this process, a group of stakeholders are gathered in Quality Attribute Workshop (QAW) meetings [17] which aim to develop scenario profiles to characterize the quality attributes required of the system whose architecture is to be analysed. However, each of the stakeholders develops his/her individual scenario profile before meeting with the group and developing a group scenario profile. The two-staged process (Shown in Figure 1) was assessed as the most effective and efficient process in a previous experiment [12].

The **independent variable** manipulated by this study is the size of a group (number of members) for generating quality sensitive scenarios.

The **dependent variable** is the quality of scenario profiles developed by groups of different sizes (3, 5, and 7 members).

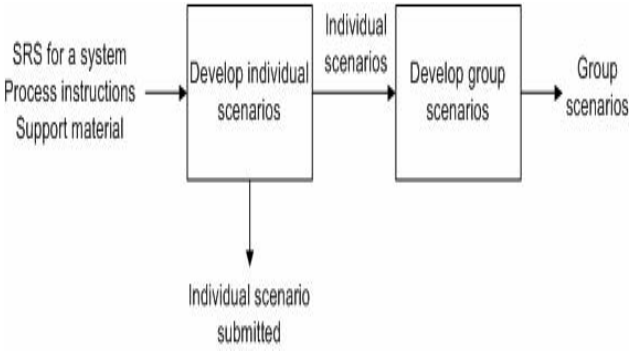

Figure 1: Two-staged scenario development process.

3.3 Participants and training

The participants were 3rd and 4th year students of software engineering and computer engineering degrees at the University of New South Wales, Australia. In order to motivate the participants, the experimental tasks were parts of assessment tasks of the course. However, the participants were explicitly not advised that the assessment tasks were parts of a formal experiment in order to avoid any spurious effect (Avoiding the "Good Subject" effect [18]) as a result of the participants being aware of being studied. There were 165 students enrolled in the course. The participants had a strong technical background, varying

degrees of work experience, and familiarity with the quality attribute sensitive scenarios.

For training purposes, there were two lectures, each of two hours, covering the software architecture evaluation process, and current methods of developing change scenarios to specify quality attributes. Participants were also provided with support material at the beginning of experiment. They were quite familiar with the Lab booking system as they had been using that system to book the terminals for their daily use since their entrance to the degree program at the university.

3.4 Experimental Instruments

3.4.1 Software requirements specifications

This study used the SRS for a dumb terminal based system for booking computer laboratories of the school of computer science and engineering at the University of New South Wales, Australia. This system is the only way of booking computer terminals for day-to-day use of the school's computers. We prepared a simplified version of SRS and descriptions of this system. We did not provided any screen shots of the system as the participants were very familiar with the system as they had been using it on daily basis since their entry to the course. However, there was a verbal briefing on various aspects of this system. The SRS document consisted of 2 A4 size pages and reviewed by course delivery team.

3.4.2 Measuring quality of scenario profiles

In order to assess the performance of different sizes of the groups during the scenario development activity, we needed to compare the quality of the scenario profiles, i.e., a set of scenarios, developed by three groups of different sizes (i.e., 3, 5, and 7 members). We needed a comparison approach to assess quality of scenario profiles created by each size of the groups. Bengtsson [12] proposed a method of ranking scenario profiles to measure their quality by comparing each scenario profile with a "*reference scenario profile*". This method has successfully been used to measure the quality of scenario profiles generated to evaluate software architecture in other experiments [12, 19], thus we consider this approach appropriate for our context: comparing the quality of scenario profiles developed in our controlled experiment.

Using this method, the actual scenario profile for each individual and group must be recoded into a standard format for analysis. The quality of each scenario profile is evaluated by comparison with a "*reference scenario profile*" constructed from all the unique scenarios found in the recoded scenario profiles (see Section 4.1).

3.4.3 Post-experiment questionnaire

At the end of the experiment, each participant completed a questionnaire. The post-experiment questionnaire was designed to collect information about the participants' satisfaction with the meeting process, quality of discussion, and solution, and commitment to and confidence in the solution. Most of the questions required the participants to respond by circling a choice on a four point scale (i.e., Strongly disagree, disagree, Agree, Strongly agree). The questionnaire also collected demographic data (see Appendix A).

