A Controlled Experiment on Team Meeting Style in Software Architecture Evaluation

Dietmar Winkler Stefan Biffl Christoph Seemann

Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
Favoritenstrasse 9-11/188, 1040 Vienna, Austria
{Dietmar.Winkler, Stefan.Biffl}@tuwien.ac.at

Software architecture change can have a major impact on product and project quality. Software architecture evaluation systematically investigates architecture variants with respect to their quality attributes, e.g., modifiability and maintainability of a software system. Scenarios and system properties are key elements to focus architecture evaluation on most likely future changes in the software architecture. Scenario elicitation processes typically include individual scenario brainstorming and a team meeting of all relevant stakeholders to create an aligned list of important scenarios. A major research question is which meeting style, face-to-face meetings, nominal (i.e., non-communicating) teams, or distributed tool-supported team meetings are most effective. In this paper we report on a controlled experiment investigating the meeting style of scenario brainstorming processes. Major result is that face-to-face meetings outperformed in the experiment tool-supported meetings in finding more important scenarios.

Keywords: Software Architecture Evaluation, Team Meeting Effectiveness, Scenarios, Controlled Experiment

1. INTRODUCTION

Well-defined and stable software architecture is a success-critical aspect in software engineering because architecture provides the foundation for the software product [6]. However, in practice the architecture is often not described very well and fulfils only some stakeholder requirements [22], which may lead to a software product that does not address all relevant non-functional requirements [5]. Changes, affected by customer change requests and defects can have a major impact on software architecture and can lead to high rework effort.

Non-functional quality requirements, e.g., modifiability and maintainability, have to be considered during designing the software architecture early in the development process. Based on different functional and quality requirements various architecture variants are available. Therefore, a key task of software architects is to select of an appropriate architecture variant regarding candidate changes of requirements during development and operation.

Software architecture evaluation, e.g., ATAM [17], SAAM [18], ALMA [7], and PASA [24], systematically investigates architecture variants with respect to their quality attributes and enables the selection of best-practice software architectures in a given project context [4]. Architecture reviews support engineers to en-

sure the quality of the selected architecture early in the software development process and help engineers in efficiently evaluating the underlying architecture of a software product [6]. Scenarios have been successfully used to get a well-defined focus on relevant architecture requirements in a concrete context and to determine how the architecture of a software product should look like to meet key requirements of the product [2][8]. The identification of most likely and probably risky scenarios is a key question in architecture evaluation. Typically, individual scenario brainstorming activities are an integral component of architecture evaluation processes, e.g., ATAM [17]. Nevertheless, individual scenarios are by nature subjective and reflect the experience and background of the reviewer. A team of heterogeneous reviewers will identify a wider range of scenarios. As face-to-face team meetings take considerable effort in particular for distributed teams, e.g., travelling and meeting effort, tool-supported team meetings have been suggested as alternative approach for conducting scenario brainstorming in a team. However, the question remains open, which type of meetings is most effective.

In this paper we report on a controlled experiment investigating the meeting styles of scenario brainstorming processes. The controlled experiment replicates and extends the design a previous empirical study (see [2] and [25]).

The remainder of this paper is structured as follows: Section 2 summarizes related work on software architecture evaluation approaches, in particular, scenario-based architecture reviews and on meeting styles in architecture evaluation. Section 3 motivates the research issues and derives research hypotheses. Section 4 describes the empirical study design and the experiment materials. Section 5 reports the study results, which are discussed in Section 6. Section 7 concludes and suggests further research work.

2. RELATED WORK

This section summarizes related work on software architecture evaluation, scenarios and team collaboration, which may have significant impact on the effort and effectiveness of scenario brainstorming processes.

2.1 Architectural Views and Evaluation Techniques

Various stakeholders use architecture artefacts – for different purposes and with different levels of experience. A programmer may want to see the relationship between individual modules, a project manager needs information for planning the project and the customer/user may want to know the software product structure for a buying decision. Kruchten [19] summarizes software architecture as a multi-dimensional artefact, which enables all stakeholders to discuss, communicate, and reason over the architecture. This multi-dimensional artefact can be described using the 4+1 view model of architecture [19] including a logical, implementation, process, and deployment views on software architecture, clearly assigned to the stakeholder roles user, programmer, system engineer, and system integrator. The use case view bridges all four individual views, typically described as scenarios. These different views on software architecture represent baseline of modern software architecture and build the foundation for high-quality software products.

Quality requirements can address possible upcoming needs, e.g., modifiability and maintainability, during development and maintenance. Potentially risky changes should be addressed early in the development life-cycle [5]. Note that changes can have a major impact, e.g., high rework effort and cost, on the software system if the underlying software architecture is affected by these changes. Thus, effective review processes are necessary to (a) identify defects in the architectural documents and (b) to evaluate various architecture variants with respect to quality requirements, e.g., modifiability and maintainability.

