Experiences in System Architecture Evaluation: A Communication View for Architectural Design

Jari A. Lehto\(^1\) and Pentti Marttiin\(^1,2\)

\(^1\)Nokia Technology Platforms, \(^2\)Helsinki School of Economics / Information Systems Science

Jari.A.Lehto@nokia.com, Pentti.Marttiin@nokia.com

Abstract

Successful system architecture design requires an understanding of the needs and requirements of a wide range of stakeholders covering the complete system architecture-related life cycle. Communicating with various stakeholders is a highly delicate task and may cause misunderstandings during the translation of functional needs into architectural properties. We developed the Architecture Evaluation Framework (AEF) to improve the design of telecommunication network elements architecture. The various uses of AEF include checking the appropriateness of architecture in relation to business drivers, suitability of external components (COTS, open source), evaluating internal coherence of a product family, or prioritizing development efforts. Furthermore, the AEF is scalable in terms of process phases and framework comprehensiveness.

In this paper, we discuss the creation and use of the AEF. The important architectural issues gathered from product life cycle stakeholders form a framework, a hierarchy of architectural factors. We use a bottom-up approach in building the hierarchy. Thus, the selected factors reflect the terminology used among the stakeholders. Based on our experiences, we claim that such a framework is easy to learn and helps to communicate architectural preferences. These benefits are achieved by using the internal origin, familiar factors, and embedded support for all definitions used in the factor hierarchy.

1. Introduction

There are several ways to improve the product creation process. We have chosen to focus on the system (or product) architecture for the following reasons. Firstly, system architecture is a powerful means to combine the diversified needs of several stakeholders. The importance of stakeholder views in designing architecture has been addressed in systems engineering [1] and software engineering [2, 3] literature. For example, Katzman et al. [4] have developed an analysis method to cover stakeholders views. Secondly, system architecture is highly abstract by nature. Factors affecting architecture are often vague and hard to express. Furthermore, the processing of architectural factors remains informal and, as we have noticed, many assumptions are hidden. Therefore, an architecture that describes the high-level structures and solutions of the system serves as an important communication tool between different stakeholders. The communication aspect has been emphasized in literature: architecture has been regarded ‘a mutual reality’ [5] and a shared mental model [6]. Thirdly, we have noticed that choices behind the system architecture are seldom explained well. It is hard to find an explicit rationale for the design decisions that have led to the current implementation of the architecture. In large organizations, an explicit consensus may never be reached and commitments before actions taken remain unclear. In such a case, decisions can be regarded as artificial constructs without rationale [7], and it is hard to standardize architectural choices.

There are several definitions for architecture [2, 8, 9, 10, 11]. Instead of creating a new definition, we will characterize our understanding of architecture. We will focus on the evaluation of the properties that affect the system in question during the whole product life cycle. It is often useful to treat the creation of architecture as a domain-definition problem. The set of problems is divided to concentrate on the critical efforts and to exclude the less-critical ones [12]. Such a division is implemented by determining the relevant set of stakeholders and their relationships with the product. We define the architecture role as ‘reflecting the art of order’. Architecture guides our effort when creating order in the world of ever-lasting diffusion. Stakeholders are used to giving reasons for ordering via expressing their needs, constraints, policies, rules of behaviour, and other means of describing a state of affairs they hope will come into existence.

This paper presents our experiences in product architecture evaluation carried out during 2002 and 2003. We collected architecture-related issues from different stakeholders and constructed an evaluation framework (Architecture Evaluation Framework, AEF) to cover the whole life cycle of the product. The framework was created for the evaluation of telecommunication products and it has been piloted at Nokia Networks in a project which focused on a network element. The lead author facilitated the evaluation that was carried out by the experts of the selected product.

The paper is organized as follows. In Section 2, we review a set of architecture evaluation methods and evaluate their suitability for our purposes. Section 3 summarizes the results of the workshops we arranged for
telecommunication experts in order to widely discuss product creation and architecture challenges. The workshops generated a set of drivers that assisted in the creation of the AEF. In Section 4, we discuss the creation of the AEF and describe the resulting AEF method. Section 5 focuses on the use of the AEF in a pilot project. We summarize our lessons learned during both the creation and use of the AEF in Section 6, and draw conclusions in Section 7.

