P. Jönsson and C. Wohlin, "Understanding the Importance of Roles in Architecture-Related Process Improvement - A Case Study", Proceedings of International Conference on Product Focused Software Process Improvement (PROFES05), LNCS-series, Springer Verlag, Oulu, Finland, 2005.

Understanding the Importance of Roles in Architecture-Related Process Improvement - A Case Study

Per Jönsson and Claes Wohlin

School of Engineering, Blekinge Institute of Technology
PO-Box 520, SE-372 25, Ronneby, Sweden
per.jonsson@bth.se, claes.wohlin@bth.se

Abstract. In response to the increasingly challenging task of developing software, many companies turn to Software Process Improvement (SPI). One of many factors that SPI depends on is user (staff) involvement, which is complicated by the fact that process users may differ in viewpoints and priorities. In this paper, we present a case study in which we performed a pre-SPI examination of process users' viewpoints and priorities with respect to their roles. The study was conducted by the means of a questionnaire sent out to the process users. The analysis reveals differences among roles regarding priorities, in particular for product managers and designers, but not regarding viewpoints. This indicates that further research should investigate in which situations roles are likely to differ and in which they are likely to be similar. Moreover, since we initially expected both viewpoints and priorities to differ, it indicates that it is important to cover these aspects in SPI, and not only rely on expectations.

1 Introduction

Constraining factors such as time and budget make software development a challenging task for many organisations – a challenge that is leveraged by the fact that software plays an increasingly large role in society. In order to handle the challenge and to turn the software industry into an engineering discipline, it is necessary to put the processes in focus [27]. The goal of Software Process Improvement (SPI) is to create an infrastructure that enables effective methods and practices to be incorporated into the business [1].

The success of SPI depends on a number of factors, one of which is user (staff) involvement [21]. It has been reported that process users' attitudes often are disregarded in quality initiatives, and that putting them in the spotlight when designing SPI is an important step towards success [1, 10]. To involve process users and to regard their attitudes can be far from trivial, because process users do neither necessarily have the same viewpoints, nor the same priorities. This paper presents a case study in which we examined the viewpoints and priorities of process users at Ericsson AB, Sweden, to pinpoint differences and similarities among roles. We selected the role perspective since a number of publications report that role can be a discriminating factor when it comes to views in SPI [2, 4, 5, 6, 10, 14, 25].

Generic SPI frameworks, such as SEI's IDEALSM [17], Quality Improvement Paradigm (QIP) [3], and PROFES [20], all contain two important ingredients: a characterisation (or assessment, or appraisal) of the current process, and an improvement plan (or roadmap, or actions). The viewpoints of the process users are crucial in the characterisation phase, because the more diverse they are, the harder it becomes to form a baseline. Similarly, the priorities of the process users are crucial in the planning phase, because the more diverse they are, the harder it becomes to create a plan that satisfies everyone.

1.1 Background and Research Setting

Ericsson AB, Sweden, is one of the largest suppliers of mobile systems in the world, and has as customers some of the world's largest mobile operators. The study was conducted at one of Ericsson's offices (hereafter referred to as the company), which at the time had about 400 employees.

The objective of the study was to prepare improvement of the architecture documentation process at the company by examining process users' viewpoints and priorities with respect to their roles. By doing so, we were able to create an awareness of the need for and scope of SPI. With "architecture documentation process", we refer to the process of documenting the software architecture and keeping the documentation up-to-date. This is not necessarily an explicitly defined process of its own, but could, for example, be part of the development process. Our tool for examining viewpoints and priorities was a questionnaire with quantitative questions about architecture documentation.

In advance, we expected to see both diverse viewpoints and diverse priorities among process users regarding the architecture documentation process. The reason for this was mainly that architecture documentation typically has different stakeholders, such as project managers, product managers, designers and testers, most of whom have different needs and knowledge. Both needs and knowledge are factors that tend to affect how you view things and what you think is important. This is one of the reasons that software architectures should be designed and documented using multiple architectural views [5]. Since the organisational role most likely affects both needs and knowledge, we anticipated differences in both viewpoints and priorities.