3.5 Experimental validity

3.5.1 Threats to Internal Validity

Threats to internal validity are those factors that may affect the value of the dependent variables apart from the independent variable [20, 21]. Wholin et al. [22] identify four main threats to internal validity: selection effects, maturation effects, instrumentation effects, and presentation effects. Only the selection effect was the relevant to our experimental design and we took appropriate measure to address this threat.

A selection effect is any difference between individuals in different treatment groups such that differences in dependent variables ensue. We minimized the differences between individuals in the different treatment groups by randomly assigning individuals participants to different teams.

Another threat to the internal validity of our experiment is the method used to measure the quality of the scenarios. The method has been developed and validated for another experiment and various threats to its internal validity have been discussed and addressed in [12]. However, one of the potential threats, skill, knowledge, and bias of reference profile builder, associated with this method should be addressed for each experiment. We addressed this issue by having two researchers create the reference scenario profile separately and exchanged their respective reference scenario profiles to review. During the review process, each marker could add more scenarios or split old scenarios in the reference profile developed by the other maker. Any disagreement regarding the scenario profile was discussed and resolved before building a reference profile for marking each scenario profile.

3.5.2 Threats to external validity

Threats to external validity are those that may limit the applicability of the experimental results to industry practices [21]. The experiment considered three threats to external validity: participant representativeness, instrumentation representativeness and process representativeness.

Participant representativeness is an issue because the participants were the 3rd and 4th year undergraduate students with predominantly technical background. The participants had limited experience of developing scenarios for quality attributes. This may seem a threat to the applicability of the results of our study. However, we do not consider that it is a crucial issue as there are not many organizations that have institutionalized systematic and formal process of evaluating software architectures. Nor do organizations provide extensive training to their employees for participating in software architecture evaluation process or developing scenarios to characterise quality attributes. Additionally, there is support for the use of software engineering students instead of professionals under certain conditions (e.g., research-in-the-small, initially evaluative studies etc.) [23, 24].

However, the fact that most of the participants of our study had technical background, computer science or engineering, may pose a real threat as software architecture evaluation in an industrial setting may involve different classes of participants from technical as well as non-technical fields. Such stakeholders are expected to have much more domain knowledge than the participants in this study. However, this aspect is an unavoidable factor when conducting an experiment with student participants.

To minimize threat to instrumentation representativeness, we used SRS for a system that is being used in a real world situation and the particpants were the regular users of that system. Another point to note is that the SRS provided to the participants may be considered atypical of industrial SRS in terms of length and complexity. The participants used a relatively short and simple SRS document. However, the scenario profile development method used can work equally well with large and more complex SRS provided participants are given longer time period to complete the activity.

Finally, there may be a threat to the external validity if the scenario development process used in our experiment is not representative of industrial practices for developing scenarios for software architecture evaluation. The participants of our study followed a scenario development process that is quite similar to the one used in Quality Attribute Work [17], which is a means of gathering scenarios from stakeholders for evaluating software architecture using ATAM [4]. The two-staged scenario development process has been evaluated as the most effective one [12].

3.6 Experiment operation

The flow of the experiment is shows in Figure 2. There are several sources of recruiting participants for an experiment such as practicing software engineers, students in postgraduate or undergraduate courses offered by tertiary educational institutes [24]. We recruited the participants for this experiment from an undergraduate course on total quality management offered at the University of New South Wales, Australia. The experiment was conducted as a part of scenario development workshop during that course.