2. Architecture evaluation methods

In this section, we present a review of five architecture evaluation methods accompanied by a short analysis of their suitability for our case.

The Architecture Trade-off Analysis Method (ATAM) [13] is developed for the architecture of complex software intensive systems and used as a risk identification method. The ATAM aims at raising architectural awareness and it improves the quality of architecture documentation. The method determines and generates sensitivity points that are measurable quality attributes in the architecture. The ATAM requires a statement of quality attributes and a clear specification of architectural decisions. The result is organized as a quality attribute tree (or hierarchy). The ATAM is composed of the following tasks: the definition of business drivers, the definition of architectural quality factors, and the generation of the quality tree and results. Users are guided to perform these analysis tasks in a detailed manner.

The Software Architecture Analysis Method (SAAM) [14] suggests that the architecture design of the software system is composed of three elements: function, structure, and the allocation of functions into the structure. SAAM’s scope has been broadened since this original publication to encompass user interfaces [15]. Methodically, the SAAM has also been extended towards scenario-based identification of quality factors [4]. Furthermore, Lassing [16] has extended the SAAM to uncover architectural inflexibility. The SAAM is used for software architecture analysis.

The Architecture Level Modifiability Analysis (ALMA) [17] focuses on analysing the modifiability of architecture. The ALMA is a similar scenario–based method as the SAAM and the ATAM. It describes architecture as a part of the process.

The System Engineering Process Activities (SEPA) [18] deals with requirements and architecture. The SEPA is a domain reference architecture and it separates and represents requirement types among a set of interrelated types of architecture. It makes architecture derivation and evaluation an integral part of product creation. The SEPA aims at measuring the performance of the resulting implementation in the early phases of architecture design.

Finally, the objective of the ISO/IEC 9126 [19] is to provide a framework for the evaluation of software quality. The standard defines a quality model which is applicable to various kinds of software. It defines six product quality characteristics and provides a suggestion of quality sub-characteristics.

All the methods discussed above were benchmarked prior to our final decision to develop an in-house evaluation framework. The decision was based mainly on two arguments. Firstly, the scope of concerns and related terminology should be tied to the those of the company. We noticed that the use of external frameworks would bring on the additional step of learning the meanings of factors, thus, forcing us to use a separate language. Secondly, the help and availability of an external consultant made the decision to start the AEF development easier. The AEF is based predominantly on the ATAM and ISO/IEC 9126, while the scope of the other three methods was different.

The objective of the ATAM is close to ours which is to raise the awareness of architectural trade-offs. However, after benchmarking the available methods, system architects stressed the need for a pragmatic and simple method used in a busy business situation. This need was still present after the AEF pilots and it has lead us to diversify the framework in depth (to use either 20, 70, or 120 factors in analysis).

As a quality factor framework, the ISO/IEC 9126 relates with our hierarchy. It has introduced us to formidable guidelines in creating sub-hierarchies in a common manner. However, the ISO/IEC 9126 presents an external framework which we purposefully have tried to avoid as discussed earlier.

The SAAM focuses on the software architecture aspects and, as such, it provides too narrow scope for our purposes to evaluate various types of system level architecture (e.g. telecom networks and network elements). Similarly, we aim at a more general level architecture evaluation than the one provided by the ALMA. The ALMA requires an architectural description as an input document while we are dealing with architectural factors. Besides, the ALMA focuses on the architectural modifiability, which is only a part of the problem that needs attention. The SEPA provides a full methodology and has a wider scope than in our trade-off analysis. However, the SEPA may provide an important target for our future studies.

3. Motivation

3.1. Challenges

Business needs in the telecommunications industry and in our company require product development cycles to be shortened. In order to successfully respond to this
challenge, the producer (of telecom products) has to understand the stakeholder needs encompassing the entire product life cycle. The producer must also be able to transform these needs as properties of product architecture. In order to improve the product creation effectiveness and efficiency, architectural factors should be identified, evaluated, balanced and made visible for all architects and developers [20].

This study originates from the belief that we can improve the product affordability by affecting its architecture. In AEF workshops we identified the following set of challenges as necessary to overcome in order to attain this affordability.

