Abstract. When creating an architecture for a software system it is important to consider many aspects and different sides of these aspects at an early stage, lest they are misunderstood and cause problems at later stages during development. In this paper we report from an industry study to understand and select between different architecture candidates. The company uses a method that focuses on discussions of architecture candidates to where there is disagreements between the participating domain experts. The results indicate that the used method pinpoints for the company where further investigations are necessary and that the decision concerning which architecture to use is taken with more confidence as a result of the focused discussions.

1 Introduction

When developing software, it is important to have an appropriate architecture for the system, or sub-systems comprising the full system. The choice of, or evolution into, an appropriate architecture is not only governed by functional requirements, but to a large extent by quality attributes [Bass et al. 1998][Bosch 2000][Hofmeister et al. 2000].

However, knowing this, it is still a non-trivial task to create an appropriate architecture. There are usually more than one quality attribute involved in a system, and the knowledge of the benefits and drawbacks of different software architectures with respect to different quality attributes is not yet an exact science. Decisions are often taken on intuition, relying on the experience of senior software developers, which may lead to the developed software systems being forced into a mould which may not be entirely suitable for the problem at hand.

As shown by [Johansson et al. 2001] different stakeholders tend to have different views of the importance of various quality requirements for a system, and the differing experiences of the software developers may also lead to a different interpretation of the strengths and weaknesses of architecture candidates. A structured method facilitates in this situation because it enables us to identify where stakeholders and developers have differing opinions.

This paper presents a study of a company facing the task of identifying which among a set of architecture candidates that have the most potential for fulfilling the quality requirements of a real life system to build. The company also desires an understanding shared by everyone involved of the architecture candidate ultimately chosen, in order to avoid problems later on during the development process. To this end the company is using a structured method to extract and quantify the opinions of the people involved in the software architecture decision process. The outcome of this is then used to hold focused discussions to reach a joint understanding of the benefits and liabilities of each architecture candidate, and to identify where further investigations are necessary. This, in turn, increases the confidence that a well-informed and correct architecture decision is taken.

The method used captures the perceived potential of the architecture candidates to fulfill a blend of quality attributes. There are also many qualities that are not visible in a software architecture, e.g. the development organisation. Hence the method provides one among many important inputs to decision-makers when selecting a suitable software architecture.

The remainder of this paper is organised as follows. Related work is presented in Section 2. In Section 3 we describe the background and motivation for this paper. Section 4 describes the research questions for this study. In Section 5 the planning of the study is presented, to set the foundation for the results obtained. The operation of the study is described in Section 6. We provide the results of the study, and an analysis to provide answers to the research questions in Section 7. Finally, a summary and some conclusions are presented in Section 8.

2 Related Work

Architecture evaluations can be separated into early architecture evaluations and late architecture evaluations [Lindvall et al. 2003], and different methods are more or less suitable for either of these two types.

Late architecture evaluation is conducted during later stages of the software development process when there is a software system or at least a detailed design available on which more concrete metrics can be collected. [Lindvall et al. 2003] is an example of this.

Early architecture evaluation, on the other hand, is concerned with deciding at an early stage during the software development process what qualities a software architecture have a potential for exhibiting.

Early architecture evaluations are commonly based on the experiences of the software developers and logical reasoning, as there are usually no tangible artifacts...
on which to e.g. perform simulations or collecting metrics. Oftentimes, this is aided by first specifying, categorizing and prioritizing scenarios. These scenarios then enables the evaluators to focus on one issue at a time.

Examples of evaluation methods focused on early evaluation and using scenarios are the Software Architecture Analysis Method (SAAM) [Bass et al. 1998] that is solely based on the use of scenarios, its successor the Architecture Tradeoff Analysis Method (ATAM) [Clements et al. 2002] that is more flexible in the possible evaluation techniques, various methods focusing on specific quality attributes such as modifiability [Bengtsson 2002], and some techniques such as Global Analysis [Hofmeister et al. 2000] that are more focused towards architecture creation than architecture evaluation (although the process is driven though continuous evaluations).

Part of the evaluation in ATAM may be done by using Attribute Based Architectural Styles (ABAS), which is basically a description of a particular architecture style aimed at a particular quality attribute. Hence, an ABAS includes a certain architecture style and a specific method for evaluating architectures based on this style with respect to a certain quality attribute.

While these methods tend to be performed as a group activity, the method used in this paper allows each participant to first form his or her own opinion, and then the method facilitates in focusing discussions around those issues where the involved parties have different opinions. This means, of course, that issues that all participants agree on are not covered and that the outcome is an increased joint understanding of what needs to be done.

Moreover, the methods above are focused on evaluating a single architecture to find out if and where there may be problems in it. The method in this paper is more aimed towards finding out which architecture candidate, of a set of architecture candidates, has the most potential to support the mix of quality attributes for a particular system to build. Accordingly, the method used in this paper does not produce any absolute measures on the architecture candidates. Rather, it produces judgements relative to the other architecture candidates on how good or bad the architecture candidates are with respect to a particular quality attribute, much in the same way as [Morisio et al. 2002] compares software artifacts with predefined ideal artifacts.

A final difference is that the method in this paper is not based on scenarios as the aforementioned methods are (except for [Morisio et al. 2002]). This may, however, be a possible and interesting future extension.