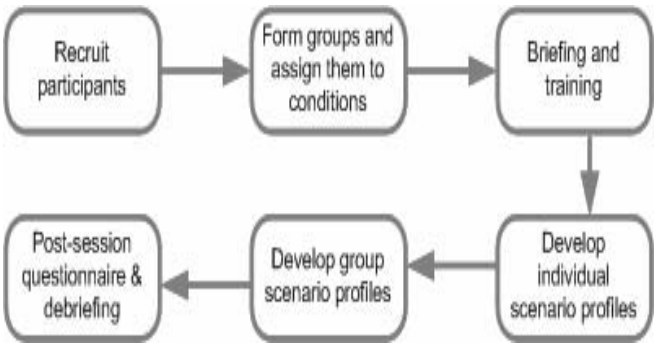

Figure 2: A control flow diagram of the experiment steps.

Table 3 shows the execution plan. The experiment started with a 30 minutes session designed to brief the participants about the process to be followed and provide an overview of the system for which the participants were supposed to create scenarios, the process of generating scenarios. Our study did not require the participants to have an extensive knowledge of and experience in generating quality-sensitive scenarios. The duration and format of our training was designed to make the participants representative of most of the stakeholders involved in generating scenarios in industrial setting, where stakeholders normally receive minimum training.

The participants were given a simplified version of requirements for a lab booking system and asked to developed system change scenarios individually for 15 minutes. When 15 minutes of time had elapsed the participants were asked to submit an electronic copy of their scenario profiles and also get a printout of their scenario profiles before joining their respective groups for developing group scenario profiles.

Table 3: Experimental execution plan

Amount of time	Group of 3	Group of 5	Group of 7	SRS
30 minutes	A brief introduction to the process and training			
15 minutes	Develop individual scenarios	Develop individual scenarios	Develop individual scenarios	Lab Booking System
45 minutes	Develop group scenarios (Groups in category A)	Develop group scenarios (Groups in category B)	Develop group scenarios (Groups in category C)	
20 minutes	Post-session questionnaire and debriefing			

Participants were asked to develop group scenarios for another 45 minutes. The participants were advised to follow a process to develop group scenarios profile. After 45 minutes had elapsed, each group submitted electronic copy of its scenario profile.

After completing both stages of the scenario profile development process, the participants completed a post-session questionnaire designed to gather subjective opinion of the participants about various aspects of the process of developing scenario profiles in their respective groups. The questionnaire collected quantitative subjective data using closed questions with a four points scale. The experiment finished with a debriefing session, which explained the objectives of the study and answered participants' questions on any aspect of the research.

3.7 Data collection

Three sets of data are important to our study; the individual scenario profiles, group scenario profiles, and questionnaire filled by all the participants at the end of the experiment. Although the quantitative results of this experiment are based on the comparison of group scenario profiles, we needed both individual as well as group profiles to develop the reference profile.

Finally, participants' demographic data and information on their satisfaction with the meeting process, quality of discussion, and solution, and commitment to and confidence in the solution were gathered using a post-experiment questionnaire.

4. Results and Analysis

In this section, first we describe the reference scenario profile construction. Then we present an analysis of the quality of the group scenario profiles. Finally, an analysis of the data gathered from the questionnaire is presented.

4.1 Reference profiles

We gathered 134 unique scenarios from 195 scenario profiles (165 individuals, 33 groups). We lost three data points, individual scenario profiles, as three participants' submission could not be retrieved from the system; nor could we find the hard copies of their individual scenario profiles. We developed a reference profile to rank the scenario profiles developed by the participants. Table 4 shows the top 10 scenarios of the reference scenario profile.

Table 4: Reference profile Top 10 scenarios

	Reference Scenario Profile	F
1	Users need different booking privileges.	69
2	System is available without significant downtime during university sessions.	66
3	System is robust enough to provide fast access/response under heavy use.	64
4	System is compatible with various platforms / operating systems.	58
5	System provides a good mechanism to prohibit unauthorised access.	57
6	System has a user friendly interface.	56
7	System has a web-enabled interface.	49
8	System is modifiable and maintainable, i.e. changes can easily be made without system being offline for long time.	49
9	Booking and terminal information is frequently updated.	40
10	Detailed guide on using the system for various purposes is available online.	31

The process of developing a reference profile to measure the quality of the developed scenario profiles has been extensively documented in [25]. However, we provide a brief description of the process here. To build a reference scenario profile, we identified unique scenarios and put them together. We noted the frequency for each unique scenario by counting the

number of times it had been reported in various scenario profiles. Then, we calculated a score for each scenario profile developed during the experiment by summarizing the frequency of each scenario in the scenario profile.