We apply statistical methods to test for differences among roles. For this purpose, the following statistical hypotheses are evaluated (independently for viewpoints and priorities):

- Null hypothesis, H_0: *There are no differences among roles*.
- Alternative hypothesis, H_A: *There is a difference among roles*.

This paper is justified by two main reasons. First, it adds to existing research about similarities and non-similarities among roles, which is necessary to better understand the impact of roles in various research contexts. Second, it provides an example of empirical SPI research targeted at industry, with the focus on creating an understanding of process users' viewpoints and priorities.

1.2 Architecture-Related Process Improvement

As mentioned, we explored process users' viewpoints and opinions in order to prepare for improvement of the architecture documentation process. However, the questions posed in the questionnaire were more germane to the product (i.e., the architecture documentation) than to the process (i.e., documenting the architecture). Our reason for posing questions about the product rather than the process was that we considered the product to be more tangible to the process users than the process itself. In other words, we expected that we would get more well-founded answers by asking about the product. Furthermore, we argue that the quality of the documentation reflects the quality of the process of documenting it as much as, for example, the quality of requirements reflects the quality of the process of eliciting, formulating and managing them.

Since the software architecture of a system is a fundamental building block that has many stakeholders within the organisation, changes to architecture-related processes can have great organisational impact, including new workflows, altered mindsets and changed team structures. This further establishes the need for investigating differences in viewpoints and priorities among process users.

The paper is structured as follows. Section 2 addresses related work, while Section 3 explains the design of the study as well as how the study was carried out. The results are presented in Section 4, followed by a statistical analysis in Section 5, a general discussion in Section 6 and finally conclusions in Section 7.

2 Related Work

For an overview of the Software Process Improvement area, we recommend Zahran's book, which covers the topic from a practitioner's perspective [27]. Zahran discusses the general challenges of SPI, such as management commitment, buy-in from process users etc. There are some publications (see below) that discuss differences among roles in various contexts. In general, they show that there are differences among roles in some situations, but not in others. It all depends on what you evaluate.

Baddoo and Hall have studied de-motivators in SPI among software practitioners, in order to understand what hinders SPI success [2]. Dividing the practitioners into developers, project managers and senior managers, they find both common and unique de-motivators. They conclude that differences in de-motivators for SPI often are related to the roles that software practitioners have.

Berander and Wohlin have looked at agreement on SPI issues among traditional software development roles [4]. Their findings indicate that there is agreement among roles about communication of processes, but disagreement about, for example, importance of improvement and urgency of problems.

Conradi and Dybå have investigated how the use of formal routines for transferring knowledge and experience is perceived by developers and managers [6]. Their results show that there is a difference between the two groups; developers are more sceptical to formal routines, while managers take them for granted.

Hall and Wilson have studied views of quality among software practitioners in UK companies, with focus on two groups - managers and developers [10]. According to their findings, developers and managers differ in that they have different primary quality concerns, although the differences are not conflicting.

Karlström et al. present a method for aggregating viewpoints of process users in an organisation [14]. The essence of the method is to let process users rate factors believed to affect the SPI goal according to their viewpoints. The authors divide the process users into two groups, managers and engineers, based on their roles, and conclude that the groups differ on some factors, but not on all.

In [25], Svahnberg has studied how participants in an architecture assessment form groups when prioritising quality attributes and assessing architecture candidates. Svahnberg concludes that the role of the participant is the main influence when prioritising quality attributes, but not when assessing architecture candidates.

Questionnaires are often used as instruments in Software Process Assessment, for example the SEI Maturity Questionnaire for CMM-based assessment and the BOOTSTRAP questionnaire [27]. Klappholz et al. have developed an assessment tool, ATSE, for assessing attitude towards and knowledge of the development process [15]. While questionnaires in process assessment commonly measure the effectiveness of an improvement programme, our questionnaire was rather a part of the preparations for an upcoming improvement programme. In other words, it was not a substitute for a process assessment questionnaire.

3 Design

In this section, we outline the design of the study. We describe the contents of the questionnaire, the sampling and response rate, the technique for treating missing data, roles in the study and finally threats to the validity of the study.