### 3 Background

The company in this study uses a decision support method described in [Svahnberg et al. 2002]. This is a method that has previously been evaluated in an academic setting [Svahnberg & Wohlin 2002a]. The study in this paper extends on this and investigate the effects of having a known target system (as opposed to the generic setting used in the academic setting) in an industrial setting, and also emphasize the discussions that subsequently follow as a part of the method. We thus focus this study on the discussion driving aspects that creates an increased joint understanding and study the influence of having a known target system and a real problem to address.

### 3.1 Method Outline

The method used in this paper, based on [Svahnberg et al. 2002], consists of the following steps:

- **Prepare Data.** In this step we create and describe the software architecture candidates and the quality attributes to use in the study.

- **Create Individual Frameworks.** Each of the participant in the study completes a questionnaire to create an individual version of two sets of vectors (i.e. two sets of vectors for each participant in the study are created). These two sets of vectors we refer to as a Framework for Quality Attributes (FQA) and a Framework for Architecture Structures (FAS), respectively. Note that we use the term “architecture structure” very loosely here. We use it to mean an architecture candidate for any piece of software of any size - be it a component, a subsystem or an entire software system. These two sets of vectors rate the architecture candidates in terms of the quality attributes from two different perspectives.

  In addition, each participant in the study also creates a prioritized list of the quality attributes required for the system to build, called a Prioritized List of Quality Attributes, or PQA.

- **Analyse Individual Frameworks.** The individual frameworks are analysed for internal consistency and are then compared to the other individual frameworks. The outcome of this step is to identify where the participants are in agreement and where there are differing opinions regarding the strengths and weaknesses of the different architecture candidates.

- **Discuss Individual Frameworks.** In this step the points where there are disagreement, as identified in the analysis of the individual frameworks, are discussed in a discussion meeting. These discussions are used to create a joint understanding of the benefits and liabilities of the software architecture candidates, and to identify where more work is needed to really understand the impact of the quality attributes. Moreover, the individual frameworks are combined into a unified view consisting of a joint FQA, FAS and PQA.
Suggest Software Architecture. The PQA is used together with the FQA and the FAS to calculate which architecture candidate best match the quality requirements of the system. However, we do not suggest that this recommendation should be used without reflection. Instead, we suggest that the recommendation is used to spark discussions during the discussion meeting of the level of support given by the different architecture candidates for the desired mix of quality attributes (as represented by the PQA). We suggest that a median value of the individual frameworks are used to generate the recommendation before the discussions are held. We have found that using the median value is sufficient in most cases. Should the discussions conclude otherwise the individual frameworks can be combined in a different way after the discussion, before a software architecture is finally decided on.

4 Research Questions

The research questions addressed, in addition to providing results to the company, are the following:

**Q1.** Are decisions strengthened by executing a structured comparison method that generates quantifiable results to the company, are the following:

We use the method outlined in Section 3.1, and the evaluation of this question consists of two parts: (a) whether the method produces relevant results, and (b) whether the use of the method affects the confidence of the participants in taking the right decision.

We measure this by interviewing the participants after the execution of the method and the discussion meeting. We could have reduced the measurement to a simple question with an answer on an ordinal scale, but we feel that it is more beneficial to use a qualitative way with open questions in order to elicit more of the rationale as to why the decision is made with more, less or the same confidence.

We also wish to ascertain that the structured method we use provide relevant results. To this end we compare the architecture recommended by the method with what architecture the participants would prefer to use if they decide based on their experience and without any extra knowledge. If these match, we know that the discrepancies found between participants, and the discussions held are in line with the participants’ intuitive sense of what is good and bad with the different architecture candidates.

**Q2.** Is there a significant difference in the spread of the answers if the target system is known beforehand?

This question is interesting because it would give a partial answer to whether one can create generic decision frameworks based on architecture styles without considering a particular target system. This might be possible if there is an equal spread among the answers regardless of whether the target system is known beforehand or not, whereas if the answers are less spread if the target system is known, there would be more uncertainty in any statement about generic architecture styles.

We measure Q2 by calculating the squared distance to the mean for each participant and data point according to the following formula: \((x_i - \bar{x})^2\) and then compare the values obtained from this study and from a previous study [Svähnberg & Wohlin 2002a] where the target system was not known.

5 Planning

In this section we describe the planning and design conducted for the architecture decision method used by the company. Parts of the design also aims to gather data to answer the questions posed in Section 4.

5.1 Context

The study is conducted in an industry setting, with experienced practitioners of software engineering. Specifically, we conduct the experiment together with Danaher Motion Särö AB, a Swedish company that develops software for automatic guided vehicles (AGVs) which are, for example, used to automatically haul goods on factory floors.

Danaher Motion Särö AB employs approximately 20-25 software engineers who have all been working in the company for more than five years. Today, there are more than 1000 installations of their system worldwide.

Within this setting we are studying a particular situation where a redesign of the existing system allows the company to question the previous architectures (logical as well as physical architectures), in order to possibly obtain an architecture that better suit the quality requirements. We focus on a single software system from the product line of systems developed by the company, of which most are necessary to get an AGV system operational.

5.2 Variables

The architecture decision method uses two sets of variables, namely which architectures and which quality attributes to include in the study. Below, we describe these further.

Architectures. Three architecture candidates were developed for the subsystem in focus of our study. For reasons of confidentiality we are unable to give the full picture of the architecture candidates, but the architecture candidates describe three variations of how to physically distribute the logic among a set of hardware components. Specifically, the architectures involve:

1. http://www.danahermotion.se
• A one-CPU solution, where the entire system in question is gathered on a single CPU board. We refer to this as Architecture A.
• A two-CPU solution, where parts of the system have been placed on a separate CPU and communication is done over a generic communications bus. We refer to this as Architecture B.
• A centralised solution, where the first CPU has been replaced with execution cycles in a central computer, shared over all vehicles. We refer to this as Architecture C.

