Research Report 2/99

An Experiment on Creating Scenario Profiles for Software Change

by

PerOlof Bengtsson, Jan Bosch

Department of
Software Engineering and Computer Science
University of Karlskrona/Ronneby
S-372 25 Ronneby
Sweden

ISSN 1103-1581
ISRN HK/R-RES—99/2—SE

An Experiment on Creating Scenario Profiles for Software Change

by PerOlof Bengtsson, Jan Bosch

ISSN 1103-1581
ISRN HK/R-RES—99/2—SE

Printed by Psilander Grafiska, Karlskrona 1999

An Experiment on Creating Scenario Profiles for Software Change

PerOlof Bengtsson & Jan Bosch
Department of Software Engineering and Computer Science
University of Karlskorna/Ronneby
S-372 30 Ronneby, +46 457 666 91
[PerOlof.Bengtsson|Jan.Bosch]@ipd.hk-r.se
URL: http://www.ipd.hk-r.se/[~pob|~bosch]

Abstract

Scenario profiles are used increasingly often for the assessment of quality attributes during the architectural design of software systems. However, the definition of scenario profiles is subjective and no data is available on the effects of individuals on scenario profiles. In this paper we present the design, analysis and results of a controlled experiment on the effect of individuals on scenario profiles, so that others can replicate the experiments on other projects and people. Both scenario profiles created by individuals and by groups are studied. The findings from the experiment showed that groups with prepared members proved to be the best method for creating scenario profiles. Unprepared groups did not perform better than individuals when creating scenario profiles.

Keywords: Experiment studies, empirical studies, scenarios, scenario profile, software architecture, software quality factors, prediction

1 Introduction

During recent years, the importance of explicit design of the architecture of software systems is recognized [2,5,8,13]. The software architecture constrains the quality attributes and the architecture should support the quality attributes significant for the system. This is important since changing the architecture of a system after it has been developed is generally prohibitively expensive, potentially resulting in a system that provides the correct functionality, but has unacceptable performance or is very hard to maintain.

Architecture assessment is important to decide the level at which the software architecture supports various quality attributes. The need for evaluation and assessment methods have been indicated by [1,2, 9,10, 11]. Architecture assessment is not just important to the software architect, but is relevant for all stakeholders, including the users, the customer, project management, external certification institutes, etc.

One can identify three categories of architecture assessment techniques, i.e. scenario-based, simulation and static model-based assessment. However, these techniques all make use of scenario profiles, i.e. a set of scenarios. For assessing maintainability, for example, a maintenance profile is used, containing a set of change scenarios.

Although some scenarios profiles can be defined as 'complete', i.e. covering all scenarios that can possibly occur, most scenario profiles are 'selected'. *Selected scenario profiles* contain a representative subset of the population of all possible scenarios. To use the aforementioned maintenance profile as an example, it is, for most systems, impossible to define all possible change scenarios, which requires one to define a selection that should represent the complete population of change scenarios.

Scenario profiles are generally defined by the software architect as part of architecture assessment. However, defining a selected scenario profile is subjective and we have no means to completely verfiy the representativeness of the profile. Also, to the best of our knowledge, no studies have been reported about the effects of individuals on the creation of scenario profiles, i.e. what is the deviation between profiles created by different individuals. The same is the case for groups defining scenario profiles.

Above, the general justification of the work reported in this paper is presented. A second reason for conducting this study is that in [5] we proposed a method for (re)engineering software architectures and architecture assessment is a key activity. As part of that method, we have developed a technique for scenario-based assessment of maintainability [6]. An important part of the technique is the definition of a maintenance scenario profile. Since the accuracy of the assessment technique is largely dependent on the representativeness of the scenario profile, we conducted an experiment to determine what the effect of individuals and groups is on the definition of scenario profiles. Therefore, we will primarily use maintenance scenario profiles as examples, even through the results are valid for other quality attributes as well.

The intention of the experiment is threefold:

1. testing three different methods of synthesizing scenario profiles.
2. test the hypothesis that there is a difference between the methods.
3. find out which one of the three methods that are the best.

To conduct the experiment, we used volunteering students from the Software Engineering study program at the University of Karlskrona/Ronneby, all currently on their Masters year.

The remainder of the paper is organized as follows. In the next section we describe the concept of scenario profiles in more detail. The design of the experiment is presented in section 3, followed by the analysis and interpretation of the results in section 4. Related work is discussed in section 5 and the paper is concluded in section 6.

2 Scenario Profiles

Scenario profiles describe the semantics of software quality factors, e.g. maintainability or safety, for a particular system. The description is done in the terms of a set of scenarios. Scenarios may be assigned an associated weight or probability of occurrence within in a certain time, but we do not address that in this paper. To describe, for example, the maintainability requirement for a system, we list a number of scenarios that each describe a possible and, preferably, likely change to the system. The set of scenarios is called a scenario profile. An example of a software change scenario profile for the software of a haemo dialysis machine is presented in figure 1.