3.1 Questionnaire Design

In order to assess process users' viewpoints and priorities, we designed a questionnaire consisting of six questions collecting data about the current state of the architecture documentation (the *infrastructure questions*), and one question about how to improve the architecture documentation (the *improvement question*). There was also a question asking about the process user's organisational role.

The infrastructure questions, which we believed would yield different results for different roles, were formulated as follows (words in bold are keywords used for identifying the questions later):

1. "In your opinion, to what extent does architecture documentation **exist**?"
2. "How would you, in general, describe the **form** of the architecture documentation?"
3. "How would you judge the **quality** of the documentation?"
4. "In your opinion, to what extent is architecture documentation **updated** as the system evolves?"

5. “In your opinion, how well does the architecture documentation **match** the actual architecture?”
6. “Imagine being a newly hired employee – how easy would it be to gain **insight** into the system using the current architecture documentation?”

For these questions, five point Likert scales (i.e., with five response options for each question) were used. Such scales are ordinal and are often used for collecting degrees of agreement [22]. For almost all questions, a low score indicated a negative response (little, low, seldom), while a high score indicated a positive response (much, high, often). The scale for the second question was slightly different from the others, though, in that a low score indicated text-orientation, while a high score indicated model-orientation.

The improvement question was formulated as follows: “If architecture documentation was to be improved in some way, what do you think is the most important purpose it could serve in addition to the present ones?” Five predefined purposes were given: change impact analysis, risk analysis, cost analysis, driver for development and system insight. In case these were not satisfactory, the respondent could choose “other” and suggest a new purpose as the most important one.

3.2 Sampling and Response Rate

The recipients of the questionnaire were selected using systematic sampling [22]. We obtained a list of all employees from an employee directory, and selected every second person on the list for the study. The reason for this was that another study was performed simultaneously and the employees were shared evenly between the studies.

The recipients of the questionnaire were given two weeks to fill it out and return the answers. The recipients were also allowed to reject the questionnaire if they had no time available or if they felt that it was not relevant for them. The two-week deadline seemed reasonable even for people with heavy workload. After one week, a reminder was sent to those who had not already responded or explicitly rejected the first distribution.

The population consisted of the 400 persons employed at the company. The selection of employees described above resulted in a sample of around 200 persons. While some responses were discarded because they contained invalid answers to some questions, around a third of the sample, or 65 persons, did give valid responses to the questionnaire. The other two thirds explicitly rejected the questionnaire, chose not to respond or were unable to respond before the deadline. Not all 65 respondents did answer all the questions, however. Because of that, some respondents had to be discarded, while some could be kept by imputing missing data in their answers, as described in the next section. As a result, the data presented in this paper are based on answers from 58 respondents.

In order to verify that the respondents were representative for the population, we examined the departmental distribution. We could not use the role distribution, since the roles were known only for the respondents, not the entire population. We saw that there were only minor differences in departmental distribution between the respondents and the population, and consequently considered the respondents being representative for the population. In this context, it should also be noted that the roles are in many cases closely linked to departments (i.e., testers come from the test department and so forth). While the respondent drop-out is a threat to the external validity of the study, as discussed in Section 3.5, we consider the absolute size of the sample to be satisfactory.

3.3 Treatment of Missing Data

As implied in the previous section, some of the initial respondents did not answer all the questions discussed in this paper. To be able to keep as many data points as possible, missing answers for those respondents that had answered at least five of the seven questions were imputed. The respondents that had answered only four questions or less were discarded, leaving 40 complete cases and 18 cases to impute. Only the complete cases were used as basis for the imputation.

The answers were imputed using the hot-deck k-Nearest Neighbour imputation technique, which imputes values based on the k cases most similar to the target case [7]. We have, in previous work, found this imputation technique to be suitable for Likert data [13]. The similarity metric used was the Euclidean distance calculated only for the infrastructure questions (the improvement question was disregarded because of its nominal scale). As a replacement for a missing data value, the median of the k nearest cases was used for the infrastructure questions, while the mode was used for the improvement question. Based on our previous findings, the value of k was chosen to 7, which is approximately the square root of the number of complete cases ($k = 6$, while closer, would be unsuitable when calculating the median) [13]. In the few cases when the mode was not unique at $k = 7$, k was increased to 8 instead.