In general, architecture evaluation processes help analyzing a defined architecture or architecture variants to identify defects and improvement options [1]. Parnas and Weiss defined a general purpose approach to review designs – Active Design Reviews (ADRs) [21]. ADR focuses on different roles within a software project and the individual views the roles have on the architecture. Individual checklists and/or exercises support the reviews in conducting the design review ac-

tively. The Software Engineering Institute at CMU has included ADR for architecture into the software life-cycle process and came up with several methods for architecture design evaluation [14], e.g., Architecture Tradeoff Analysis Method (ATAM) [17], Software Architecture Analysis Model (SAAM) [18], Architecture-level Modifiability analysis (ALMA) [7], and Performance Assessment of Software Architectures (PASA) [24].

Architecture evaluation approaches support software architects in systematically evaluating architecture variants with respect to quality requirements, e.g., most likely upcoming changes of requirements in the future [23]. Scenarios are common approaches to capture most likely upcoming changes in the future.

2.2 Scenario Brainstorming

Scenarios and system properties are key elements of architecture evaluation processes to identify most likely future changes in the software architecture [2][8][13]. A scenario is a brief description of individual stakeholder interactions with a software system represented by workflows or system properties. For instance, a scenario can include an increased number of concurrently accessing users of a web service (system property) or modified workflows for data exchange. Thus, scenario can be used to define quality (also called non-functional) requirements [6]. Candidate scenarios are captured during software architecture by individual stakeholders in a scenario brainstorming session. Nevertheless, the quality of collected scenarios strongly depends on the experience of the individual stakeholders. Change categories [26] can support the individuals by providing guidance through the individual scenario brainstorming process. Change categories can include performance, security, and user interface. Individual reviewers can focus on a defined change category and collect candidate scenarios regarding this category more systematically. To increase the number and the quality of valuable scenarios, team-meetings are established approaches to derive a common team scenario list based on individual scenario brainstorming lists [3][26].

Therefore, scenario elicitation processes typically include individual scenario brainstorming sessions and a team meeting with all relevant stakeholders to achieve an aligned list of critical and important scenarios.

2.3 Team Meeting Collaboration

Team meetings are conducted in various software processes, e.g., requirements elicitation processes [12], inspection processes [10], to discuss and transform individual viewpoints or contributions into a team result.

In software inspection processes individual defect lists are transformed into one team defect list [15]. Fagan reported the benefit to find new defects in the team meeting [15]. However, there are also reports such as [20] that states that team meetings are not always more effective and efficient in comparison to a set of

individuals (e.g., nominal teams). In particular, the effectiveness of a team meeting for a specific purpose may depend on the meeting style, i.e., a face-to-face meeting, a nominal team aggregation, or a distributed tool-supported meeting.

A *face-to-face* team meeting is a real-world meeting, where the team members physically come together at a defined place, to discuss individual scenario lists and identify the most likely and valuable scenarios for a team scenario list. Similar to inspection meetings [10] a face-to-face meeting may bring up new scenarios because of discussion and interaction and may lose scenarios if the scenario is classified as less important. Previous studies, e.g., reported in [26] question the effectiveness of holding meeting as more scenarios were lost than gained in a face-to-face meeting. Nevertheless, face-to-face meetings typically require additional travelling activities and higher effort. Therefore it seems reasonable to conduct a distributed team meeting with appropriate tool support.

A *tool-supported meeting* has to include interaction via phone or chat and can be conducted in a distributed way. M.A. Babar *et al.* reported that a tool-supported meeting was significantly more effective than a face-to-face meeting, while the study participants preferred face-to-face meetings [3]. Grünbacher *et al.* reported on a study regarding groupware support for holding inspection meetings [16]. Main findings were that groupware-support reduce meeting effort and supported inspectors in identifying false positives (i.e., the number of wrong defects) and reduce the number of lost true defects [16].

Nominal teams [10], i.e., non-communicating teams, are virtual teams that do not meet. Instead team discussions a moderator automatically merges the individual scenario lists of team members into a common team scenario list. This is typically the cheapest form of aggregating individual lists into a team results. While some empirical studies [3] have investigated aspects of team meeting style, it remains open, whether unimportant scenarios are removed more effectively in face-to-face or tool-supported meetings and what the effects of individual experience on meeting performance are.

3. RESEARCH QUESTIONS

In this paper we investigate the effectiveness, i.e., the number of identified important scenarios (TopScenarios), of scenario brainstorming for face-to-face meetings and for tool-supported meetings to replicate and extend the results of previous studies [2][25].