These architectures were developed by a person from the company, and described in diagrams and text together with a description of the functional scope of the system, also described with diagrams and text. The description of each architecture consists of two pages: one with a UML diagram and one with an explanatory text.

The architectures were deliberately described on a rather high level. One reason for this is that for the method to be useful, it has to be able to work with early sketches of architectures created with a small effort. If too much work is needed on the architectures for the evaluation, it will simply be too expensive. Moreover, a more detailed description introduces a risk that the creator of the architecture candidates may have an undue influence on the possible interpretation of the architectures. By having less detailed descriptions, we give more room for the participants to make their own interpretations of the architecture candidates.

Naturally there is a limit to how high a level the architectures can be described on, but since the problem domain and the system’s domain is known to the involved participants, we believe the architectures can nevertheless be described on a relatively high level.

Quality Attributes. After the architectures were developed, five quality attributes relevant for this system were identified and described. We waited with this until the architectures were developed to be sure that the quality attributes were indeed relevant.

The quality attributes identified, and the descriptions of them that were distributed to the participants, are:

• Cost. This quality attribute involves aspects such as development cost, maintenance costs and production cost, but also indirectly time-to-market. Furthermore, the ability to understand the system and the ability to allocate resources with the right competence is also indirectly included in this attribute.
• Functional Flexibility. With this we mean the ability to configure and deliver with different subsets of functionality, but also the ability to add new functionality and remove or replace old functionality. To deliver with different subsets of functionality mainly means adding different sets of functionality on top of a common core, but it may also mean that the components (both hardware and software) of the common core may need to be extended or replaced to accommodate parts of the desired functionality.
• Performance Flexibility. This quality attribute concerns the ability to configure different software and hardware solutions that supports different maximum speeds. This may have impact on the hardware (e.g. faster CPUs), but may also influence the rate at which software components must communicate, or even the algorithm by which the software controls the vehicle.
• Security Flexibility. With this we mean the ability to configure and deliver solutions with varying degrees of security and safety. This can, for example, be done by adding redundant hardware and software, which means that the architecture must support redundant components, both in the software and in the hardware.
• Testability. Testability involves the ability to test and debug the parts of the system individually and/or as a group, the ability to extract test data from the system and the ability to add data measurement points when needed, but also the ease by which these things can be done.

5.3 Subjects
14 subjects at the company were involved to various degrees in the project we studied.

Specifically, we invited:
• four of the software engineers in the project,
• one hardware engineer,
• the project leader,
• three product managers who are in charge of the technical planning for the different products in the product line,
• three members of the system group whose responsibility it is to ensure that the different products work together as a system,
• the development manager.

Although two or three of these have more of a management function, their background and expertise is still such that they are able to assess the different architecture candidates from a technical perspective.

5.4 Study Design
The study consists of two parts. In the first part we gather data by letting the participants fill in a questionnaire and in the second part the results from this questionnaire are discussed. In this section we describe these two parts, i.e. the design of the questionnaires and the design of the discussion meeting.

5.4.1 Data Gathering
The questionnaire is constructed using a comparison method from multi-criteria decision making, namely the Analytic Hierarchy Process (AHP) [Saaty 1980][Saaty & Vargas 2001]. Simply put, this is a
method for making pair-wise comparisons. By making all possible pair-wise comparisons it is possible to make a ranking of the entities and to, in addition, calculate a consistency index. AHP is described further in Appendix A.

**Instrumentation.** The participants are requested to fill in a questionnaire which mainly consists of pair-wise comparisons of the different architectures and quality attributes (which were described to the participants before the questionnaire was distributed). These questionnaires consist of four parts:

- **Part A:** Generic questions, such as name of the participant, and also what architecture he or she would prefer based on experience alone. Moreover, we ask some questions whether there are any architectures or quality attributes missing that they think should have been part of the study.
- **Part B:** Questions to obtain a prioritized list of quality attributes (PQA).
- **Part C:** Questions to rate the support given for the quality attributes within each architecture candidate (FAS).
- **Part D:** Questions to rate which architecture is best at each quality attribute (FQA).

The order of the different parts for the different groups of participants is illustrated in Table 1. The participants are randomly assigned to the different groups, but each participant get a unique questionnaire that is generated according to the forms in Table 1 but that is random in all other aspects. The reason for controlling the order in this way and to randomize everything else is, of course, to ensure that the results are independent of the order of the forms, and that the learning effect does not influence the results. Parts B, C and D consist of pair-wise comparisons, and the order of these comparisons is random for each section and for each generated questionnaire.

Our decision to always start with part A in the questionnaires is based on that we want to know which architecture the participants would prefer if they had no additional knowledge, i.e. before using the method from Section 3.1.

### 5.4.2 Discussion Meeting

The second part of this study is performed in the form of a discussion meeting. For this meeting we use the results of the questionnaires and calculate, using the AHP method, nine vectors for each participant (one for the PQA, three for the FAS and five for the FQA).

Table 1: Questionnaire Design

<table>
<thead>
<tr>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Part A</td>
<td>Part B</td>
<td>Part C</td>
<td>Part D</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Part D</td>
<td>Part C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Part C</td>
<td>Part D</td>
<td>Part C</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Part D</td>
<td>Part C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These vectors correspond to the different parts of the questionnaire as described in Section 5.4.1.

For the discussion meeting, graphs are produced for each of these vectors depicting the different participants’ opinions.