Category	Scenario Description
Market Driven	**S1** Change measurement units from Celsius to Fahrenheit for temperature in a treatment.
Hardware	**S2** Add second concentrate pump and conductivity sensor.
Safety	**S3** Add alarm for reversed flow through membrane.
Hardware	**S4** Replace duty-cycle controlled heater with digitally interfaced heater using percent of full effect.
Medical Advances	**S5** Modify treatment from linear weight loss curve over time to inverse logarithmic.
Medical Advances	**S6** Change alarm from fixed flow limits to follow treatment.
Medical Advances	**S7** Add sensor and alarm for patient blood pressure
Hardware	**S8** Replace blood pumps using revolutions per minute with pumps using actual flow rate (ml/s).
Com. and I/O	**S9** Add function for uploading treatment data to patient's digital journal.
Algorithm Change	**S10** Change controlling algorithm for concentration of dialysis fluid from PI to PID.

Figure 1: Maintenance Scenario Profile Example

Scenario profiles represent a way to document and leverage from the experts knowledge about the system. It also provides its users with a way to determine where they lack knowledge about the system.

2.1 Scenario Profile Usage

A scenario profile can, basically, be defined in one of two contexts, i.e. the 'greenfield' and the experienced context. If a scenario profile is defined in an organization using the technique for the first time, for a new system and no historical data is available about similar systems, we are fully dependent on the experience, skill and creativeness of the individuals defining the profile. The resulting scenario profile is the only input to the architecture assessment. The lack of alternative data sources in this case and the lack of knowledge about the representativeness of scenario profiles defined by individuals and groups, indicates that there is a need to increase our understanding of profiles in this situation.

In the second situation, there is either an earlier release of the system or historical data of similar systems available. Since, in this case, empirical data can be collected, we can use this data as an additional input for the next prediction and thus get an more accurate result. However, even when historical data is available to be used as a reference point, it is important that the persons synthesizing the profile also incorporate the difference from similar systems or with the previous release of the system. The problem might otherwise be that, using only the historical data, one predicts the history. Consequently, important future scenarios, that the domain experts are aware of, may be over looked. The latter, however, remains to be empirically validated and will not directly be addressed in this experiment.

In the experiment reported in this paper, we address the first situation, i.e. defining a scenario profile without historical data, since few prediction methods are available for this situation. Once the source code of a (similar) system is available, traditional assessment methods exists, e.g. Li & Henry [14].

2.2 Methods of Profile Creation

Scenario profiles can be created in at least three different ways. First, an individual could be assigned the task of independently creating a scenario profile for a software quality attribute of a system. Second, a group of people could be assigned the same task. Third, a group of people could be assigned the same task, but are required to prepare themselves individually before meeting with the group.

In the case of an individual creating a scenario profile, the advantage is, obviously, the relatively low resource cost for creating the profile. However, the disadvantage is that there is a, hard to assess, risk that the scenario profile is less representative due to the individual's lack of experience in the domain, or misconceptions about the system.

The second alternative, i.e. a group that jointly prepares a scenario profile, has as an associated disadvantage that the cost of preparing the profile is multiplied by the number of members of the group, meaning maybe three to five times more expensive. However, the risk of the forecast being influenced by individual differences is reduced since the group has to agree on a profile. Nevertheless, a risk with this method is that the resulting scenario profile is influenced by the most dominant rather than the most knowledgeable person, and thus affecting the scenario profile negatively. Finally, the productivity might be very low when in group session, since obtaining group consensus is a potentially tedious process.

The third alternative, in which the group members prepare an individual profile prior to the group meeting, has as an advantage that the individual productivity and creativity is incorporated when preparing the profiles, and then the unwanted variation of individuals are reduced by having the group agreeing on a merged scenario profile. A disadvantage is the increased cost, at least when compared to the individual case, but possibly also when compared to the unprepared group alternative.

The experiment reported in this paper studies the difference in the produced results from these three methods and compares the methods.

3 The Experiment

3.1 Goal and purpose

The purpose of this experiment is to gain understanding of the characteristics of scenario profiles and the influence and sensitivity of individuals participating in the specification of the scenario profiles. The questions we are asking and would like to answer are:

- How much do profiles created by independent persons vary for a particular system?
- How does a profile, created by an independent persons, differentiate from a profile created by a group?
- What are the difference in the results from scenario profile created by a group, if the individual members have prepared their own profiles first, compared to profiles created groups with unprepared members.
- How does these variances impact the predicted values? Are they absolutely critical to the method?

In the next section these questions have been formulated as more specific hypotheses and corresponding null-hypotheses.

3.2 Hypotheses

We state the following null-hypotheses:

> **H_{01}** = *No significant difference in score between scenario profiles created by individual persons, or groups with unprepared members.*
>
> **H_{02}** = *No significant difference in score between scenario profiles created by individual persons, or groups with prepared members.*
>
> **H_{03}** = *No significant difference in score between scenario profiles created by groups with unprepared members, or groups with prepared members.*

In addition we state our six main hypotheses that allow us to rank the methods, even partially if the experiment does not produce significant results to support all stated hypotheses:

> **H_1** = *Scenario profiles created by groups with unprepared members, generally get better scores than scenario profiles created by an individual person.*

And the counter hypothesis to **H_1**, denoted **H_{10}** to more clearly show its relation to **H_1**.