- Developing telecommunication equipment for future usage during the defined period of time. This challenge requires us to design a flexible, modifiable and scalable product for future use.
- Controlling affordability during the entire life cycle of the system and its subsystems. This challenge requires us to define the future customer expectations and design the feasible value chains for the telecom producer, operators and other stakeholders.
- Ensuring the maintainability during the product life cycle. This challenge requires us to design an easy to maintain product with minimal cost to the customer.
- Adapting the equipment to its future environment originated change requests. Both the technology standards and the way of using of equipments change over time. It is important to be aware of the kinds of presuppositions that have been made and observe whether they hold or cause changes in the product. Early recognition of changes is important.
- The use of third-party components with different life cycles. This challenge requires a continuous follow-up of third party components and preparative actions to improve or replace them occasionally.

The Architecture Evaluation Framework (AEF) is created to respond to the challenges listed above. However, we are still gathering the AEF experiences in order to learn how to respond to these challenges. Thus, the pilot results presented later do not give final answers nor show the ultimate impact of coping with these challenges.

3.2. Targets

One of the most important drivers for us has been to create a way to understand the potential embedded in an explicit description of the product architecture. In this study, the explicit description of the architecture refers to the explicit hierarchy of factors based on stakeholder requirements. If the factor hierarchy is supported by a tool, these factors could be easily browsed through and the need for a new factor or any other update for the hierarchy checked. The description also gives answers for describing how a factor is considered in respect to other factors.

Our second goal was to create visibility to most of the factors that have a clear impact on the product architecture. The visibility of the factor hierarchy shows us the position of a factor in a hierarchy; its relation to the upper level factors and the possible division into parts at the lower level. Standards (such as ISO [19]) exist that are created for this purpose. However, as we discussed before, our primary aim has been to create a hierarchy that is a reflection of our company knowledge and preferences. The alignment of the AEF with the standards is in our future plans when the architects are more aware of the common terminology.

Our third goal was to support communication and organizational memory of the architecture and related decisions. The fact that the architecture can be shown as a set of factors gives us the basic support and ability to communicate about the architecture. The hierarchy of architectural factors also supports agreement about the terms and their meanings.

Architecture design is about making compromises. Because of the way the hierarchy is built all the factors having the same parent should represent a factor of their own. This results in visible design trade-offs and supports the ability to understand architectural trade-offs and to recognize the impact of the priorities of stakeholders.

Our final goal was to determine a rationale for making design decisions. We use the analytic hierarchy process (AHP) [21] and ExpertChoice™ tool to evaluate all the factors. Here, the result itself is more important than the processing of the result. We can evaluate the framework when necessary and check whether our opinions about the business situation have changed. The previous status is always available for discussion. Previously, this kind of information appeared only in oral form without centralized documentation.

4. The Architecture Evaluation Framework (AEF)

4.1. The Creation of the Framework

The AEF was developed in Autumn 2002 in co-operation with Nokia Networks, Nokia Research Center and an external consultant from Stevens Institute of Technology who represented the original concept for the AEF. The AEF development team (two facilitators and 30 experts) collected and produced business-driven stakeholder requirements and architectural design issues. The team developed the AEF for the telecom domain and focused on base station type of network elements. The experience that was gathered during the evaluation pilot was utilized later for further development of the AEF (see Section 5).
Table 1. The AEF development

<table>
<thead>
<tr>
<th>Period</th>
<th>June-August 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Development of the evaluation framework for network elements, specifically telecom base stations.</td>
</tr>
<tr>
<td>Participants</td>
<td>30 experts from the business unit. Roles covered the product life cycle (marketing, development, maintenance)</td>
</tr>
<tr>
<td>Facilitation</td>
<td>One external consultant for facilitating the framework development. One facilitator from Nokia for facilitating the pilots.</td>
</tr>
</tbody>
</table>

Way of working and time spent:
- A kick-off meeting with 30 experts (200 person hours)
- A series of workshops with approximately 15 persons (600 person hours)
- Individual work:
  - Consultant 100 person hours
  - Facilitator 100 person hours
  - Experts 500 person hours

Implementing a new and sophisticated method or process is a hard task for a large organisation with a history of local working practices. This was the main reason we decided not to start with deploying any of the existing frameworks discussed in Section 2. Instead, we collected several architecture specialists and representatives into a series of workshops. We discussed evaluation principles, gathered a set of architectural issues covering the whole product life cycle and selected the architecturally important ones. Each matter was discussed independently of the architecture of a certain product.