4.2 Results of statistical analysis

The results of the descriptive statistics analysis on the raw scores for group scenario profiles are presented in Table 5 and Figure 3. It is obvious from Figure 3 that there is a substantial difference in variation between the scores for groups of size 7 and the scores for groups of size 3 and size 5. This is confirmed by the large standard deviation for the groups of size 7 as shown in Table 5. For this reason we use the robust Kruskal-Wallis test to assess whether the scores for the three groups were significantly different from one another. The Kruskal-Wallis test confirmed that the scores for the three groups were significantly different (Chi-squared test statistic = 7.09 with 2 degrees of freedom, p=0.0249).

Since the Figure 3 shows that groups of size 3 and size 5 are relatively similar with respect to variation, so we used a standard "t" test to confirm that the average score for 5 person groups was significantly greater than the average score for 3 person groups (p=0.0008).

Table 5: Summary statistics for each group

	Statistic					
Group Size	Number of groups	Mean	Standard deviation	Median	Max	Min
3 person group	10	362.4	124.24	326.5	531	189
5 person group	13	534.23	104.52	535	679	379
7 person group	10	444.5	232.27	389	881	182

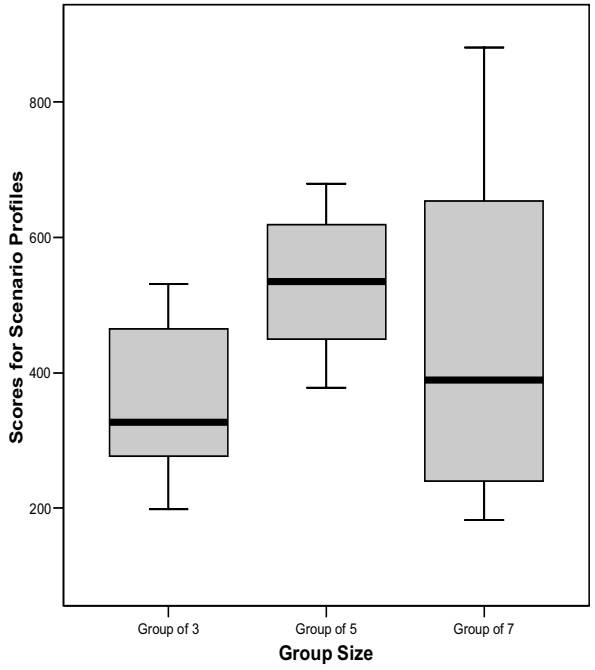

Figure 3: Box plots of score for each groups of different size

A Kruskal-Wallis test of the 3 person groups compared with the 7 person groups was unable to detect any significant difference between the two groups (note this test does not consider differences in variation). Thus, the results indicate that 5 person groups score significantly better than 3 person groups. 7 person groups do not on average produce higher scores than 3 person groups. Furthermore, individual scores are very variable, for example, both the highest performing group and the lowest performing group are 7 person groups.

4.3 Analysis of post-session questionnaire

4.3.1 Preliminary analysis

Each individual completed the questionnaire giving a total of 165 questionnaires. 30 questionnaires were received from the participants in groups of size 3, 65 from participants in groups of size 5 and 70 from participants in groups of size 7.