$\mathbf{H_{10}}$ = *Scenario profiles created by individuals generally get better score than profiles created by groups with unprepared members.*

$\mathbf{H_2}$ = *Scenario profiles created by groups with prepared members, generally get better scores than scenario profiles created by an individual person.*

$\mathbf{H_{20}}$ = *Scenario profiles created by individuals generally get better score than profiles created by groups with prepared members.*

$\mathbf{H_3}$ = *Scenario profiles created by groups with prepared members generally get better scores than group profiles with unprepared members.*

$\mathbf{H_{30}}$ = *Scenario profiles created by groups with unprepared individuals generally get better score than profiles created by groups with prepared members.*

These hypothesis will allow us to make some conclusions about the ranking between the methods, even though the data does not allow us to dismiss all null-hypotheses or support all the main hypotheses.

3.3 Experiment Design

To test these hypotheses using an experiment, we decided to employ a blocked project design with two project requirement specifications and twelve persons divided into four groups with three persons in each group.

From the hypotheses we get three types of methods for creating change scenario profiles, i.e. treatments. Since scenario profiles need to be concrete, we decided to use the definition of change scenario profiles. However, the design of the experiment is such the results are applicable to selected scenario profiles for other quality attributes as well. The three ‘treatments’ that we use in the experiment are the following:

1. One independent person create a change scenario profile.
2. A group, with unprepared members, create a change scenario profiles.
3. A group, with members prepared by creating personal profiles before meeting in the group, creates the change scenario profile.

One of the problems in executing software development experiments is the number of persons required to test different treatments in robust experiment designs. In our previous experimentation experience, our main problem has been to find sufficient numbers of voluntary participants. Because of this we have taken great care to factor the block design to allow us to test all three treatments, with a minimum amount of experiment participants. To do this we identify that the scenario profile created by an individual, i.e. treatment 1, may also be regarded as a preparation for a group meeting, i.e. treatment 3. This is exploited in the design of the experiment by having the group members prepare their own scenario profile that is collected and distributed before the group meeting is held and the

group creates the group scenario profile. Thus, data for treatment 1 is collected as part of treatment 3. This way we reduce the required number of subjects by half.

3.4 Analysis of Results

In order to confirm or reject any of the hypotheses, we need to rank the scenario profiles. The ranking between two scenario profiles must, at least, allow for deciding whether one scenario profile is better, equivalent, or worse than another scenario profile. The problem is that the profile is supposed to represent the future maintenance of the system and hence, the best profile is the one that is the best approximation of that. In the case where historical maintenance data is available, we can easily rank the scenario profiles by comparing each profile with the actual maintenance activities. However, for the project requirement specifications used in the experiment, no such data is available.

Instead we assume that the consensusof all scenario profiles generated during the experiment, i.e. a synthetic reference profile, can be assumed to be reasonably close to a scenario profile based on historical data. Consequently, the reference profile can be used for ranking the scenario profiles.

When conduction the experiment we will get 20 scenario profiles divided on two projects, 12 individually created, and 8 created by groups. These scenario profiles will share some scenarios and contain some scenarios that are unique. If we construct a reference profile containing all unique scenarios using the scenario profiles generated during the experiment, we are able to collect the frequency for each unique scenario. Each scenario in the reference profile would have a frequency between 1 and 10. Using the reference profile, we are able to calculate a score for each of the scenario profile generated by the experiment by summarizing the frequency of each scenario in the scenario profile. This is based on the assumption that the importance of a scenario is indicated by the number of persons who believed it to be relevant. Consequently, the most important scenario will have the highest frequency. Consequently, the most relevant scenario profile must be composed by the most relevant scenarios and thus render the highest score. By comparing each scenario profile to the reference profile, we can rank the scenario profiles and find out which one is better than the other.

To formalize the above, we define the set of all scenario profiles generated by the experiment $Q = \{P_1, ..., P_{20}\}$, where $P_i = \{s_1, ..., s_n\}$. The reference profile *R* is defined as $R = \{u_1, ..., u_m\}$ where u_i is a unique scenario existing in one or more scenario profiles *P*. The function $f(u_i)$ returns the number of occurrences of the unique scenario in *Q*, whereas the function $m(s_i)$ maps a scenario from a scenario profile to a unique scenario in the reference profile. The score of a scenario profile can then be defined as follows:

$$score(P_i) = \sum_{x=1}^{n_P} f(m(s_x))$$

3.5 The Selected Projects

Two requirements specifications have been selected from two different projects. Project Alpha is the requirements specification of the prototype for a successor system of an library system called BTJ 2000. This system is widely used in public libraries in Sweden. The system is becoming old-fashioned needs to be re-newed to enter the market of university libraries. The old system is built on a Unix server and connected text-based terminals. The new system must have a graphical user interface to increase the user friendliness. In addition the new system want to increase the possibility for library customers to self service. The new system is to make use of new technologies such as java and the world-wide-web.