As an aid to pinpoint what to discuss, we calculate the sum of the squared distance to the mean over all participants for each value in the nine vectors according to the following formula: \[ \sum_{i=1}^{N} (x_i - \bar{x})^2, \]

where N is the number of participants. This creates nine new vectors that consist of the values calculated according to this formula. We then identify those vectors for which any value is within the fourth quartile of all the values in the new vectors as suitable discussion points. In addition to this, we study plots of the individuals and their answers, to locate other points where discussions may be needed.

Moreover, the median value of all participants for each vector is used to obtain a recommendation of which architecture to use, according to the method in [Svahnberg et al. 2002].

The result of this is a presentation going through the different vectors and other results according to the identified discussion points.

### 5.5 Validity Evaluation

In this section the threats to the architecture decision and our study are discussed. For a thorough description of possible threats to studies, see [Wohlin et al., 2000].

**Conclusion Validity.** As the answer to each question is in the form of a subjective judgement, the answers will not be exactly the same for all participants. Indeed, it is not even certain that a participant will answer exactly the same, should the study be repeated. The reliability of each person’s individual measures is not as high as one would hope. However, it is our belief that the combined measures of all participants is of higher reliability. Moreover, the consistency index within the AHP method helps to check the consistency of the individuals. Furthermore, it is part of the study to measure the amount by which different participants disagree, and it is these disagreements that power the discussions after the questionnaires have been completed.

The questions to answer are of such a character that the answer to each question is independent of when the previous questions were answered. Because of this we see no troubles if the exercise is interrupted by e.g. phone calls, coffee breaks or fire drills.

If, during coffee breaks the participants discuss their answers, this may lead to participants changing their opinion on some questions. However, as a discussion meeting is held at a later stage to create a collective result, the only result of this would be to facilitate the process of reaching a joint understanding.
Our choice to use high-level descriptions of the architecture candidates and quality attributes gives more room for personal interpretations. This also introduces a risk that if a participant favour a particular architecture candidate, there is room to interpret the other architectures such that the favoured architecture gets better scores compared to the other alternatives.

Another source of bias may be that one of the architecture candidates is more similar to the current system than the others. This means that the participants know more about this alternative and may favour it in lieu of the other, more unknown, alternatives.

Moreover, all of the participants were asked to give their opinion of which architecture candidate they would prefer before they completed the actual questionnaire. There is a risk that this creates a bias where one seeks to answer in such a way that this initial choice is confirmed. However, it is our belief that the participants had formed such an opinion even before they started to complete the questionnaire and would hence be biased in the answers regardless of when during this process they are asked to specify which architecture they prefer. To ask this after the questionnaire instead would have introduced a risk that the process of completing the questionnaire may have caused a learning effect and the answer would be biased by having completed the study.

**Internal Validity.** As the participants answer more and more questions, we expect them to become more acquainted with the method, and possibly grow bolder in their answers, spreading them more from the centre value. To counter this effect, the forms are constructed in a random way, as described in Section 5.4. This ensures that questions asked last for one participant, i.e. when the participant has matured, are potentially asked first for another participant, when this person has not yet matured.

In a discussion with the participants after the study, we discovered further threats to the study with respect to the quality attributes used. A general view regarding the quality attribute ‘cost’ was that it was too all-encompassing, which means that it may have not only led to different interpretations, but also that it may have taken up too much of the result space. Since the result from AHP is a normalised vector that sums up to 1, if one alternative scores a very high value the distances between the other elements in the vector is decreased proportionally. I.e. high values on one element reduce the span within which the remaining elements can score.

In retrospect, the quality attribute ‘security flexibility’ should have been split into security flexibility and safety flexibility, of which the second is more important. One reason for this problem is a language issue: security and safety is the same word in Swedish.

Finally, that three of the quality attributes were concerned with flexibility confused matters and some of the participants had troubles distinguishing between e.g. absolute performance and performance flexibility.

**Construct Validity.** We see no threat if the participants are aware of the hypothesis we evaluate, and hence make no secret of it. As we, as researchers, have no interest in favouring any particular architecture or quality attribute, but rather to get a collective view of all chosen architecture candidates and quality attributes, there is no way in which the participants can tweak their answers to satisfy any hidden hypothesis we may have.

We are aware that the method used in this study can also be used to evaluate persons, e.g. to see how close to some company norm their answers fall, and that some persons may perform differently as the result of knowing that they are being evaluated.

To counter this, we considered making an anonymised version of the presentation of the results, but after discussions with the participants we concluded that the company climate is such that nobody is punished for speaking his or her mind.

**External Validity.** Most of the participants in the study are exactly the target group for which the method is supposed to be used, i.e. software developers, so we see no threat in this. However, the remainder of the participants may not be entirely representative, which may mean that the results are not generalisable to other companies. Moreover, it may not be the fortune of every company to have a staff that have so much experience as the company in this study which may increase the disparity between the answers. As always, it is vital to gather experiment data from more companies, and we encourage replications of the presented study.

During the first week of the study, a workshop (unrelated to our study) was conducted at the company concerning the safety concepts of the project. As not all participants in our study also participated in this workshop, the answers with regards to the quality attribute security flexibility may be skewed.

**6 Operation**

In this section we describe the execution of the architecture decision method. We describe the two phases of the study separately. In the first phase we gather data from the participants in the study, which we then use in the second phase, where we discuss the resulting data thus gathered.