The context of this study is a scenario brainstorming workshop including a two-stage approach, i.e., an individual scenario brainstorming and a team meeting in two sessions, i.e., covering two different software products. We applied a cross design for the team meeting styles: face-to-face meeting and tool-supported meet-

ing. Additionally we captured the individual experience of the participants to investigate the impact of individual and team experience on meeting style processes. Therefore, we can derive two main research questions:

- (a) *Does the meeting style (face-to-face and tool-supported meetings; nominal teams) have an impact on team effectiveness?* This research questions has initially been addressed in [3] and this study is therefore a replication in this part. As an extension of the study design in [3], we conducted a second architecture evaluation session to exclude the impact of the collaboration tool for team meeting discussions, as tool handling issues might hinder discussion processes.
- (b) *Does reviewer experience affect team effectiveness?* We assume that the individual experience can have an impact on the communication approach (meeting style). Therefore, we investigate team performance with respect to the experience of review teams based on individual experiences.

3.1 Variables

Following the standard practice of empirical software engineering we focus on performance measures, i.e., the number of identified scenarios and the number of important scenarios (TopScenarios"). See section 4.2 for details. The *independent variables* are the individual experience of the participants and the meeting styles used by the teams during the team phase, i.e., face-to-face and tool-supported meeting. *Dependent variable* is the number of identified scenarios per individual and team.

3.2 Research Hypotheses

From the research issues we derive the following hypotheses.

H1. Face-to-Face vs. Tool-Supported Meetings.

The first research issue investigates whether teams with face-to-face meetings find more important scenarios than teams with tool-supported communication.

More formally, the first null hypothesis is:

- H1.0: The number of identified important scenarios for face-to-face meetings is *higher* with respect to tool-supported meetings because real team meetings include discussion and interaction and communication via tools might hinder efficient interaction.

The alternative hypothesis is:

- H1.1: The number of identified important scenarios for face-to-face meetings is *lower* with respect to tool-supported meetings.

H2. Nominal Teams vs. Team Meetings. The second research hypothesis investigates whether face-to-face teams find more important scenarios than nominal (non-communicating) teams.

The second null hypothesis is:

- H2.0: The number of identified important scenarios of real team meetings is *higher* with respect to nominal teams because less important scenarios and false positives are excluded from the team scenario list. Nominal team lists will include also less important scenarios, reported only once by one individual.

The second alternate hypothesis is:

- H2.1: The number of identified scenarios of real team meetings is *lower* than nominal team results.

H3. Team Experience vs. Meeting Style.

The third research hypothesis focuses on the impact of team experience (based on individual experience questionnaires) on the meeting style, i.e., face-to-face and tool-supported meetings.

The third null hypothesis is:

- H3.0: The number of identified important scenarios is *higher* for face-to-face meeting with respect to tool-supported meetings for higher experienced reviewers because interaction and communication of higher experienced reviewers enables lively discussions and interaction. Tool supported meetings might hinder efficient and effective interaction.

The third alternate hypothesis is:

- H3.1: The number of identified scenarios of more experienced teams conducting face-to-face meetings is *lower* than the effectiveness of less experienced teams.

4. EMPIRICAL STUDY

This section summarizes study design, participants, applied material, and the experiment process.

4.1 Experiment Design and Study Process

The study was designed in a cross-over design regarding [3] the team meeting style in 2 sessions including two different applications. We use a distributed collaboration tool (Livenet [11]) in the first session and a Wiki system in the second session.

Figure 1 illustrates the basic study setting and arrangements including a pre-defined time-interval for executing the individual tasks. Prior to the study we gave a 2h tutorial on scenario-based architecture evaluation and a basic introduction to the study. At the beginning of the study, we used an experience questionnaire to capture the individual background information of the participants to assess individual and team experience. After the first individual brainstorming session (session 1) we captured feedback from the participants on the brainstorming process (feedback questionnaire). As the brainstorming activity does not vary in the second session, we did not capture feedback on the brainstorming process after the second session.

Because the meeting style changed in the second session, we captured feedback information on the meeting style after every meeting activity. Finally, we collected feedback on the overall study and the individual experiences with different meeting styles at the end of session 2.

Session 1 - Livenet	Experience Questionnaire		45 min
	Individual Phase		
	Individual Scenario Brainstorming		
	Feedback Individual Questionnaire		
	Team Phase		60 min
	Face-to-Face (F2F)	Tool-Support (TS)	
	Feedback Team-Meeting Questionnaire		
Break			30 min
Session 2 - Wiki	Individual Phase		45 min
	Individual Scenario Brainstorming		
	Team Phase		
	Tool-Support (TS)	Face-to-Face (F2F)	60 min
	Feedback Team-Meeting Questionnaire		
	Final Questionnaire		15 min

Figure 1: Experiment Design.

To support brainstorming activities, we introduced change categories for individual scenario brainstorming to support reviewers in the brainstorming process [25]. Note that we do not investigate the impact on change categories in this paper. Nevertheless, we applied parts of the study setting from previous studies, reported in [2][25]. After individual scenario brainstorming teams of 3 people were assigned randomly to hold a team meeting regarding the assigned meeting style, i.e., face-to-face meeting and tool-supported meeting. Note that we defined an upper time limit (illustrated in Figure 1) of 45 min for individual scenario brainstorming and 60 min for team meeting activities.