The requirements specification of project Beta defines a support and service application for haemo dialysis machines. The new application shall aid service technicians in error tracing when the system behaves in erroneous ways or for doing diagnostics on the system for fault prevention.

Both projects have been performed by teams of between 10 and 15 students as part of their software engineering education with commercial companies as customers and represent commercial software applications. In fact, one of the projects resulted in a ready product that has been included in the product portfolio of the customer.

3.6 Operation

The experiment is executed according to the following steps: (schedule in figure 2)

1. A selection of individuals with varying programming and design experience are appointed.
2. All individuals receive a presentation/tutorial of the method. This is done as a part of the experiment briefing. A document describing the method is also available for all individuals to study during the experiment.
3. Each person fills in a form with some data about his or her experience and knowledge.
4. Individuals are assigned to groups of three using the matched pairs principle (see section 3.8). We planned to involve 12 subjects divided into 4 groups, i.e. group A through D.
5. The groups are assigned a ‘treatment’ and the requirement specification for the first project is handed out. The group A and B that are assigned to the prepared group profile method, start on individual basis which is part of both treatment 1 and treatment 3.
6. When 1.5 hours of time have passed the profiles of the individuals are collected during a short break. During the break the individual profiles are photocopied and handed back to the respective authors. Great care must be taken in that the profile returns to the correct person without any other person getting a glimpse.
7. After the break groups A and B continue in plenum and each group prepares a group scenario profile.

8. At noon, all the group profiles are collected and groups proceed to lunch.
9. After lunch, the process is repeated from step 5, but groups A and B now produce a group profile from start, and groups C and D begin with preparing an individual scenario profile before proceeding in plenum to produce their respective group scenario profile.

Time	Group A	Group B	Group C	Group D	Project
08.00	Introduction and experiment instructions				
09.00	Individual Profile Preparation	Individual Profile Preparation	Unprepared Group	Unprepared Group	Alpha
10.30	Prepared Group	Prepared Group			
12.00	LUNCH BREAK				
13.00	Unprepared Group	Unprepared Group	Individual Profile Preparation	Individual Profile Preparation	Beta
14.30			Prepared Group	Prepared Group	
16.00	De-briefing				

Figure 2: Experiment One Day Schedule

All information collected during the experiment is tracked by an identification code also present on the personal information form. Consequently, the data is not anonymous, but this is, in our judgement, not an imminent problem since the data collected is not, in any clear way, directly related to individual performance. Instead being able to identify persons that had part in interesting data points is more important than the risk of getting tampered data because of lack of anonymity.

3.7 Data Collection

The data collection in this experiment is primarily to collect the results of the work performed by the participants, i.e. the scenario profiles. However, some additional data is required, for example, a form to probe the experience level of the participants. The following forms are used:

- personal information form (Appendix I)
- individual scenario-profile form (Appendix II)
- group scenario-profile form (Appendix III)

The forms have been designed and reviewed with respect to gathering the correct data and ease of understanding, since misunderstanding the forms pose threats on the validity of the data. The personal information form is filled in by the participants after the introduction and collected immediately The others forms are collected during the experiment. During the experiment briefing all forms are presented and explained.

The Personal Scenario Profile form (Appendix II) is handed out to the experiment subjects at the beginning of the Individual Profile Preparation activity. It will be collected at the end of the activity and photocopies will be made for archives.

The Group Scenario Profile form (Appendix III) is handed out to the groups, along with the respective individuals completed profile form, at the start of the Group Synthesis Consensus activity.

3.8 External Threats

Some external threats can be identified in the experiment design, e.g. differences in experience and learning effects. For the most part of the identified threats measures have been taken to eliminate these by adapting the design. By using the blocked project design we eliminate, for example, the risk of learning effects, and in the case of differences in participants experience we use the matched pairs technique when composing the groups, to ensure that all groups have a similar experience profile.

Although precautions has been taken in selecting a system from a large student project which is the result of a 'real' customers demands. This cannot absolutely exclude that the system is irrelevant. The industry customer often use this kind of student projects as proof of concept implementations. However, there are no reasons for the scenario profile prediction method not to be applicable in this situation like the above mentioned. And for the purpose of the experiment we feel that it is more crucial to the results that the individuals in the project have no experience with the particular system’s successors.

3.9 Internal Threats

The internal validity of the experiment is very much dependent on the way we analyze and interpret the data. During the design, preparation, execution and analysis of the experiment and the experiment results, we have found some internal threats or arguments for possible internal validity problems. We discuss them are their impact in the following subsections.

Ranking Scheme. Some problems exists with this method of ranking. First, the reference profile will be relative to the profiles since it is based on them. Second, there might be one single brilliant person that has realized a unique scenario that is really important, but since only one profile included it, its impact will be strongly reduced.

The first problem might not be a problem, if we accept the assumption that the number of profiles containing a particular scenario is an acceptable indicator of its relevance. In the case of having significant differences between the individually created profiles and the group profiles, the differences will be normalized in the reference profile. Given that the individually prepared profiles are more diverse than the profiles prepared by groups, those profiles will render on average lower rank scores, while the group profiles will render on average higher rank scores. In case the results of the experiment is in favor to the null-hypothesis, we will not be able to make any distinction between the group prepared profiles or the individually prepared profiles ranking scores.