3.4 Roles

The organisational roles of the respondents are rather specific for the company. Thus, in order to increase the generalisability of the results, and the replicability of the study, we have mapped the organisational roles to traditional software engineering roles. We mean that these represent the parts of software development normally discussed in software engineering literature [24]. To ensure the relevance of the resulting role list, the mapping was carried out together with a company-appointed specialist. The resulting roles, with abbreviated form and number of respondents within parentheses, are:

- *Designer* (D, 6) - creates system descriptions that the programmer can base the implementation on [19].
- *Programmer* (P, 8) - writes code that implements the requirements [19].
- *Tester* (T, 13) - catches faults that the programmer overlooks [19].
- *Functional manager* (FM, 5) - responsible for staffing, organizing, and executing project tasks within their functional areas [18].
- *Project manager* (PRJ, 8) - plans, directs and integrates the work efforts of participants to achieve project goals [18].

- *Product manager* (PRD, 4) - responsible for product related planning activities [16].
- *Architect* (A, 11) - makes decisions, coordinates, manages dependencies, negotiates requirements, recommends technology etc. [8, 11]
- *Process group member* (PGM, 3) - facilitates the definition, maintenance and improvement of the software processes used by the organisation [12].

The process group member role is derived from Humphrey's Software Engineering Process Group (SEPG) [12]. Persons who work with infrastructure, which is seen as a kind of process support, are also included in this role. The architect role stems from an organisational role with responsibility for the system as a whole and its survival in the product chain. Architecture work is a part of this role, but implementation and low-level design is not.

3.5 Validity Threats

In this section, the most important threats to the validity of the study are presented together with measures that have been taken to avoid them.

Construct Validity. Construct validity is concerned with the design of the main study instrument (i.e., the questionnaire) and that it measures what it is intended to measure [22]. A threat to construct validity is that not all respondents may have had the same perception of what architecture is. In order to avoid that, a common definition of the term "software architecture" was given in the questionnaire, and each respondent was given the opportunity to disagree and give his or her own definition.

To counter the threat that some questions would be easier to answer for persons with certain roles, the questionnaire was checked by a screening group at the company. Because several roles (product manager, project manager, architect, designer and programmer) were represented by the persons in the screening group, there should be little bias towards any particular role. The screening group also verified that the role list in the questionnaire did contain relevant organisational roles.

External Validity. External validity is concerned with the generalisability of the results [26]. The main threat to external validity is that differences (and non-differences) found among roles could be true only for the studied company. Since we are dealing with just one case, this threat cannot be ruled out. There are, however, two circumstances that we believe make our findings interesting in a wider context. First, the studied company operates, as stated, on the world market with several large, international customers. As such, it should be considered a strong industrial case. Second, by mapping the organisational roles to traditional software engineering roles, generalisability to a wider software engineering community, and greater replicability of the study, has been strived for.

Another threat to external validity is the fact that we cannot guarantee that the distribution of roles for the respondents equals the distribution of roles in the population. Since we do not know the roles of all the persons in the population, we cannot avoid this threat. We have tried to ensure that the respondents are representative of the population by looking at the departmental distribution instead.

Internal Validity. Internal validity is concerned with the relationship between the treatment (i.e., the questionnaire) and the outcome (i.e., the results collected) [26]. The imputation of data described in Section 3.3 is a threat, since it essentially is fabrication of data. However, of the incomplete cases, almost all had only one question out of seven unanswered. Only one incomplete case had two questions unanswered. Furthermore, we based the choice of imputation technique on our previous, supporting, findings.

Conclusion Validity. Conclusion validity is concerned with the statistical relationship between the treatment and the outcome [26]. A threat to conclusion validity is that we, in the data analysis, use the Kruskal-Wallis test when the number of ties in the data is large. To address this, we apply an additional Chi-Square test to validate the outcome of the Kruskal-Wallis test. It should also be noted that the Kruskal-Wallis test includes correction for ties. Thus, the effect of this threat should be minimal.