This design enables an investigation of meeting style performance in two sessions and the impact of individual experience on brainstorming effectiveness, i.e., the number of identified scenarios. For data analysis we applied the t-test at a significance level of 95% for statistical evaluation.

4.2 Experiment Material

The study material was based on a previous study, which aimed at initially investigating the effects of brainstorming effectiveness with active guidance, i.e., using change categories [2], the impact of team size [25] and effects of nominal teams [26] on scenario brainstorming processes. Note that we used similar material and included a second session as well as an improved study process and improved materials.

Application Domain

In this study we applied two different software products, i.e., two distributed collaboration tools, in two sessions: (a) a collaborative workflow tool (Livenet) and (b) a well-known Wiki system.

Livenet [11] is a web-based collaborative tool that supports synchronous (same time, different places) and asynchronous (different time, different places) activities. In this application a user is able to communicate with others via discussion-forums and real-time chat. Every user is assigned to a different role, which means that every user has different permission within the system. A team leader might schedule tasks for his team members, a “normal” user may create a work-space where he can give some users access to. In such workspaces documents can be stored. Note that *Livenet* aims at supporting software engineering workflows. The *Wiki system*, e.g., *wikipedia*, is a web-based collaborative content management system to organize individual content information according to specific topics. The Wiki-system can be accessed through a web browser. Users can create new articles and edit existing contributions. All articles in the Wiki can link to each other using defined naming convention. The applied Wiki system uses text and simple tagging information to keep the system as simple as possible and textual chat features. Note that Wiki does not provide any workflow support.

Questionnaires and Data Collection

We used four questionnaires to capture (a) *individual background* information for qualification assessment, (b) *feedback from individual scenario brainstorming*, (c) *feedback questionnaires regarding different meeting styles* (face-to-face and tool-supported meeting), and (d) an *overall feedback* questionnaire. To capture the identified scenarios we used a *data capturing* sheet to collect all candidate scenarios identified by the participants during the brainstorming process. Additionally we used a similar sheet to capture team scenarios during the meeting (paper-based approach).

Supporting Material (Guidelines)

To guide the participants through the study process we used an *application guideline* illustrating the detailed schedule of the study (including details of the individual tasks during the study) and a basic introduction to the most important features of the application. Additionally, we provided a detailed guideline for the individual brainstorming task and the team meeting regarding the sequence of steps to successfully identify/discuss most valuable scenarios.

Reference Scenario Profiles and TopScenarios

After finishing the study we analysed the scenarios according to their importance to achieve a ranking of scenarios according their value contribution. Assuming a high number of similar scenarios identified per individual and teams indicates a highly valuable scenario. Based on this assumption, also supported by [8] and [2], we aggregated the identified scenarios to a top-list in descending order according to their appearance in individual and team brainstorming lists and defined a reference scenario profile. Note that we used a weight for individual and team scenarios (score). If an individual found a certain scenario, it got 1 point. If a certain

scenario was found by a team, it got 2 points. Note that scenarios of team lists were discussed and classified as valuable scenario by all team members. Thus, the importance of the scenario was confirmed. To focus on most valuable scenarios we used the top 20% scenarios based on the score for *Livenet* and *Wiki* for further investigations (TopScenarios). See section 5.1 for a sample evaluation of TopScenarios for *Livenet*.

4.3 Participants

The participants were 54 master students who attended quality related courses, e.g., advanced aspects of quality management, software testing, and empirical software engineering. We assigned the participants randomly using a sort card algorithm to different groups and 2-3 person teams. To achieve comparable team distribution for both sessions, we used 16 3-person teams and 3 2-person teams. Thus, the team assignment was balanced. Note that two participants left the study after the first session (both were assigned to the same team). Table 1 illustrates the distribution of participants to meeting styles and applications. Note that every participant starts with *Livenet* in the first session and continued with *Wiki* in the second session.

Table 1: Number of participants in *Livenet* (session 1) and *Wiki* system (session 2).

	Livenet	Wiki	Total
Tool-Support	30	22	52
Face-to-Face	24	30	54
Total	54	52	

4.4 Threats to Validity

In order to increase internal and external validity we considered a set of threats and implemented appropriate countermeasures to address them:

Internal threats to validity:

- We avoided *communication* during the individual brainstorming task during the study. The study was conducted in classroom setting supported by supervisors, i.e., the study team.
- The overall *duration* of the study was limited to 45 minutes for individual brainstorming and to 60 min for the team meeting process. Because of the duration of the individual tasks, no breaks were possible during the study.
- *Background knowledge and experience* was captured prior to the study by using a background questionnaire. Thus we are able to assess individual qualification of the participants and can reason to the team experience.
- *Feedback questionnaire*. Additionally, we captured feedback after the individual brainstorming process and each team process to get information whether the participants followed the study process properly. Thus, we were able to exclude participants from the analysis if they did not follow the study

process as indicated. Note that two participants (one team) left the study after the first session.