Our next step was to evaluate the selected issues from the viewpoints of their representatives. Some parallel discussions were carried out to understand the relationships and dependencies of the issues expressed. Results were collected and organized into a hierarchy of architectural factors. This was done with the help of our external consultant. We used the AHP method [21] and the Expert Choice tool for this work.

When creating the factor hierarchy and deciding upon the upper-level structure, we utilized our existing understanding of architecture factor hierarchies. In addition, we used a “bottom-up” tactic to formulate the lowest level factors in the hierarchy. These were formulated into concrete properties of the hierarchy. The result was a hierarchy of factors (or ‘ilities’ – the term we used during the process). The meaning of a node in a hierarchy can be derived from the child factors. Thus, the hierarchy supports apprehension of the hierarchy. The example is seen in Figure 1.

![Fig 1. The top-level view of the Architecture Evaluation Framework](attachment:image.png)
We create the value background and give weights for each individual factors (pair-wise comparison). The weighted hierarchy is used to compare a set of products (or e.g. product blueprints). Each product is evaluated according to the factors and the result shows their relative match with the value background. The next section describes the evaluation activities at a more detailed level.

4.2. The AEF Method as a Process

This section presents the AEF method as a process containing four activities: defining business drivers, setting-up the framework, adjusting the domain and evaluating the product(s). Next, we present these activities each divided into two steps. The actual process is composed in the form of evaluation workshops.

4.2.1. Defining Business Drivers. Before making any evaluations, the goal of the evaluation must be clarified. The AEF method helps in defining the target of evaluation and in determining business drivers for the relevant period of time. This activity includes two steps: selecting representatives and determining business drivers.

Step 1: Selecting representatives: One of the most important objectives for architecture evaluation is guaranteeing the whole life cycle of the product is covered and that all needs and requirements are balanced. Therefore, it is necessary to establish an evaluation team with representatives from the whole life cycle and relevant business areas.

Step 2: Determining business drivers: One major problem when making a multi-factor evaluation is the elimination of evaluators who change their points of view during the evaluation process. In this step, we create the context for the evaluation by agreeing on a domain definition (the topic), business drivers (the planned effect on the markets), and a time perspective (the duration of the effect on the markets pertaining to the product). In addition to the agreed context, pair-wise comparison (Step 5) helps evaluators maintain their points of view.

4.2.2. Setting-up the Framework. This activity includes two steps: “Collecting data” and “Rationalizing the hierarchy of factors according to domain”.

Step 3: Collecting data: In this step, we align the framework to include relevant factors. We assume that an existing framework exists, but we need to align it to the specific domain of concern. Our framework will be scalable to three levels of granularity (including 20, 70 or 120 factors). At the top-level, the framework is most compatible to road mapping and development programs. At the detailed level (120 factors), we are dealing with domain specific factors, e.g. those related to base stations. In this step, we recommend that the evaluation team goes through the existing framework and determines potential changes to be discussed and agreed upon during the next step. This may include, e.g. new candidates for leaf level factors. Furthermore, this step helps to get a common understanding of what each factor means. Without this information, the process cannot be continued successfully.

Step 4. Rationalizing the hierarchy of factors according to the domain. We first created a candidate framework and then selected two pilot cases for testing it (one of them presented in this study). Soon, we found dependencies between the framework, the target domain, and the product to be evaluated. This caused the following consequences. If we strive for very exact measures, we have to create a hierarchy of several layers. The top-level is a generic one (e.g., operational efficiency) that is split into domain specific factors (e.g., a set of operational efficiency measures of the target product). At the leaf-level, the domain specific factors can be used as metrics.

The current AEF, an example of which is presented in Figure 1, is composed of five top-level factors of architecture. Each of them holds one or more sub-level factors and the detailed leaf-level metrics take the form of a question.

4.2.3. Adjusting the Domain. This activity includes two steps: Making the domain specific evaluation base line and defining the metrics.

Step 5: Making the domain specific evaluation base line (Pair-wise comparison): The pair-wise comparison and sensitivity analysis are enabled by the Analytic Hierarchy Process (AHP) technique [21] and the ExpertChoice™ tool [13]. The AHP is a decision-making technique that utilizes pair-wise comparisons in order to balance multiple objectives, criteria, or alternatives. These are represented in a hierarchical breakdown structure. ExpertChoice™ is equipped with an inconsistency indicator to keep track of the accuracy between estimations.