**6.1 Data Gathering**

During this phase of the study, the participants individually study the architectures and quality attributes involved and assess the strengths and weaknesses of each of the architectures. This is done according to a given form, as described in the design of the ques-
tionnaires in Section 5.4. In this part of the evaluation each participant individually completes a uniquely generated questionnaire.

**Preparation.** One week before the execution of the study, information was sent out to the participants concerning the purpose and form of the study, including an estimate of how much time we expected them to spend on completing the questionnaire. The participants were also given an opportunity to opt out of the study if they wanted.

**Execution.** On a Monday, each of the participants received an envelope containing a description of the study, an overall description of the architecture problem to solve, a description of the different architectures, a description of the different quality attributes and a questionnaire. They were given until Friday the same week to complete and return the questionnaires. Moreover, they were asked to try and complete the questionnaire as uninterrupted as possible, and not to discuss the architectures or quality attributes with each other.

By Friday, nine participants had handed in their questionnaires. A fortunate power outage during the week helped the study as many took the opportunity to complete the questionnaires during this time.

By the next Friday, four more participants had returned the questionnaire, thus leaving only one participant unaccounted for.

**Data Validation.** As one of the goals is to pinpoint where people disagree, it seems unreasonable to disqualify some participants because their answers are not in conformance to the others.

However, AHP provides, as mentioned in Section 5.4, a consistency ratio. This ratio can be used to determine whether the participants have answered consistently, i.e. in agreement with themselves.

Since we have 13 participants, each answering questions regarding nine topics (one prioritization of quality attributes, ratings of quality attributes in three architectures, and rankings of architecture candidates for five quality attributes) we have a total of 117 such consistency ratios to examine. Of these, 23 were higher than 0.15. This means that there are cases where the participants have not been consistent in their answers, and in some cases even outright contradictory.

With one exception no single participant was consistently inconsistent. Instead, the high consistency ratios were found to be distributed over all participants and over all the nine topics.

Therefore we are reluctant to, at this stage, remove any participants. If one participant’s data points are removed for one topic, the question arises whether one should remove all of that participant’s data points for all topics, which seems unfair. Instead, we keep all of the data points but treat the ones with high consistency ratios with more scepticism and care, especially the participant who only had an acceptable consistency ratio in two cases.

After studying the values from this participant we concluded that they were not deviating in any extreme way from the other participants’ values, and hence decided to keep them, leaving it up to the subsequent discussion meeting to decide what to do with this type of data points. The participant may still have relevant concerns that should be listened to at the discussion meeting. It may be that a high degree of inconsistency indicates that the architectures or the quality attributes are too vague to be more concrete in the answers.

**6.2 Discussion Meeting**

By filling in the questionnaires, the participants have created the data necessary to identify their respective opinion of the involved architecture candidates. This is used as a basis for discussions, focusing on the points where several participants disagree from the general consensus.

The ultimate purpose of these discussions is to obtain a joint understanding of the strengths and weaknesses of the different architecture candidates, and to decide which architecture is most suited for the problem at hand. However, a major benefit is also that the participants learn from the others’ experiences, thus allowing knowledge transfer to take place. Moreover, by identifying at an early stage where there are disagreements, this enables development projects to plan for further studies of these areas and possibly create some prototypes to get a better view of the situation.

**Preparation.** As input to this phase we take the completed questionnaires, and process the data from these according to the AHP method. This results in a series of nine vectors for each participant, corresponding to the different parts of the questionnaire.

These are then compiled into a presentation, as described in Section 5.4. Discussion points are identified, both those mechanically identified using the calculations in Section 5.4.2 and those points where visual inspections of graphs show that several participants have a differing opinion from the majority.

An example of how discussion points are identified can be found in Figure 1. In this figure, we see the individual answers for the rating of the quality attributes for architecture B. Each participant’s answers are represented as one line in this graph (even though there is no meaningful ordinal scale on the x-axis, we find that it is much easier to find and follow each individual if we use lines rather than mere points).

We see that for performance flexibility the participants are in disagreement. Some rate architecture B highly, others rate it lowly and a few rate it to be moderately good at performance flexibility. This was identified using the formula in Section 5.4.2.
The two smaller circles for functional flexibility and security flexibility indicate points that were identified by visually examining the graph. In these circles we see that a few participants disagree with the majority.

The resulting presentation is organised according to Table 2. The main focus of this presentation is, of course, on topics 3 to 11, i.e. discussions of the individual answers and those answers that differ from the general opinion. In Table 2, the number of identified discussion points for each topic are presented in the comment column, with the number of points that are mechanically identified using the method in Section 5.4.2 within parentheses. This sums up to a total of 20 discussion points, of which 10 are mechanically identified, out of a total of 35 data points (For each participant there is one vector with five values, five vectors with three values and three vectors with five values, and the discussion is about which participants differ from the general consensus).

20 discussion points out of a total of 35 available points may seem a large number. Being unsure ourselves of where to put the threshold values we opted to include more discussion points than may have been necessary, to be on the safe side. To forego events, during the discussion meeting many of the discussion points identified by inspecting the graphs were treated together and others just warranted a quick comment. The 10 mechanically identified points were the source of the most in-depth discussions.

**Execution.** This meeting was held shortly after the summer holidays, because of which we asked the participants to refresh their knowledge of the architecture candidates and quality attributes before the meeting.

The meeting went according to the pre-defined schedule, and lasted for three hours. During this time, each of the identified discussion points were brought into focus and discussed until the different opinions
had been heard (both the opposing views and the majority views). As mentioned earlier, this took longer time for some of the points than for others.