- *Learning.* The study was conducted in two sequential sessions. Thus, the participants were probable able to remember aspects of the previous task and might be able to identify additional scenarios. Nevertheless we use a different software product, which was not connected to the first software product (Livenet vs. Wiki).

External threats to validity:

- *Application domain.* We used well-known application domains for both software products (Livenet and Wiki). As Livenet is not as popular we gave the participants the opportunity to test the system in the month prior to the study. In addition, we provided a small tutorial to get familiar with the system. We considered Wiki to be well-known to all participants.
- *Selection of participants.* The participants were master students and attended advanced courses in quality management, software testing and empirical software engineering. Furthermore, most of the participants work in an industry setting as professionals. Thus, they can be considered as semi-professionals. Nevertheless, none of them had prior experiences on architecture evaluation, which might also be the case in industry applications.

5. EXPERIMENT RESULTS

This section summarizes the data analysis process and the results of the controlled experiment including descriptive statistics and statistical tests.

5.1 Data Capturing and Data Analysis Procedure

We captured the scenarios using a data capturing sheet for individual scenario brainstorming and different meeting styles. We applied a check for consistency regarding multiple entries and similar scenarios in the individual and team scenario lists. Afterwards we generated reference scenario profiles (see section 4.2) to get the “TopScenario” list for further investigation. Nominal team lists were calculated by merging individual scenario lists excluding multiple similar entries of real teams (multiple entries were counted once). Note that we used real team members for the calculation of nominal teams. To achieve comparability, we calculated nominal team lists for face-to-face teams and tool-supported teams separately.

For analysis purposes we apply boxplots and descriptive statistics. Additionally, we applied the t-test at significance level of 95% for statistical testing, i.e., comparing groups who applied different meeting styles, i.e., face-to-face, tool supported meeting, and nominal teams.

The main goal of this study is to investigate the impact of the meeting style and team experience with respect

to team meeting performance, i.e. the number of scenarios and TopScenarios. Figure 2 illustrates the distribution of the identified scenarios and the share of TopScenarios (top 20% of all identified scenarios) for the Livenet application.

Figure 2: Identified and TopScenarios for Livenet.

Note that we applied the 20/80 distribution to classify the scenarios according to their individual likelihood of occurrence in the future. This assumption is based on value-based requirements considerations [9] already applied in previous publications [25]. Table 2 shows the number of identified scenarios and derived TopScenarios for Livenet and Wiki.

Table 2: Number of Identified Scenarios.

	All scenarios	TopScenarios	Share
Livenet	174	48	18%
Wiki	193	49	20%

5.2 Face-to-Face vs. Tool-Supported Meeting

The first hypothesis focuses on the impact scenario brainstorming in face-to-face meetings and tool-supported meetings. Following previous publications [3], we expect that the number of scenarios is significantly higher for tool-supported meetings. Thus, we investigated the number of scenarios for both applications, i.e., Livenet (in the first session) and Wiki (in the second session). Accordingly, we expected a similar behaviour in this study regarding face-to-face and tool-supported team meetings.

Table 3: Scenarios found per Meeting Style (real teams).

	Face-to-Face		Tool-Support		p-value
	Mean	SD	Mean	SD	
Livenet	10.6	2.19	8.5	2.88	0.113(-)
Wiki	11.3	3.59	9.4	3.16	0.274(-)

Table 3 shows the result of all identified scenarios in two team meeting styles. The results showed benefits for the face-to-face meeting style for both application (Livenet and Wiki). Applying the t-test at a significance level of 95%, we did not observe any significant differ-

ences. Focussing on most likely scenarios we use TopScenarios, i.e., the top 20% of scenarios for further analysis.

Figure 3 shows the number of scenarios found for real team meetings, i.e., face-to-face (F2F) and tool-supported (TS) meetings regarding the Livenet and the Wiki application.

Figure 3: TopScenarios found by Real Teams (Livenet and Wiki).

F2F-teams found on average 6.9 TopScenarios (std.dev 1.96) and tool-supported teams found on average 5.0 (std.dev. 1.63) TopScenarios in the Livenet application. Applying the t-test at a significance level of 95%, we do not observe significant differences ($p\text{-value} = 0.035(-)$). Because these results differ from previous studies [3] we can assume that the participants were not familiar with the collaboration tool. Expecting similar results in the second session, we analyzed the team effects for the Wiki-system. Note that this application was investigated after the Livenet in a second session. Figure 3 also illustrates the findings for Wiki. We observed advantages for the face-to-face meeting style for the Wiki-system but we did not identify any significant differences ($p\text{-value} = 0.203(-)$). Note, that the number of identified scenarios increases for face-to-face meetings and tool-supported meetings in the second session. See table 4 summarizes the descriptive statistics for Livenet and Wiki.

Table 4: TopScenarios found by Meeting Style.