By following the customary method for handling questions with multiple response choices that address related issues, initially, we treated the ordinal scale responses to question 5 to 14 as if they were interval scale and calculated the correlation among all pairs of variables. The correlation matrix indicated that the questions addressed three separate concepts:

- The overall performance of the process (Q5, Q6, Q7)
- The overall outcome of the process (Q8, Q9, Q10)

- The individual commitment to the group outcome (Q11, Q12, Q13, Q14)

Applying factor analysis to:

- Questions Q5, Q6 and Q7: only one factor was selected that accounted for 72% of the variation. The contribution of each question to the factor was similar. The Cronbach alpha for the three questions was 0.80
- Questions Q8, Q9, Q10: only one factor was selected that accounted for 60% of the variation. The contribution of each question to the factor was similar. The Cronbach alpha for the three questions was 0.67.
- Questions Q11, Q12, Q13, Q14: only one factor was selected that accounted for 51% of the variation. The contribution of each question to the factor was similar. The Cronbach alpha for the four questions was 0.68.

These results suggested that grouping question Q5-Q14 into 3 groups was reasonable. The composition of the first group was strongly supported by the factor analysis and the Cronbach alpha value. The second and third groups were not so strongly supported but were consistent with reducing the dimensionality of the data from 7 variables to 2.

We constructed new variables based on the question subgroups by averaging the response for question in the group. Thus:

- GroupProcess = (Q5+Q6+Q7)/3
- GroupOutcome = (Q8+Q9+Q10)/3
- GroupCommitment = (Q11+Q12+Q13+Q14)/4

4.3.2 Impact of group size on questionnaire-related variables

We analysed the three new variables using nonparametric analysis of variance (Kruskal-Wallis). The results indicated that:

- GroupProcess was not significantly affected by group size (p=0.1012). Overall individuals in all groups thought the process was good (GroupProcess median=3).
- GroupOutcome was significantly affected by group size (p=0.0181). The box plot (Figure 4) shows that overall the outcome was rated highly but individuals in groups of size 3 were slightly more convinced of the quality of the outcome than individual in groups of size 5 and size 7.
- GroupCommitment was significantly affected by group size (p=0.0409). The median value for subjects in three-person groups was 3.5, for subjects in the five-person groups was 3.25, and for subjects in the seven-person groups was 3.25. Thus, overall individuals experienced a positive commitment to their group.

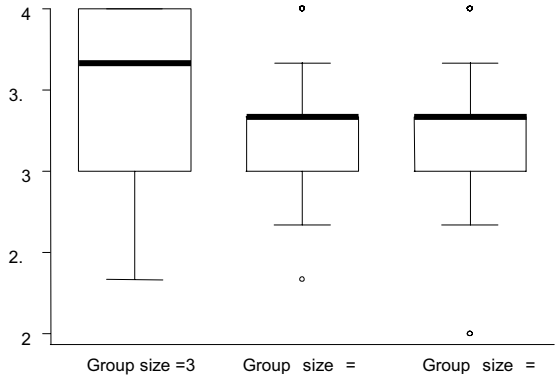

Figure 4: Box plot of GroupOutcome construct

Overall individuals in the smallest group (i.e. three-person groups) were the most satisfied. This shows a marked contrast with the analysis of scenario profile quality which indicated that the best quality outcome was obtained from five-persons groups.

5. Discussion

Our results suggest that there is an optimum group size (i.e. the quality of the output from a group does not increase linearly with group size). However, individual participants prefer small groups. This means there is a trade-off between group output quality and the personal experience of group members.

Although our participants were 3rd and 4th year students we have no reason to believe that stakeholders in an industrial situation would exhibit very different patterns with respect to group interaction. The students were more technically-oriented than industrial stakeholders, who may come from different backgrounds including business, sales, marketing and others. However, the amount of training provided to the participants of this study was similar to that of provided to the industrial stakeholders. The participants used SRS that was simpler than would be the case in an industrial architecture evaluation exercise but they were give proportionally less time for developing scenarios for quality attributes.

We believe these results are important because they show clearly that increasing group size does not necessarily improve the quality of the group outcome in the context of software architecture meetings.

In previous experiments, we have investigated the issue of geographically dispersed teams and shown that such teams can be more effective than collocated

teams, although individual participants preferred face-to-face meetings [26, 27]. These results suggest an effective process for software evaluation meetings, when it is important to involve a large number of stakeholders, might be:

1. Assuming that only a small number of stakeholders are likely to be collocated, use several small groups at different locations to develop preliminary quality sensitive scenario profiles.
2. Use a distributed meeting arrangement to develop an integrated scenario profile from the preliminary profiles produced by the distributed groups [26, 27].