The candidate framework (fine-tuned for a specific domain in Step 4) shall be adjusted for the evaluation by giving values to describe the importance of factors. Firstly, pair-wise (or reciprocal) comparison is to be conducted. Evaluators go through the entire hierarchy and compare each choice against others within a branch (i.e., reciprocal comparison). The facilitator asks two questions for each pair of factors during the comparison phase: Which one of the two factors is more important in regard to the business drivers? To what degree is the one that is selected more important?

Evaluators assess the importance of the factors according to business drivers and the valid time perspective. Different business drivers may drive
architecture towards a special value base. This task is hard and time consuming, but it forms the foundation for the value background of all the factors related to the product architecture.

Secondly, before using the value background for comparing products the correctness of the scales of alternate answers should be checked. In our candidate framework linear scaling was used and preferred. In order to use another scales (e.g., logarithmical, yes/no) an adjustment shall be conducted.

Step 6: Defining the metrics: Leaves in the hierarchy form the metrics and they are presented as questions. Example questions include: What is the system level availability? and What is the mean time between service interruptions? In the case of any added or removed questions or changes to the contents of questions we need to check the scale for the metrics. The scale includes a selection of ready-made answers. There are several possible scaling functions. In the AEF method we have suggested using linear scales.

4.2.4. Evaluating the product(s). This activity includes two steps: running comparison and sensitivity analysis.

Step 7: Running comparison: The comparison can be regarded as multi-factor decision-making. The comparison process is a rather straightforward process if one has several products to be compared. When evaluating only one product we need to create an ideal reference product to be used as a counterpart for the comparison. The process is as follows: the facilitator asks leaf-level questions and the evaluators give answers with a certain product in mind. This will be repeated for each product compared. The result is a detailed profile of how the product responds to the importance of factors.

Step 8: Running sensitivity-analysis: The main goal of the analysis is to ensure that the result can be relied upon. After this has been done, one can test how the product score changes when changing the value background. The first issue is to control the inconsistency indicator provided by Expert Choice™ by keeping it low. In our case less, than 0,1 was satisfactory. The second issue to be confirmed is the importance of top-level factors following business drivers agreed upon during Step 2. Business drivers usually have a different structure in contrast to the factor structure. One factor may include several business drivers and vice versa. When the importance of factors follows business drivers, then the result obviously can be trusted. In other cases, one must take care of drawing conclusions from the comparison results.

Finally, we go through the whole hierarchy and check that all sublevels correspond with what has been discussed. We suggest that the discussions during business driver determination (Step 2) and pair-wise comparison (Step 5) are recorded. This information will give the background for understanding the results of the framework.

Sensitivity analysis allows us to change the importance of one individual factor and to perceive the effect on other factors of the evaluated products. The dynamic graph was preferred for sensitivity analysis. It is used to change the importance of one or several factors. The following three effects can be recognized: (1) the relative quality of the product increases, (2) the relative quality of the product decreases, or (3) the relative quality of the product does not change. The general interpretations can be:

- The product is good in that area and new investments may not provide much added value.
- The product can be improved in that area and there is a possibility that the improvements will provide added value compared to other products.
- Any improvement effort may not have an effect on the product in the extent of the framework factors. (This is an obstacle and is not worth considering as a real choice.)

5. Evaluation case

5.1. Case environment

The selected pilot was a Base Station platform architecture that has been developed during the years from 1999 to 2002 when anticipations of the third generation mobile networks market and services development and IP-based network evolution had a lot of attention in the industry. Since then, the scope of BS platform-based product programs has been redirected a few times. Therefore, two business areas and program management felt it was essential to verify the validity of the business driven assumptions that were set for BS and the compliance of the technical solution.

Architecture evaluation took place during the Spring of 2003 in several workshop sessions. These included sessions for business driven goal setting, pair-wise comparison and actual evaluation of the BS compliance with the attributes. These groups participated in the sessions: representatives from Technology Modules and Radio Networks business, program and product management, the FBS architecture team and the evaluation team as facilitators.