	Face-to-Face		Tool-Support		p-value
	Mean	StdDev	Mean	StdDev	
Livenet	6.9	1.96	5.0	1.63	0.035(-)
Wiki	7.5	2.32	6.3	1.67	0.203(-)

Nevertheless the results support the assumption that F2F meetings identify more scenarios with respect to face-to-face meetings. Nevertheless, we did not observe significant differences. A possible reason might be that the participants got more familiar with architecture evaluation and scenario brainstorming process in

the second session. The results for both sessions show benefits for face-to-face meetings with respect to the tool-supported meeting style. Because we did not observe significant differences, H1.0 and the alternative hypothesis H1.1 must be rejected.

5.3 Nominal Teams vs. Team Meetings

Previous studies question the effectiveness of holding team meetings because more scenarios were lost than gained [26]. Thus, the application of nominal teams, i.e., non-communicating teams without holding a team meeting, is a promising approach for generating team scenario lists. The generation of team scenario lists is based on individual scenario lists, delivered by individual reviewers without holding a team meeting. Note that every scenario, captured by at least one individual, is included in the team scenario list; we do not count multiple similar scenarios in the team list. No scenarios are lost but there are no new additional scenarios, derived by discussion and interaction of team members. Table 5 and Figure 4 present the results of nominal teams and real teams with respect to all scenarios and TopScenarios for the Livenet application. Note that we calculate the nominal teams for every meeting style group to show the effect with respect to the real team. For nominal team result calculation we applied the individual scenarios of individual real team members of the corresponding real team.

Table 5: Nominal teams vs. Real Teams (Livenet)

	Nominal Teams			Real Teams		
	ALL	TOP	%	ALL	TOP	%
F2F	14.2	7.1	50%	10.6	6.9	65%
TS	16.5	8.0	48%	8.5	5.0	59%

Note: F2F ... face-to-face team meeting, TS ... tool-supported team meeting.

Figure 4: TopScenarios found by Nominal and Real Teams in the Livenet application.

It is no surprise that the nominal team results include on average more scenarios and more TopScenarios

with respect to real team meetings. It is notable, that TS-team members identified more scenarios and more TopScenarios in the individual scenario brainstorming task. Summarizing nominal team results, 50% of scenarios identified by F2F members and 48% of scenarios captured by TS members are TopScenarios. Note that individual scenario brainstorming activity is similar. Regarding real team meetings, real teams identified a higher share of TopScenarios (65% F2F for teams and 59% for TS teams) with respect to the corresponding nominal team list. These findings indicate that fewer scenarios were excluded during team meetings based on discussion and interaction during a F2F meeting. Note that there are benefits for F2F team meetings.

Because the reviewers found on average more scenarios and more TopScenarios in the second session (Wiki) we expect similar behaviour regarding these team effects. Additionally, the participants are more experienced in scenario brainstorming in the second session. Note that the individual brainstorming activity is similar to the first session, the meeting style is different. Table 6 and Figure 5 present the results of average scenarios for all scenarios and TopScenarios for the Wiki system regarding nominal teams and real team meetings.

Table 6: Nominal teams vs. Real Teams (Wiki)

	Nominal Teams			Real Teams		
	ALL	TOP	%	ALL	TOP	%
F2F	18.7	9.9	52%	11.3	7.5	66%
TS	19.25	11.6	60%	9.4	6.3	67%

Note: F2F ... face-to-face team meeting, TS ... tool-supported team meeting.

Regarding nominal teams the share of TopScenarios increases by 2% (F2F) and 12% (TS). Additionally, the share of TopScenarios with respect to real teams increases by 1% (F2F meeting) and 8% for TS team meetings. It is notable that the share of TopScenarios increases in the second session for F2F team members and TS team members.

Figure 5: TopScenarios found by Nominal and Real Teams in the Wiki application.

One reason can be a learning effect on scenario brainstorming because the number of identified scenarios increases as well. Additionally, we observed a similar trend compared to the Livenet application. Concerning nominal teams, the TS members found significantly more defects individually. Note that these results are not related to the meeting style as this is an individual task. Regarding real team meetings, we observed advantages for face-to-face meetings regarding TopScenario identification. Nevertheless, we did not observe any significant differences. These results indicate that discussion and interaction in real team meeting support scenario capturing processes better.

Real team meetings found fewer scenarios and less important scenarios than nominal teams. Note that we observed benefits for TS regarding nominal teams and benefits for F2F meetings regarding real team meetings for the livenet and wiki application.

Based on these results, H2.0 (number of identified TopScenarios is higher for face-to-face teams) must be rejected for Livenet and Wiki. H2.1 is supported by the results. Additionally, we did not observe significant differences between face-to-face and tool-supported meetings.

5.4 Impact of Experience on Meeting Style

Based on the experience captured prior to the study, we are able to identify the individual experience of the participants and a team experience value for team qualification. The individual experience questionnaire included 18 questions covering general software engineering skills, project and architecture experience, and quality assurance skills. The questionnaire includes topics like:

- Estimate your experience in quality assurance in a professional environment.
- Estimate the experience with architecture reviews.