As it may not be uncommon to prioritize quality attributes before architecture decisions are taken, we also provided an opportunity for the participants to change their initial choice of which architecture candidate they preferred after we had presented the prioritized list of quality attributes. Everyone agreed that their initial choice was still the best for the particular blend of quality attributes.

A compilation of the data that was presented during this meeting can be found in Section 7.1.

During the meeting several points were identified where further investigations were necessary before development commenced. This involved both investigations of the impact of some quality aspects as well as further investigations of technical issues that might cause problems during later stages. We also noticed that all of the participants, regardless of seniority and experience, took part in the discussions and shared their previous experiences. Thus the meeting allowed the younger developers to learn from the older and, to some extent, vice versa.

Wrapup Interview. Together with the discussion meeting we also asked a few questions and let the participants “think aloud” concerning how they perceived the method, and what benefits or liabilities they found while using the method.

The general perception was that the method was useful as it helped focus discussions and brought different views into light. Moreover they stated that even if it would have been even more useful with more concrete architecture candidates and quality attributes, it was still useful as the amount of uncertainty reflects the confusion generally present at early stages of a project.

75% of the participants were willing to try the method again in a later stage of the project, when the architecture candidates were clearer and the quality requirements more well defined.

7 Analysis and Interpretation
In this section we describe how we analyse the gathered data with respect to the research questions listed in Section 4.

7.1 Compilation of Data
The individual vectors previously gathered are used as input to the discussion meeting but for the rest of the method, we use the median value of all of the participants, thus forming three tables according to the following:

PQA (Prioritization Quality Attributes). This is a vector of the priorities the participants give to the different quality attributes. This is presented in Table 3 and Figure 2. We see that functional flexibility is the highest prioritized quality attribute (0.324), followed by cost (0.246), testability (0.181), performance flexibility (0.145) and security flexibility (0.104), in that order. We were concerned that the low score for security flexibility was due to the ambiguity of this term in Swedish. However, when questioned the participants agreed that with the interpretation ‘security’ this was the least relevant quality attribute for this particular study and the specific subsystem.

FAS (Framework for Architecture Structures). This is a ranking of the quality attributes for each architecture candidate. This is presented in Table 4 and in Figure 3. In this figure, each column of dots represents one architecture, and each dot represents the value for one particular quality attribute. Note that it is impossible to compare the actual values between different architectures, but only the position relative to the other quality attributes for that particular architecture. We see, for example, that all architectures seem to be good at functional flexibility, and that architecture B (the middle column) is specialised on functional flexibility and performance flexibility, but is not at all very good at the cost attribute.

FQA (Framework for Quality Attributes). This is a ranking of the architecture candidates for each qual-
ity attribute. This is presented in Table 5 and in Figure 4. Here, we see that architecture B is, except for testability, the top rated architecture for all quality attributes. Moreover, we see that architecture A is usually ranked lowest, except for security flexibility, where it is second, and performance flexibility, where it is equal to architecture C.

With these as input we first create a refined FQA (simply referred to as FQAr), using the values from the FAS. It is outside the scope of this paper to describe exactly how this is done, but a full description of how to do it and the motivation for doing it can be found in [Svalnberg et al. 2002].

We then use the PQA to get a value of the match between the required quality attributes and the support given for these by the different architecture candidates. This is done using the following formula:

$$\sum_{i=1}^{k} PQA_j FQAr_{j,i}$$

where $k$ is the number of quality attributes and $i$ is an index referring to each of the architecture candidates in turn.

The result from this is presented in Table 6. In this table, we see that architecture B is the architecture that our method recommends, as it has the highest number. Architecture C comes second, followed by architecture A.

### 7.2 Analysis for Q1

Q1 concerns whether the use of a structured method strengthens the confidence that the right decision is taken with respect to which architecture candidate to use. To perform this evaluation, we must first ensure ourselves that the structured method used produces relevant results.

Based on nothing but their experience, all of the participants preferred architecture B, which is also the same as the structured decision support method recommends.

After being presented with a prioritized list of quality attributes, the participants still preferred architecture B.

This answers the first part of the evaluation for Q1: the method does indeed produce relevant results, according to our measurement.

For the second part of the evaluation, spontaneous comments during the wrapup interview reveal that the discussion meeting, because it brought several views to the surface, indeed strengthen the confidence in that the right decision is taken.

Hence, we can conclude for the second part of this question: there is an increase in confidence when using the structured method for comparing architecture candidates, compared to not having a structured method including discussions.

Of course, this would probably be true for any discussion held about a set of architecture candidates, regardless of whether a structured method is used to prepare before the discussion or not. However, there was also a general agreement that the method used helped in pinpointing in an effective way where discussions are more necessary. A more focused discussion increases the confidence that the right things are discussed, which also in turn increases the confidence that the right decision is taken. However, we have no proof of this, as it would require a control group where unstructured discussions are held and perhaps a rating of the confidence in the decision.
For example, one can expect that people with a low ability to judge the qualities of software architectures. People from different positions may have different expertise of software architectures. In the study in this paper, we use people with different positions in the company, ranging from managers to developers. In the first study, we used only people with high expertise of software architectures. In the study in 1996, we presented the participants with a total of six pages describing the architecture context and the candidate architectures.

As mentioned in Section 5.2, we use the squared distance to the mean to measure the amount of disagreement each participant has in every data point. As the number of participants differ, we cannot simply use the sum of these values, as we have done in other situations. Instead, we use the mean value of all participants.