The second problem can be dealt with in two ways. First we can make use of the delphi method or the wide band delphi [7]. In that case we would simply synthesize the reference profile, distribute it and have another go at the profiles and get more refined versions of

the profile. The second approach is to make use of the weightings of each scenario and make the assumption that the relevance of a scenario is not only indicated by the number of profiles that include it, but also the weight it is assigned in these profiles. The implication of this is that a scenario that is included in all 20 of the profiles but has a low average weighting, is relevant but not significant for the outcome. However, a scenario included in only one or a few profiles is deemed less relevant by the general opinion. If it has a high average weighting, those few consider it very important for the outcome. Now, we can incorporate the average weighting in the ranking method by defining the ranking score for a profile as the sum of rank products (the frequency times the average weighting) of its scenarios. This would decrease the impact of a commonly occurring scenario with little impact and strengthen the less frequent scenarios with higher impact.

Our conclusion, however, is that the ranking scheme used in this paper does not suffer from any major threats to the validity of the conclusions that we base on it.

Technique itself based on hypothesis. The ranking of profiles is based on the assumption that frequent scenarios are more important and, thus, lead to higher scores. A possible threat to internal validity could be that this would be beneficial for, especially prepared, group profiles since many of the scenarios in the group profile will also be present in one or more of the individual profiles of the group members. One could suspect that the ranking technique is biased towards prepared groups and consequently implicitly favors the hypotheses we hope to confirm.

A closer analysis shows that scenarios that are included in the profiles defined by prepared groups indeed have higher frequencies. However, this does not just benefit the score for the prepared group profile, but also the individual profiles of the group members. Since both profile types benefit, this does not influence the outcome of the experiment.

The analysis technique biased for quantity instead of quality. It could be the case that a profile reaches a high score by using many, unimportant scenarios. Some scenarios may not even be related to the project. This profile, that intuitively should obtain a low score, scores higher than a profile with fewer, but more important scenarios.

When we examine the example closer, we find that the first profile in the example could render a maximum score of 60, because of the limitation on six categories and ten scenarios in each category. A profile with only ten scenarios would have to score on average more than six per profile to out rank the long profile. In the first profile we would get a ratio of the number of scenarios in the profile and the score for that profile of exactly one. In the other profile example, the ratio would be more then one. In figures 8 and 9, the ratios are presented for the projects.

Concluding, although this may, theoretically, be an internal validity threat, it did not occur in the experiment reported in this paper.

The coding of the scenarios to produce the reference profile is biased towards one of the proposed hypotheses. To create the reference profile all scenarios are put together in a table, i.e. the union of all profiles. Since scenarios may be equivalent in semantics in spite of being lexically different, the experimenter needs to establish what scenarios are

equivalent, i.e. coding the data. The coding is done by taking the list of scenarios and for every scenario check if there was a previous scenario describing a semantically equivalent situation. This is done using the database table and we establish a reference profile using a frequency for each unique scenario.

The possible threat is that the reference profile reflects the knowledge of the person coding the scenarios of all the profiles, instead of the consensus among the different profiles. To reduce the impact of this threat, the coded list has been inspected by an additional person. Any deviating interpretations have been discussed and the coding have been updated according to the consensus after that discussion.

4 Analysis & Interpretation

In the previous section, the design of the experiment was discussed. In this section, we report on the results of conducting the experiment. We analyze the data by preparing the reference profiles, calculating the scores for all profiles and determining average and standard deviations for each type of treatment. Finally, the hypotheses stated in section 3.2 are evaluated.

4.1 Mortality

The design of the experiment requires the participation of 12 persons for a full day. For the experiment we had managed to gather in excess of 12 voluntary students, with the promise of a free lunch during the experiment and a nice á la carte-dinner after participating in the experiment. Unfortunately, some students did not show up for the experiment without prior notification. As a result, the experiment participants were only nine persons. Instead of aborting the experiment, we chose to keep the groups of three and to proceed with only three groups, instead of the planned four. As a consequence, the data from the experiment is less complete as intended (see figures 5 and 6). But nevertheless, we feel that the collected data is useful and allow us to validate our hypotheses and make some interesting observations.

Once the experiment had started, we had no mortality problems, i.e. all the participants completed their tasks and we collected the data according to plan.

4.2 Reference profiles

During the experiment we collected 142 scenarios from 9 profiles for project alpha, 85 scenarios from 6 profiles for project beta, totalling 227 scenarios from 15 profiles. The scenarios were coded with references to the first occurring equivalent scenario using a relational database to later generate one reference profile per project. The reference profile for project alpha included 72 scenarios and project beta included 39. The top 10 and top 8 scenarios of the reference profiles are presented in figures 3 and 4. In the alpha case, we note that one scenario has been included in all nine profiles, i.e. has the score nine. In the beta project, we note that the top scenario is included seven times in six profiles. This

could be an anomaly, but when investigated, we recognized that in one of the profiles from the beta project two scenarios have been coded as equivalent. This is probably not the intention by the profile creator, an individual person in this case, but we argue that the two scenarios is only slightly different and should correctly be coded as equivalent to the same scenario in another profile.