We measured the experience-value of every question on a scale from 0 (no experience) to 4 (professional application and highly experienced). The team experience value was calculated by a mean value of individual answers per real teams. The results showed a mean value between 1.47 and 1.55 experience “points”. To investigate the impact of team experience and the relationship between the number of found TopScenarios and the experience of the teams, we conducted a t-test at a significance level of 95%. Balancing the team experience, we split the teams into two experience groups – more experienced and less experienced teams – and set the threshold to 1.6 experience points. Based on this threshold we identified 9 high experienced teams and 10 less experienced teams.

The results in Table 7 show that there is no clear relationship of team experience and the number of identified scenarios.

Table 7: TopScenarios found by Application and Team Experience (Livenet).

	High-Experience		Less Experience		p-value
	Mean	StdDev	Mean	StdDev	
Livenet	6.1	1.7	5.7	2.3	0.662(-)
Wiki	7.5	2.7	6.5	1.5	0.330(-)

The results showed that there is no significant difference regarding the experience and the number identified TopScenarios in both applications. In context of this analysis there seems to be no relationship between team experience and the number of found TopScenarios. Thus, H3.0 and H3.1 must be rejected.

6. DISCUSSION

In this section we discuss the results for each research issue with finding from other empirical studies.

Face-to-face vs. Tool-supported Meetings. This research issue investigated whether teams that communicate face-to-face (F2F) find more important scenarios than teams communicating with some kind of tool support (TS). Following previous publications [3] we expected a similar behavior regarding the identification of TopScenarios for tool-supported meetings. The results showed benefits regarding the identification of TopScenarios for the face-to-face meeting style. This finding is contradictory to previous studies, where significant benefits for tool-supported team meetings were reported [3]. One possible reason for this deviation might be that the applied collaborative tool for tool-supported communication hindered efficient collaboration and interaction. Nevertheless, both meeting styles identified more scenarios in a second session. The most likely reason is that the participants were more familiar with scenario brainstorming in the second session as the overall number of identified candidate scenarios is higher in the second session. Because we did not observe significant differences, we have to reject H1.0 and H1.1.

Nominal Teams vs. Team Meetings. The second research question focused on whether teams find more important scenarios than nominal teams, i.e., non-communicating teams without conducting a team meeting. Assuming that more important scenarios should be identified during interaction and discussion, we expected advantages for real teams (face-to-face and tool-supported meetings) with respect to nominal teams. The results showed that individual participants always found more scenarios in general than in the team phase. They also found more TopScenarios individually than real teams. A possible reason is that the 60 min duration for a team meeting might be too short. In a longer team phase the teams may have found less scenarios in general, because of the elimination of unimportant scenarios, but the participants found

nearly the same number of TopScenarios in the team phase as they found individually. From data analysis we can see that F2F teams always had a higher share of TopScenarios in their profiles than TS teams. F2F teams also had a higher increase of TopScenarios in their profiles compared to the individual phase, so we can reject H2.0. H2.1 is supported by the results.

Impact of Team Experience on the number of identified TopScenarios. The third research issue investigates whether teams with more experience have better results, i.e., they find more important scenarios, than teams with less experience. Because of a comparable qualification level of the participants, the threshold for higher and lower qualified participants seems to be less selective (we did not observe any significant differences regarding team qualification). We compared the experience rating points of the teams with the number of TopScenarios found. Main result was that teams with higher experience did not find significantly more TopScenarios than less experienced teams. H3.0 and H3.1 must be rejected.

Lessons learned. From conducting this empirical study we found several issues that should be considered for future studies in this line of research.

- *Experiment duration.* What we learned that the team phase for constructing a team scenario list should be extended at at least 50%. This estimation was confirmed by the feedback questionnaire.
- *Relevant experience for architecture evaluation.* We also learned that experience in software engineering topics had very little impact on experiment results. Thus, a replication of this study including a higher distribution of various qualification levels seems to be necessary.

7. CONCLUSION AND FURTHER WORK

Scenario-based software architecture evaluation systematically investigates architecture variants with respect to their quality attributes, e.g., modifiability and maintainability of a software system. Scenarios and system properties are key elements of architecture evaluation processes. Nevertheless, scenarios must be captured in an effective and efficient way. Team scenario brainstorming activities can help to focus on important scenarios. Nevertheless, different meeting styles, i.e., face-to-face meetings and tool-supported meetings, can be alternatives to nominal team approaches.

In this paper we reported on a controlled experiment investigating the meeting style of scenario brainstorming processes to find out which meeting style, face-to-face or tool-supported team meetings with respect to the number of derived scenarios. Major result was that face-to-face meetings outperformed in the experiment tool-supported meetings in finding more important scenarios.

Based on the experiment data we found that F2F teams found more and better scenarios than TS teams.

Further work is (a) to elaborate on the study results to get a deeper insight in architecture evaluation processes with respect to team collaboration and (b) capture and a more detailed view on experience of participants, in particular software architects. Additionally we plan a replication of the study including a longer study execution phase with respect to the team meeting process.