During the Summer of 2003, the evaluation team analyzed the outcome of the recorded notes from the workshop discussions and the framework results. The conclusions were classified into three categories according to the expected importance. These include recommendations as requirement statements on strategic matters, on requirements and feature related issues, and on architecture and design related issues.
Results were handled in feedback sessions including the Technology Modules and Radio Networks businesses as well as the program management and architecture team for further prioritization and consequent concrete actions. Table 2 summarizes both the case project and the AEF evaluation pilot characteristics.

Table 2. The AEF pilot: case project and evaluation characteristics

<table>
<thead>
<tr>
<th>Case project</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Creation of a next generation base station</td>
</tr>
<tr>
<td>Projecting</td>
<td>Over 100 persons</td>
</tr>
<tr>
<td>Project phase</td>
<td>In the beginning of its life cycle</td>
</tr>
<tr>
<td>Maturity of used technology</td>
<td>New technology</td>
</tr>
<tr>
<td>Market</td>
<td>International</td>
</tr>
<tr>
<td>AEF Evaluation</td>
<td>May 2003</td>
</tr>
<tr>
<td>Period</td>
<td>To evaluate the architecture of a network element</td>
</tr>
<tr>
<td>Goal</td>
<td>Four persons from the business unit (marketing, development, maintenance), a facilitator and a recorder</td>
</tr>
<tr>
<td>Participants</td>
<td>A series of face-to-face meetings. Four meetings for the evaluation baseline and feedback. Three meetings for the product evaluation. Total time spent: 350 person hours: - Experts 150 person hours - Consultant 100 person hours - Facilitator 100 person hours</td>
</tr>
</tbody>
</table>

5.2. Results

The most notable result of the evaluation was to change participants’ “way of thinking” about architecture. The framework made architecture concrete and visible to support the discussion. The participating experts succeeded in understanding three subjects: the desired architecture, the actual architecture and explanations on what may cause the difference. We tested the evaluation method itself and collected improvements and needs. We gathered experience in arranging multifactor evaluations and learned of their difficulty. This has forced us to change the AEF method to contain also a quicker version. Currently, we provide a scalable AEF including lighter versions (checklists, 20 factors, 70 factors) from the comprehensive one with 120 factors we used in this pilot.

5.3. Technical influence

The most important technical result was to get an understanding of the status of our blueprint architecture. The pilot project described an ideal architecture, which was the ultimate goal without any technical constraints. The pilot used the ideal one as a counterpart in the comparison when evaluating the blueprint architecture of the base station.

The result was visualized in a graph with two curves where the gap shows where our blueprint architecture didn’t reach the ideal architecture. The real benefit was when we used sensitivity-analysis to see if we could improve the blueprint architecture. The sensitivity-analysis helped us to focus our efforts on the most effective targets. The framework identified nine main architectural factors and in five of them, we noticed our strengths. In four of the factors, which were analyzed in more detail, we could do improvements. In one of the factors, the effect of improvement remain unclear. These were strictly product related technical improvements, but in addition to that we created 40 improvement proposals mainly concerning the process and methods used in it.

5.4. Business influence

The current business drivers were defined in the beginning of the evaluation process and were kept visible the entire time. This helped the evaluation process, especially when the evaluators needed to answer technical and detailed questions. The awareness of business drivers showed evaluators that the business situation has an impact on the architecture. This leads to practices where the evaluation shall be re-run periodically. If the business drivers change, a new evaluation shall also be conducted.

The range of participating stakeholders in the framework creation was larger than normally used in architecture design. Several single cases of value setting (Step 4) and pair-wise comparison (Step 5) showed us the limited knowledge of stakeholder requirements. As a consequence, several improvement proposals to the requirements elicitation process were made.

Despite of the many new ideas and increased understanding, evaluators regarded the result as expected. We regard such an experience as an obstacle to both individual learning and process improvements.

6. Lessons learned

6.1. Technological perspective

The target of the evaluation was the architecture blueprint of a base station. The evaluation brought in several improvement proposals and the architects learned how to prioritize them based on their effect on the product. This kind of reasoning was supported with fact-based
argumentation. If the business drivers are changing, the facts can be checked and prioritizations changed when necessary.

A year after the AEF pilot, we asked whether there were any prominent impacts. The main changes are seen in competences and the ways of thinking about architecture. Improvements are carried out in the architectural design: interface specifications and encapsulation of software (layering of base-band software as an example) were emphasized and a new concept of design block development was introduced. However, it is hard to verify the real impact of the AEF experiment.