Another threat to conclusion validity is that we use the Chi-Square test when the expected frequencies in the cells are relatively small. While this may mean that the outcome is not reliable, Siegel acknowledges that for a large number of cells, the expected frequencies should be allowed to be small [23]. In our case, the number of cells is nearly 50, which Siegel hints should be enough to allow small expected frequencies. Thus, we believe that the effect of this threat should be small.

4 Results

In this section, we present the results on the questionnaire that was distributed at the company. We show the distribution of answers for the infrastructure questions as well as for the improvement question. The results are analysed statistically in Section 5.

4.1 Infrastructure Questions

Fig. 1 shows, for each of the infrastructure questions, how the answers are distributed on the response options. The exact phrasing of the questions can be found in Section 3.1. The question about to what extent documentation exists is special in the sense that the respondents made use of all five response options. As response options 3 and 4 account for 50% or more of the answers from all roles, the agreement among roles can be seen as large. Internally, architects (A) and designers (D) disagree more than the other roles, whereas on the whole, there is a consensus that documentation exists to a large extent.

For the question about documentation form, it is apparent that response option 2 dominates the answers (i.e., accounts for 50% or more) from all roles except the architects, meaning that most roles consider the documentation to be more text-oriented than model-oriented.

The answers to the question about how well the documentation matches the system indicate that all roles consider the match to be medium to good (as response options 3 and 4 dominate the answers). Again, architects have greater internal disagreement than the other roles. Also, functional managers (FM) and product managers (PRD) have a slightly more positive view than the other roles.