ACKNOWLEDGEMENT

We want to thank M. Ali Babar for his valuable feedback on the study design and the experiment package.

REFERENCES

- [1] Abowd G., Bass L., Clements P., Kazman R., Northrop L., Zaremski A.: *Recommended Best Industrial Practice for Software Architecture Evaluation*, Software Engineering Institute Technical Report CMU/SEI-96-TR-025, Software Engineering Institute, Carnegie Mellon University, 1997.
- [2] Babar M.A., Biffl S.: *Eliciting Better Quality Architecture Evaluation Scenarios: A Controlled Experiment On Top-Down vs. Bottom-Up*, Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering, 2006.
- [3] Babar M.A., Kitchenham B.A. Jeffery D.R.: *Comparing distributed and face-to-face meetings for software architecture evaluation: A Controlled Experiment*, EMSE Journal 13(1), 2008.
- [4] Bahsoon R.: *Evaluating Software Architectures for Stability: A Real Options Approach*, Proc. of the 25th International Conference on Software Engineering, Portland, USA, 2003.
- [5] Barbacci M., Klein M., Longstaff T., Weinstock C.: *Quality Attributes*, CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University, 1995.
- [6] Bass L., Clements P., Kazman R.: *Software Architecture in Practice*, Addison Wesley, 1998.
- [7] Bengtsson P., Lassing N., Bosch J., Van Vliet H.: *Architecture-level modifiability analysis (ALMA)*, The Journal of Systems and Software 69, 2004.
- [8] Bengtsson P., Bosch J.: *An Experiment on Creating Scenario Profiles for Software Change*, Annals of Software Engineering, 9, 2000.
- [9] Biffl S., Aurum A., Boehm B., Erdogmus H., Grünbacher P.: *Value-Based Software Engineering*, Springer, 2005.
- [10] Biffl S., Halling M.: *Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams*, IEEE Transactions on Software Engineering Vol 29(5), 2003.
- [11] Biuk-Aghai R.P., Hawryszkiewycz I.T.: *Analysis of Virtual Workspaces*. Proc. of the Database Applications in Non-Traditional Environments, 1999.
- [12] Boehm B., Grünbacher P., Briggs R.: *EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation*, Int. Conf. on SE, 2001.
- [13] Bosch J., Bengtsson P.: *An experiment on creating scenario profiles for software change*, Annals of Software Engineering, Vol. 9, Issue 1-4, 2000.
- [14] Clements P., Kazman R., Klein M.: *Evaluating Software Architectures: Methods and Case Studies*, Addison Wesley, 2002.
- [15] Fagan M. E.: *Design and Code Inspections to Reduce Errors in Program Development*. IBM Syst. J., Vol. 15, no. 3, pp. 182-211, 1976.
- [16] Grünbacher P., Halling M., Biffl S.: *An Empirical Study on Groupware Support for Software Inspection Meetings*, Proc. of Automated Software Engineering, 2003.
- [17] Kazman R., Klein M., Barbacci M., Lipson H., Longstaff T., Carrière S.J.: *The Architecture Trade-off Analysis Method*, Proceedings of the 4th ICECCS, pp. 68-78, 1998.
- [18] Kazman R., Abowd G., Bass L., Webb M.: *SAAM: A Method for Analyzing the Properties of Software Architectures*, Proceedings of the 16th International Conference on Software Engineering, pp. 81-90, 1994.
- [19] Kruchten P.: *The Rational Unified Process – An Introduction – 3rd Edition*, Addison Wesley, 2003.
- [20] Lawrence G., Votta J.: *Does Every Inspection Need A Meeting?* Proceedings of the 1st ACM SIGSOFT symposium on Foundations of software engineering, pp. 107-114, 1993.
- [21] Parnas D.L., Weiss D.: *Active Design Reviews: Principles and Practices*, Proceedings of the 8th International Conference on Software Engineering, pp. 132-136, 1985.
- [22] Taylor R.N., Medvidovic N., Dashofy E.: *Software Architecture: Foundations, Theory, and Practice*, Wiley, 2009.
- [23] Van Vliet H., De Bruin H.: *Scenario-Based Generation and Evaluation of Software Architectures*, Berlin, Springer, pp. 128-139, 2008.
- [24] Williams L. G., Smith C. U. PASA: *A Method for the Performance Assessment of Software Architectures*, Proceedings of the Third International Workshop on Software and Performance, 2002.
- [25] Winkler D., Biffl S., Babar M.A.: *Impact of experience and team size on the quality of scenarios for architecture evaluation*, Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering (EASE), 2008.
- [26] Winkler D., Biffl S., Ali Babar M.: *An Empirical Investigation of Scenarios Gained and Lost in Architecture Evaluation Meetings*, International Symposium on Empirical Software Engineering and Measurement (ESEM), Short Paper, Kaiserslautern, Germany, 2008.