Description	frequency
new DBMS	9
new operating system on server	7
new version of TOR	7
introduction of smart card hardware	5
additional search capabilities	5
pureWeb (cgi) clients	4
support for serials	4
new communication protocol	4
user interface overhaul	4
new java technology	4

Figure 3: Alpha Reference Profile Top 10

Description	frequency
remote administration	7
upgrade of database	6
upgrade of OS	6
real-time presentation of values	4
change of System 1000 physical components (3-4 pcs.)	4
rule-based problem-learning system	3
change from metric system to american standard	3
new user levels	3

Figure 4: Beta Reference Profile Top 8

Another interesting observation to make is that among the top three scenarios in both projects we find changes of the database management system and changes of the operating systems, either new version or upgrade and we find it in just about all the profiles. This suggests that these two changes to a system are among the first scenarios that come to mind when thinking about future changes to a system.

Finally, it is worth noting that the major part of the top scenarios are related to interfacing systems or hardware. Only a few of the scenarios in the top 10 or 8 are related to functionality specific to the application domain, e.g. “support for serials” in figure 3 or “new user levels” in figure 4.

4.3 Ranking & Scores

In this section, we presented the coded and summarized data collected from the experiment. In the table presented in figure 5 the score for each of the profiles generated for the Alpha project are presented. It is interesting that group A and B, that both are prepared groups, score strikingly high scores, compared to the other profiles in project Alpha. Further, we notice little difference between the profiles created by the individual persons and the unprepared groups, in neither project.

Identity	Members	Profile Score	Remarks
group A	C,D,E	72	*individual preparation*
group B	H,I,F	77	*individual preparation*
group C	A,B,G	49	*no individual preparation*
Arthur			*only participated in a group*
Bertram			*only participated in a group*
Charlie		39	
David		38	
Ernie		45	
Frank		19	
Gordon			*only participated in a group*
Harald		66	
Ivan		55	

Figure 5: Project Alpha

In the table in figure 6, the profile scores for project Beta are presented. The prepared group, C in this case, scores a very high score, but the unprepared groups, A and B, score much less. This is interesting since the groups members are the same for both projects.

Identity	Members	Profile Score	Remarks
group A	C,D,E	44	*no individual preparation*
group B	H,I,F	37	*no individual preparation*
group C	A,B,G	72	*individual preparation*
Arthur		36	
Bertram		43	
Charlie			*only participated in a group*
David			*only participated in a group*
Ernie			*only participated in a group*
Frank			*only participated in a group*
Gordon		33	
Harald			*only participated in a group*
Ivan			*only participated in a group*

Figure 6: Project Beta

In figure 7 the average scores for each type of treatment is presented for the Alpha, Beta project and in total. In addition, the standard deviation over the scores and the number of cases is presented. The average score for prepared groups is substantially higher than the score for unprepared groups or individuals. Secondly, the standard deviation is the largest for individuals, i.e. 13, but only 6 for unprepared groups and 3 for prepared groups. Finally, it is interesting to note that the standard deviation for all profiles is larger than for any of the treatments, which indicates that the profiles for each type of treatment are more related to each other than to profiles for other treatment types.

Treatment	Alpha	Beta	Total	Std. Dev.	#cases
Individual	43	37	*41*	*13*	*9*
Unprepared group	39	40	*43*	*6*	*3*
Prepared group	75	72	*74*	*3*	*3*
Total	51	44	*48*	*17*	*15*

Figure 7: Average and Standard Deviation Data

In section 3.9, we discussed various threats to the internal validity of the experiment. One of the discussed threats is the risk that a profile with many unimportant scenarios scores higher than a profile with fewer, but more important scenarios, while this is counter intuitive. Based on the data in figure 8 and 9, we can conclude that although a theoretical threat was present, it did not occur in the experiment.

Identity	Profile Score
group B	77
group A	72
Harald	66
Ivan	55
group C	49
Ernie	45
Charlie	39
David	38
Frank	19

Identity	Ratio
Ernie	4,5
Harald	3,9
group A	3,8
Charlie	3,5
group B	3,3
David	3,2
Ivan	2,9
group C	2,7
Frank	1,5

Identity	Profile Length
group B	23
group A	19
Ivan	19
group C	18
Harald	17
Frank	13
David	12
Charlie	11
Ernie	10

Figure 8: Project Alpha

Identity	Profile Score
group C	72
group A	44
Bertram	43
group B	37
Arthur	36
Harald	33

Identity	Ratio
Arthur	3,6
group A	3,4
Bertram	3,3
group C	3
Harald	2,8
group B	2,8

Identity	Profile Length
group C	24
group A	13
group B	13
Bertram	13
Harald	12
Arthur	10

Figure 9: Project Beta

4.4 Evaluating the Hypotheses

The experiment data does not allow us to identify any significant difference in ranking between profiles created by an independent person or profiles created by a group with unprepared members. Hence we *cannot dismiss* the null hypothesis, $\mathbf{H_{01}}$.