6. Conclusion

One of our main research goals is to improve software architecture processes by developing and empirically assessing various support mechanisms. We aim to reduce the time, resources and skills required to effectively and efficiently assess software architectures with respect to desired quality attributes. This paper reports the impact of team size on group performance.

Our results show a strong non-linear relationship between group size and performance which indicates that very large software architecture evaluation workshops may have a significant negative impact on the quality of scenario profiles. Not only do large groups appear to deliver poorer quality profiles, they are also less agreeable for participants.

When a software architecture evaluation requires a large numbers of stakeholders, we believe single large meetings should be avoided. Instead of one large meeting, group activities should be organized as a number of small-size group meetings. Fortunately, this approach would integrate well with distributed software architecture evaluations process [26, 27]. Architecture evaluations could be organized as small distributed groups. This would allow participants to benefit from the advantages of face-to-face group working in small groups while avoiding the overhead of collocating large numbers of stakeholder. Further research is necessary to confirm the feasibility of this approach.

7. Acknowledgement

We thank the experiment participants. Cynthia Wang helped in preparing reference scenario profile and marking. Both authors were working with NICTA when this paper was written.

8. Appendix A: Post Experiment Survey

Tutorial Name: Group Size:

1. Gender: Male ❑ Female ❑
2. Age : < 20 ❑ 21-25 ❑ 26-30 ❑ > 30 ❑

Please answer the following questions based on your experience at the university as well as at work:

3. Experience of working in a team (Number of years) ----------
4. What is the average size of the teams that you have worked in?

2-3 members ❑ 4-5 members ❑ 6-10 member ❑ > 10 members ❑

5. Our group decision making process was **efficient**?

Strongly disagree	Disagree	Agree	Strongly agree

6. Our group decision making process was **well coordinated**?

Strongly disagree	Disagree	Agree	Strongly agree

7. Our group decision making process was **effective**?

Strongly disagree	Disagree	Agree	Strongly agree

8. Our group decision making process was **fair**?

Strongly disagree	Disagree	Agree	Strongly agree

9. The outcome of the group discussion was **satisfactory**?

Strongly disagree	Disagree	Agree	Strongly agree

10. How satisfied are you with **the quality** of your group solution?

Not at all	Little	Very much	Completely

11. To what extent does the final solution reflect **your input**?

Not at all	Little	Very much	Completely

12. To what extent do you **feel committed** to the group solution?

Not at all	Little	Very much	Completely

13. To what extent are you **confident** that the group solution is correct?

Not at all	Little	Very much	Completely

14. To what extent do you feel **personally responsible** for the correctness of the group solution?

Not at all	Little	Very much	Completely

9. References

[1] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice. 2 ed. 2003: Addison-Wesley.
[2] P. Clements, R. Kazman, and M. Klein, Evaluating Software Architectures: Methods and Case Studies. 2002: Addison-Wesley.