The usability of the method was criticized because the pair-wise comparison was laborious. It became apparent that one node in a factor hierarchy should contain five sublevel nodes at most. One more branch doubles the amount of pair-wise comparison cases. Another criticism was related to the factor hierarchy: each node should more clearly compete with other nodes, thus, turning the factor hierarchy into a hierarchy of architectural choices.

6.2. Business perspective

The evaluation resulted in 40 improvement proposals, which where prioritized during the AEF pilot project. All were important and some of them were screened the product properties. The framework also provided background material for marketing to explain architectural design decisions to customers. A set of proposals was referred to product creation processes, methods and practices. These proposals aimed to improve the quality of the process and also to the quality of the product indirectly.

6.3. Methodological perspective

The population of the hierarchy with feasible factors is crucial for successful evaluation. However, the creation of such a hierarchy is not trivial. We followed the rule of explaining a higher-level term by the set of lower level terms. Such a result is never semantically right or wrong; it only represents one kind of understanding of stakeholder needs. The second rule we used concerns the elements in the same level of hierarchy. These elements are regarded as independent and orthogonal compared to each other. This rule was helpful for constructing the hierarchy. The third rule for structuring the hierarchy is that all nodes under one upper level node should compete with each other as discussed before. One node should represent a choice which excludes others (at least partially) and forces evaluators to make a selection.

Furthermore, there should not be a single top-level factor, which represents only one stakeholder area. Rather, each factor should represent issues that concern several stakeholders at the same time. Otherwise the hierarchy prevents us to solve the possible conflicts between stakeholders. The hierarchy should bring up multi-stakeholder situations, in which several conflicts exist.

6.4. Strengths and weaknesses of the pilot case

The framework helps architects to process architectural factors in their design work and to communicate with each other. The framework has increased the concreteness of the product architecture for examination e.g. for sensitivity analysis.

The most important weakness is how to validate the result. We can communicate about architecture and identify the most effective points of improvement in systems development, but we have no measurable evidence that the result has real business effects.

7. Conclusions

We have developed an architecture evaluation framework (AEF) in order to improve architectural creation of telecom network elements. The framework aims to boost communication between and decision making of various stakeholders that covers the whole product life cycle. As facilitators, we had the possibility to study what these experts regard as important from their point of view. The AEF organizes a variety of architectural factors into a hierarchy. The structure purposefully stresses conflicting factors in order to bring out trade-offs in architecture design.

We have developed a method for using the framework and streamlined it for several usages: changing the depth of the hierarchy allows light and comprehensive evaluations. We have run two pilots to test the practice before the first real architecture evaluation. Experiences of one pilot are presented in this study. The main result in regard to the evaluated product and its product creation process has been the increased awareness and solidified understanding of architecture as a structured entity. With the help of the method and related tool support, the architecture can be discussed and manipulated also at a detailed level. Another benefit is the increased coverage of later phases of the product life cycle. When product experts participated in the process the hierarchy covered several valuable factors of logistics, maintenance and retirement. In a conventional situation, the technical part of the architecture would have been emphasized excessively.

Our findings of the product creation process were that architects did not know the architectural requirements adequately. They knew the requirements from their responsibility area which caused some important requirements to be underestimated or to remain unfamiliar. We have not been able to measure the impact of the AEF in our pilot case at the product level. However, the main benefit has been to change how architects think and understand architecture. In addition,
modularity and interface development has been emphasized in architectural design of base stations. In the future, we shall focus on creating metrics for verifying the effect of evaluation to the architecture in focus.

There are several expected benefits that make architecture evaluation frameworks valuable. They help to make architectural factors more explicit, concrete and increase common understanding. With the help of the AEF, architectural issues can be raised in perspective to the product life cycle. The AEF creates more emphasis on non-functional factors and catalysts new ideas. It also helps to locate trade-offs and to teach customers about principles and reasoning behind product architecture.

Acknowledgements
A part of this study has done in EU-ITEA MOOSE project (EUREKA program) that is partially funded by the National Technology Agency of Finland (TEKES). We would like to thank Dr. Dinesh Verma (Stevens Institute Of Technology) for his experience on architecture evaluation and systems engineering and Mr. Kari Sahlman (Nokia) for his contributions to the AEF project and evaluation case.

References