The first null hypothesis, $\mathbf{H_{01}}$, counters the two hypotheses $\mathbf{H_1}$ and $\mathbf{H_{10}}$. Since the experiment data does not allow us to dismiss the null hypothesis $\mathbf{H_{01}}$, we cannot expect to validate those two hypotheses and therefore, we can *dismiss* $\mathbf{H_1}$ and $\mathbf{H_{10}}$. We can, however, make an *interesting observation* on the variation in the ranking scores between the profiles of the individuals and the unprepared groups. The scores of the profiles created by independent persons range from 19 to 62 over both projects, while the scores of the profiles created by the unprepared groups only ranges from 32 - 49 over both projects. The observation is also supported by the standard deviation values presented in figure 7. This suggests that using unprepared groups does not lead to higher scores on the average, but provides more stable profiles and reduces the risk for extreme results, i.e. outliers.

With respect to the second null hypothesis, $\mathbf{H_{02}}$, we find compelling evidence in the analyzed data for a significant difference between the profiles created by individuals, with an score average of 43, and profiles created by a group with prepared members, with an score average of 74 (see figure 7). We also observe that no profile created by an independent person has scored a higher score than any profile created by a group with prepared members. Hence, we can *dismiss* the second null hypothesis, $\mathbf{H_{02}}$.

Because we were able to dismiss the second null hypothesis, it is worthwhile to examine the two related hypotheses, $\mathbf{H_2}$ and $\mathbf{H_{20}}$. The scores clearly show that the group with prepared members in all cases have scored higher than the profiles created by independent persons. This allows us to *confirm* of the hypothesis, $\mathbf{H_2}$ and allow us to *dismiss* the counter hypothesis, $\mathbf{H_{20}}$.

The last null-hypothesis is $\mathbf{H_{03}}$. With respect to this hypothesis, we find evidence that a significant difference exists between the average scores for unprepared and prepared groups. Profiles created by groups with prepared members score 74 on average, as opposed to profiles from groups with unprepared members, that score 41 on average. Hence, we *can dismiss* the null hypothesis, $\mathbf{H_{03}}$, and evaluate the related hypotheses $\mathbf{H_3}$ and $\mathbf{H_{30}}$. The average score for prepared groups is 74, which is considerably higher than the average score for unprepared groups, i.e. 41. Based on this, we are able to *confirm* hypothesis $\mathbf{H_3}$ and, consequently, *dismiss* the counter hypothesis $\mathbf{H_{30}}$.

5 Related Work

Architecture assessment is important for achieving the required software quality attributes. Several authors propose and advocate scenario based techniques for architecture assessment. A well-known method is the scenario-based architecture

assessment method (SAAM) [12]. SAAM assesses the architecture after the architecture design and incorporates all stakeholders of the system. Other methods include the architectural trade-off analysis method (ATA) [11] that uses scenarios to analyze and bring out trade off points in the architecture. The 4+1 View method [13] uses scenarios in its fifth view to verify the resulting architecture. To this point, no studies have been reported on the creation of scenario profiles for architecture assessment.

In [3] a framework for experimentation in software engineering is presented along with a survey of experiments conducted up to 1986. In our work with the experiment design we have used this framework to ensure an as robust design as possible.

6 Conclusions

During recent years, the importance of explicit design of the architecture of software systems is recognized. This is because the software architecture constrains the quality attributes of the system. Consequently, architecture assessment is important to decide how well the software architecture supports various quality attributes. One can identify three categories of architecture assessment techniques, i.e. scenario-, simulation- and static model-based assessment. However, these techniques make all use of scenario profiles. Although some scenarios profiles can be defined as 'complete', i.e. covering all scenarios that can possibly occur, most scenario profiles are 'selected'. *Selected scenario profiles* contain a representative subset of the population of all possible scenarios.

Scenario profiles are generally defined as a first step during architecture assessment. However, defining a selected scenario profile is subjective and we have no means to decide upon the representativeness of the profile. Also, to the best of our knowledge, no studies are available about the effects of individuals on the definition of scenario profiles, i.e. what is the deviation between profiles defined by different individuals. The same is the case for groups defining scenario profiles.

In this paper we have presented the design and results of an experiment on three methods for creating scenario profiles. The methods, or treatments, for creating scenario profiles that were examined are (1) an individual prepares a profile, (2) a group with unprepared members prepares a profile and (3) a group with members that, in advance, created their individual profiles as preparation.

We also have stated a number of hypotheses, with the corresponding null-hypotheses and, although, the results of the experiment data do not allow us dismiss each of our null-hypotheses, we find support for the following hypotheses:

> **H_2** = *Scenario profiles created by groups with prepared members, generally get better scores than scenario profiles created by an individual person.*
>
> **H_3** = *Scenario profiles created by groups with prepared members generally get better scores than group profiles with unprepared members.*

Thus, based on the experiment data, we are able to conclude that using groups with prepared members is the preferable method for preparing scenario profiles.