The mean values for each of the two studies are presented in Table 7. In this table we see that when the target system is known, the mean value of the disagreement among the participants is 0.0103 for the ranking of quality attributes within each architecture (FAS) and 0.0276 for the ranking of architectures for each quality attribute (FQA).

Surprisingly, these values are actually lower when the target system is unknown. This means that there is more consensus in the first study than there is in this study. A number of explanations for this can be found.

For example, in the first study the consistency ratios were very low with no values larger than 0.13, whereas in this study 23 of 117 (20%) vectors have values larger than 0.15, sometimes considerably larger. Unfortunately, removing vectors with consistency indexes larger than 0.15 only has a marginal effect on the mean values of the squared distances to the mean.

Another factor is the expertise of the participants. In the first study, we used only people with high expertise of software architectures. In the study in this paper, we use people with different positions in the company, ranging from managers to developers. People from different positions may have different ability to judge the qualities of software architectures. For example, one can expect that people with a low ability to judge software architectures give answers that are more or less random, which of course increase the spread among the answers. This would also be visible in each participant’s consistency indexes produced by AHP. However, we can see nothing out of the ordinary when examining these numbers.

A third factor may be that our assumption that a high level description of the architectures is sufficient may be false. In the previous study, the participants were given 100 pages of descriptions (text, images and examples) of the architecture candidates, whereas in this study we presented the participants with a total of six pages describing the architecture context and the candidate architectures.

Table 7: Measurement of Disagreement with Known and Unknown Target System

<table>
<thead>
<tr>
<th></th>
<th>FAS</th>
<th>FQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known Target System</td>
<td>0.0103</td>
<td>0.0276</td>
</tr>
<tr>
<td>Unknown Target System</td>
<td>0.00877</td>
<td>0.0123</td>
</tr>
</tbody>
</table>

In any case we cannot confirm our theory that there will be more consensus among the answers when the target system is known. However, we find several issues for future research here. First, we would like to see whether there are different groups of participants (e.g. management, marketing and development) that answer similarly as a group, but where the groups disagree. As mentioned above, this would be one reason for the spread among the answers. Moreover, this would indicate that the communication between these groups in the company is deficient and could be improved upon.

Second, we would like to study how much influence the level of detail and the amount of information given for each of the architecture candidates have on the spread of the answers when the target system is known.

Third, an experiment should be conducted where exactly the same architectures are evaluated by two groups of people where one is aware of the target system and the other is not.

8 Summary and Conclusions

In this paper we describe a study of a company in the process of determining which software architecture to use in a system they are currently designing. The company uses a method that measures peoples’ view of the benefits and liabilities of different software architecture candidates in a quantified way. This is then used to identify which architecture best match the quality requirements for a system to build.

The method pinpoints where people disagree so that structured discussions can be held. The objective of these discussions is to increase the understanding of the architecture candidates and indeed the software system in question. Two major benefits are that these discussions enable the software designers to pinpoint where further investigations are needed and to generally increase the confidence that the right architecture decisions are taken. Another benefit is that the participants learn from each other.

Based on data from this study, we draw the following conclusions regarding the research questions:

Confidence in Decisions (Q1). The spontaneous reactions of the participants seem to support our
hypothesis that not only does the discussions help, our method also shows what different opinions to look for and where to look for differing opinions. The resulting discussion is hence more focused and thus increases the confidence that the right decision is taken, compared to if an unstructured discussion is held.

**Less Spread among Answers (Q2).** This was our most surprising result as our analysis revealed that there were more consensus among the participants in a previous experiment where the target system was unknown, as opposed to this study where the target system was known and familiar to the participants. There are, however, several factors that need to be investigated further before anything conclusive can be said about this result.

### 8.1 Future Work

In addition to the areas of future work that we outline in Section 7.3, we also see the following tracks for future research:

**Replication.** We now have empirical evidence from two cases, one from academia [Svahnberg & Wohlin 2002a][Svahnberg & Wohlin 2002b] and one from industry in applying our method for quantifiably discerning between architecture candidates with respect to quality attributes. The study provides some initial results and points to a potential way forward to informed decisions regarding architecture selection. Nevertheless we encourage readers to replicate this study in order to increase the experience base.

**Student Experiment.** We intend to conduct an experiment where a number of master year students evaluate the same architecture candidates and quality attributes as we use in this study (although with a more thorough introduction to the company and the problem domain). By using students, we capture the limited experience and knowledge that a newly hired employee can be expected to have. This experiment would serve several purposes:

- It would test if the method is a useful learning tool for new employees in a company. A discussion meeting with a mixture of experienced people and newly graduated people is likely to enable a knowledge transfer, in both directions.
- It would help in providing an answer to whether domain knowledge is necessary to compare architectures or whether this can be done for generic architecture styles as well.
- If repeated with the same students within reasonable time, it would help investigate the degree of consistency in the answers to questionnaires of this sort by measuring the degree of difference between the first and the second time the questionnaire is completed.

### 8.2 Acknowledgements

We would like to thank Danaher Motion Särö AB for their help and participation in this study.

**References**


Appendix A. Analytic Hierarchy Process

Simply put, the Analytic Hierarchy Process (AHP) is a process for making pairwise comparisons. By making all possible pairwise comparisons it is possible to make a ranking of the entities and to, in addition, calculate a consistency index. AHP consists of four basic substeps and an added fifth substep to analyse the consistency in the results, described below.