[3] M. Ali-Babar, L. Zhu, and R. Jeffery, A Framework for Classifying and Comparing Software Architecture Evaluation Methods, *Proceedings of the 15th Australian Software Engineering Conference*, 2004.
[4] R. Kazman, M. Barbacci, M. klein, and S.J. Carriere, Experience with Performing Architecture Tradoff Analysis, *Proceedings of the 21th International Conference on Software Engineering*, 1999.
[5] R. Kazman, L. Bass, G. Abowd, and M. Webb, SAAM: A Method for Analyzing the Properties of Software Architectures, *Proceedings of the 16th International Conference on Software Engineering*, 1994.
[6] N. Lassing, P. Bengtsson, J. Bosch, and H.V. Vliet, Experience with ALMA: Architecture-Level Modifiability Analysis, *Journal of Systems and Software*, 2002. **61**(1): pp. 47-57.
[7] L. Bass, Conducting Architecture evaluation with different sizes of groups, *(Personal communicatoin)*, 2004.
[8] IEEE, IEEE Standard 1061-1992, Standard for Software Quality Metrics Methodology. 1992, New York: Institute of Electrical and Electronic Engineers.
[9] N. Lassing, D. Rijsenbrij, and H.v. Vliet, The goal of software architecture analysis: Confidence building or risk assessment, *Proceedings of First BeNeLux conference on software architecture*, 1999.
[10] R. Kazman, G. Abowd, L. Bass, and P. Clements, Scenario-Based Analysis of Software Architecture, *IEEE Software Engineering*, 1996. **13**(6): pp. 47-55.
[11] N. Lassing, D. Rijsenbrij, and H.v. Vliet, How Well can we Predict Changes at Architecture Design Time? *Journal of Systems and Software*, 2003. **65**(2): pp. 141-153.
[12] P. Bengtsson and J. Bosch, An Experiment on Creating Scenario Profiles for Software Change, *Annals of Software Engineering*, 2000. **9**: pp. 59-78.
[13] D.L. Parnas and D.M. Weiss, Active Design Reviews: Principles and Practices, *Proceedings of the 8th International Conference on Software Engineering*, Aug., 1985.
[14] S. Biffl and W. Gutjahr, Influence of team size and defect detection technique on inspection effectiveness, *Proceedings of the 7th International Software Metrics Symposium*, 2001.
[15] S. Boodoo, K.E. Emam, O. Laintenberger, and N. Madhavji, The Optimal Team Size for UML Design Inspections, *Tech Report NRC-44149*, Institute for Information Technology, National Research Council Canada, 2000.
[16] P. Yetton and P. Bottger, The Relationships among Group Size, Member Ability, Social Decision Schemes, and Performance, *Organizational Behavior and Human Performance*, 1983. **32**(1): pp. 145-159.
[17] M.R. Barbacci, et al., Quality Attribute Workshops (QAWs), *Tech Report CMU/SEI-2003-TR-016*, SEI, Carnegie Mellon University, USA., 2003.
[18] R.L. Rosnow and R. Rosenthal, People Studying People: Artifacts and Ethics in Behavioral Research. 1997: W.H. Freeman and Company.
[19] M. Ali-Babar, B. Kitchenham, and P. Maheshwari, The Value of Architecturally Significant Information Extracted from Patterns: A Controlled Experiment, *Proceedings of the 17th Australian Software Engineering Conference*, 2006.
[20] L.E. Toothaker and L. Miller, Introductory Statistics for the Behavioral Science. 1996, Pacific Grove, CA, USA: Brooks/Cole Publishing Company.
[21] B.A. Kitchenham, et al., Preliminary guidelines for empirical research in software engineering, *IEEE Transactions on Software Engineering*, 2002. **28**(8): pp. 721-734.
[22] C. Wohlin, et al., Experimentation in Software Engineering: An Introduction. 2000: Kluwer Academic Publications.
[23] N. Fenton, S.L. Pfleeger, and R.L. Glass, Science and substance: a challenge to software engineers, *IEEE software*, 1994. **11**(4): pp. 86-95.
[24] M. Host, B. Regnell, and C. Wholin, Using Students as Subjects - A Comparative Study of Students and Professionals in Lead-Time Impact Assessment, *Empirical Software Engineering*, 2000. **5**: pp. 201-214.
[25] P. Bengtsson. Architecture-Level Modifiability Analysis. Ph.D. Thesis. Blekinge Institute of Technology, Sweden, 2002.
[26] M. Ali-Babar, B. Kitchenham, and R. jeffery, Distributed Versus Face-to-Face Meetings for Architecture Evaluation: A Controlled Experiment, *Proceedings of the International Symposium on Empirical Software Engineering*, 2006.
[27] M. Ali-Babar, et al., An empirical study of groupware support for distributed software architecture evaluation process, *Journal of Systems and Software*, 2006. **79**(7): pp. 912-925.