In addition we have also made a number of observations during the experiment and during the analysis of the data. These are as follows:

1. Two change scenarios occurring in just about all the profiles were new version or upgrade of the database management system and the operating system.
2. Few scenarios among the top 10 or 8 are related to the application, instead most of the scenarios in the top are related to interfacing systems or hardware.
3. The standard deviation in score is lower for profiles created by unprepared groups, than for individuals, although the average of the profiles scores cannot not be said to differ significantly between the two. A plausible interpretation is that the group reduces the variation by filtering out the scenarios that are questionable, in contrast to the individually created profiles.

Acknowledgments

We would like to thank the students who participated in the experiment.

References

[1] G. Abowd, L. Bass, P. Clements, R. Kazman, L. Northrop, A. Moormann Zaremski, *Recommend Best Industrial Practice for Software Architecture Evaluation,* CMU/SEI-96-TR-025, 1997.

[2] Basili, V.R., Selby, R.W., Hutchens, D.H., *"Experimentation in Software Engineering"*, IEEE Transactions on Software Engineering, vol. se-12, no. 7, July, 1986

[3] L. Bass, P. Clements, R. Kazman, *'Software Architecture In Practise'*, Addison Wesley, 1998.

[4] P. Bengtsson, '*Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics*', First Nordic Workshop on Software Architecture (NOSA'98), Ronneby, August 20-21, 1998.

[5] P. Bengtsson, J. Bosch, 'Scenario Based Software Architecture Reengineering', *Proceedings of International Conference of Software Reuse 5 (ICSR5)*, Victoria, Canada, June 1998.

[6] P. Bengtsson, J. Bosch, 'Architecture Level Prediction of Software Maintenance', *Proceedings of Third European Conference on Software Maintenance and Reengineering*, pp. 139-147, March 1999.

[7] Boehm, B.W, *Software Engineering Economics*, Prentice Hall, 1981.

[8] J. Bosch, P. Molin, 'Software Architecture Design: Evaluation and Transformation', *XXXXXXXXXXXX*

[9] J. Carrière, R. Kazman, S. Woods, "Assessing and Maintaining Architectural Quality", in proceedings of The Third European Conference on Software Maintenance and Reengineering (CSMR'99), IEEE Computer Society, pp. 22-

30, 1999

[10] J.C. Dueñas, W.L. de Oliveira, J.A. de la Puente, 'A Software Architecture Evaluation Method,' *Proceedings of the Second International ESPRIT ARES Workshop,* Las Palmas, LNCS 1429, Springer Verlag, pp. 148-157, February 1998.

[11] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, J. Carriere, The Architecture Tradeoff Analysis Method, Proceedings of ICECCS, (Monterey, CA), August 1998

[12] R. Kazman, L. Bass, G. Abowd, M. Webb, 'SAAM: A Method for Analyzing the Properties of Software Architectures,' *Proceedings of the 16th International Conference on Software Engineering*, pp. 81-90, 1994.

[13] P.B. Krutchen, 'The 4+1 View Model of Architecture', *IEEE Software*, pp. 42-50, November 1995.

[14] W. Li, S. Henry, 'Object-Oriented Metrics that Predict Maintainability', *Journal of Systems and Software,* vol. 23, no. 2, pp. 111-122, November 1993.

Appendix I

Individual Information Form

Identification:	_____
Started SE Curriculum:	❑ 1994 ❑ 1995 ❑ 1996 ❑ 1997
Working Since:	_____ (Year)
Number of Study Points:	_____
Maintenance Experience:	_____ (Years/Months)
Software Development Experience:	_____ (Years/Months)

Knowledge in:

C	❑ None	❑ Novice	❑ Skilled	❑ Expert
C++	❑ None	❑ Novice	❑ Skilled	❑ Expert
Java	❑ None	❑ Novice	❑ Skilled	❑ Expert
Eiffel	❑ None	❑ Novice	❑ Skilled	❑ Expert
Pascal/Delphi	❑ None	❑ Novice	❑ Skilled	❑ Expert
Visual Basic	❑ None	❑ Novice	❑ Skilled	❑ Expert
Assembly language	❑ None	❑ Novice	❑ Skilled	❑ Expert

Software Modelling Experiences:

Booch	❑ None	❑ Novice	❑ Skilled	❑ Expert
OMT	❑ None	❑ Novice	❑ Skilled	❑ Expert
Objectory	❑ None	❑ Novice	❑ Skilled	❑ Expert

Training in Software Architecture:	_____ (Days or Credits)
Requirements Experience:	_____ (Years / Months)

Appendix II

Individual Scenario Profile

Identification: _____

Project: ❑ Alpha ❑ Beta

Previous Domain experience: _____ Years

No.	Cat.	Scenario Description	Weight
S1			
S2			
S3			
S4			
...			
S80			

ID	Category
C1	
C2	
...	
C10	

Appendix III

Group Scenario Profile

Project: ❑ Alpha ❑ Beta

Group Identification: _____

Identification: _____ & _____& _____

Previous Domain Experience: _____ & _____& _____ Years

No.	Cat.	Scenario Description	Weight
S1			
S2			
S3			
S4			
...			
S80			

ID	Category
C1	
C2	
...	
C10	