Substep 1: Create an \( n \times n \) matrix (denoted \( N \)), where \( n \) is the variables to be compared, for example, quality attributes. In the diagonal in the matrix the value “1” is inserted. The matrix is referred to as the comparison matrix. Element \( n_{ij} \), when \( i \) is not equal to \( j \), records the relative importance of variable \( i \) versus variable \( j \).

Substep 2: Perform a pairwise comparison of the variables with respect to the importance. The scale for comparing the variables pairwise is illustrated in Figure 5. Each pairwise comparison means that it is necessary to determine which of two variables is most important and how much more important it is. For example, a marking to the left on the scale means that variable \( i \) is more important than variable \( j \). The interpretation of the values is shown in Table 8.

The pairwise comparison is conducted for all pairs, i.e. \( n \times (n-1)/2 \) comparisons. The relative importance values are put into the matrix created in the first step and the reciprocal for each pair is determined and also put into the matrix. This results in a complete \( n \times n \) matrix.

Substep 3: Compute the eigenvector of the \( n \times n \) matrix. A simple method is proposed by [Saaty 1980][Saaty & Vargas 2001] to do this; the method is known as averaging over normalised columns, and the procedure is as follows:

- Calculate the sum of the columns in the matrix, \( n \times n \). More formally if

\[
\sum_{i=1}^{n} n_{ij} = \sum_{i=1}^{n} n_{ij} = 1
\]

- Each element in a column is divided by the sum of the column, \( m_{ij} = n_{ij}/n_j \). This results in a new matrix, denoted \( M \), with elements \( m_{ij} \).

<table>
<thead>
<tr>
<th>Relative intensity</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Of equal importance</td>
<td>The two variables ((i \text{ and } j)) are of equal importance.</td>
</tr>
<tr>
<td>3</td>
<td>Slightly more important</td>
<td>One variable is slightly more important than the other.</td>
</tr>
<tr>
<td>5</td>
<td>Highly more important</td>
<td>One variable is highly more important than the other.</td>
</tr>
<tr>
<td>7</td>
<td>Very highly more important</td>
<td>One variable is very highly more important than the other.</td>
</tr>
<tr>
<td>9</td>
<td>Extremely more important</td>
<td>One variable is extremely more important than the other.</td>
</tr>
<tr>
<td>2, 4, 6, 8</td>
<td>Intermediate values</td>
<td>Used when compromising between the other numbers.</td>
</tr>
<tr>
<td>Reciprocal</td>
<td></td>
<td>If variable ( i ) has one of the above numbers assigned to it when compared with variable ( j ), then ( j ) has the value ( 1/ ) number assigned to it when compared with ( i ). More formally if ( n_{ij} = x ) then ( n_{ji} = 1/x ).</td>
</tr>
</tbody>
</table>

Figure 5. The scale for the AHP comparison.
• The sum of each row in the new matrix is calculated, \( m_i = \sum_{j=1}^{n} m_{ij} \).

• The sums of the rows are normalised by dividing by the number of variables \( n \), \( P_i = m_i/n \).

This results in an estimation of the eigen values of the matrix, and it is referred to as the priority vector. The vector is denoted \( P \) with elements \( P_i \) for \( i = 1 \ldots n \).

**Substep 4:** Assign a relative importance to the variables. The first variable is assigned the element in the priority vector. It is said that the first variable accounts for \( P_1 \) percent of the importance of the variables. The second variable is assigned the second element in the priority vector and so on. Let \( P_1 \) to \( P_n \) be the percentage values for the importance of variables 1 to \( n \).

**Substep 5:** Because AHP conducts more comparisons than minimally necessary, it is possible to evaluate the consistency of the ranking. This consistency ratio captures how consistently the pairwise comparison has been conducted. The consistency check is particularly important when a large number of pairwise comparisons are necessary, making it easier to make errors and hence become inconsistent.

The consistency ratio is computed in two steps.

• First, a consistency index (CI) is computed as \( CI = (\lambda_{max} - n)/(n - 1) \), where \( \lambda_{max} \) is the maximum principal eigen value of the \( n \times n \) matrix. The closer \( \lambda_{max} \) is to \( n \) the smaller is the error in the comparison. \( \lambda_{max} \) is calculated by first multiplying the comparison matrix, i.e. matrix \( N \), with the priority vector. Let the resulting vector be denoted \( R \) with elements \( R_i \), \( R = N \times P \). For the resulting vector, each element in the vector is divided by the corresponding element in the priority vector, \( \lambda_i = R_i/P_i \).

\( \lambda_{max} \) is now computed as the average of the elements in the resulting vector,

\[
\lambda_{max} = \left( \frac{1}{n} \sum_{i=1}^{n} \lambda_i \right) / n.
\]

CI can now be calculated.

• The consistency ratio (CR) is determined by dividing the consistency index (CI) by a random index (RI). The random index has been generated to take into account randomness and it is used to normalise the consistency index. Thus, \( CR = (CI)/(RI) \), where RI is determined from Table 9, where the first row shows the order of the matrix \( n \) and the second row shows the corresponding RI value. The smaller CR, the more consistent is the comparison.

According to [Saaty 1980][Saaty & Vargas 2001], a consistency ratio of 0.10 or less is considered acceptable. It is, however, pointed out in the literature that in practice higher values are often obtained. This indicates that 0.10 may be too hard, but it indicates an approximate size of the expected consistency ratio.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.58</td>
<td>0.90</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
<td>1.45</td>
<td>1.45</td>
<td>1.51</td>
<td>1.48</td>
<td>1.56</td>
<td>1.57</td>
<td>1.59</td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Matrix order and corresponding RI value [Saaty 1980][Saaty & Vargas 2001].