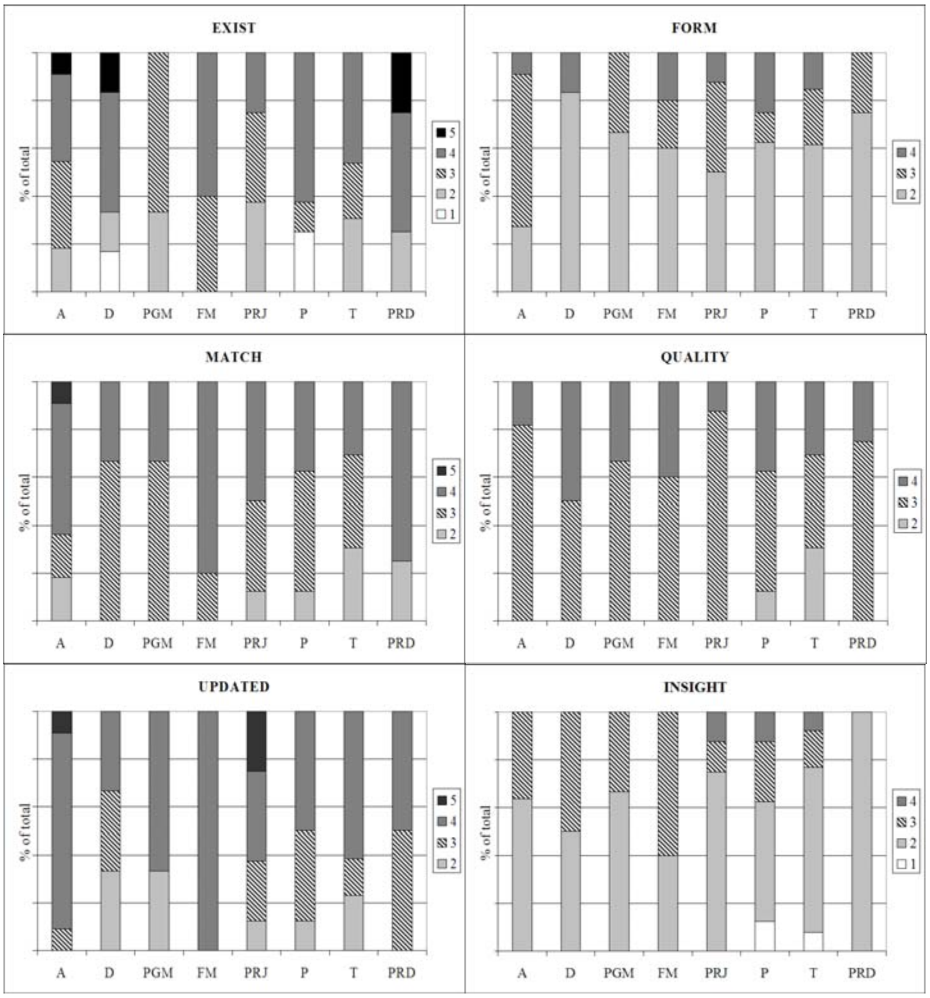

Fig. 1. Infrastructure questions; *answer distribution* (*y axis*) per *role* (*x axis*)

The question about quality of documentation has the narrowest distribution of answers of all the infrastructure questions. The respondents only made use of three of the response options in total, while most roles only made use of two, namely 3 and 4. Hence, the roles seem to agree that the documentation quality is medium or slightly above medium. The designers have a somewhat more positive view than other roles, whereas testers and programmers stand out because some of them consider the quality to be below medium.

In the answers to the question about to what extent documentation is updated, there seem to be a general consensus that the update frequency is high, with architects and functional managers having the most positive views. For this question, project managers (PRJ) have larger internal disagreement than the other roles.

Finally, for the question about how easy it is to gain insight into the system using the documentation, response option 2 dominates the answers from all roles except designers (where it is tied with response option 3) and functional managers. This means that there is agreement among the roles also for this question. Here, both programmers (P) and testers (T) have more internal disagreement than the other roles.

4.2 Most Important Improvement

When answering the improvement question, the respondents could choose “other” and add a new purpose if the predefined ones were not satisfactory (see Section 3.1). However, only one of the respondents chose to do this. We did not ask the other respondents to reconsider their answers with this additional purpose in mind for two reasons: (1) we estimated the added value to be minimal, and (2) we anticipated that it would be difficult to get new answers from all respondents. Consequently, only the predefined purposes are included in the analysis.

Table 1 shows the results from the improvement question. An ocular inspection reveals a couple of interesting differences (shown in the table as shaded cells with bold text). First, system insight (SI) was frequently specified by all roles except the product managers. Second, risk analysis (RA) and in particular cost analysis (CA) were more frequent for the product manager role than for any other role. In fact, risk analysis was not specified at all by most roles except product managers and testers. Third, change impact analysis (IA) was considerably more frequent for designers than for any other role.

Table 1. Most important improvement; answer distribution among roles

	IA	RA	CA	DD	SI
Architect	27.3	0.0	9.1	27.3	36.4
Designer	**66.7**	0.0	0.0	0.0	33.3
Process group member	33.3	0.0	0.0	0.0	66.7
Functional manager	20.0	0.0	0.0	0.0	60.0
Project manager	0.0	0.0	12.5	37.5	50.0
Programmer	25.0	0.0	0.0	25.0	50.0
Tester	0.0	7.7	15.4	0.0	76.9
Product manager	0.0	**25.0**	**50.0**	25.0	**0.0**

5 Analysis

In this section, the results from the questionnaire are analysed statistically. The null hypothesis stated in Section 1.1 is tested at significance level $\alpha = 0.05$. Two statistical tests are used, the Kruskal-Wallis test and the Chi-Square test [23]. We use these tests because we consider the data, being on ordinal and nominal scales, respectively, unsuitable for parametric tests. Both tests are applied to the infrastructure questions, whereas only Chi-Square is applied to the improvement question, because of its nominal scale.

5.1 Infrastructure Questions

The statistical significances of the results for the infrastructure questions are first calculated using the Kruskal-Wallis test. As can be seen in the second and third columns in Table 2, the results from the infrastructure questions are not significant at the selected significance level, as p exceeds 0.05 for all questions. This outcome aligns well with the results from the infrastructure questions presented in Section 4, where it can be seen that there is much agreement among the roles.

Since the Kruskal-Wallis test should be used with care when the number of ties in the data is large (as in our case) [9], we use the Chi-Square test for the infrastructure questions as well. The outcome of this test, presented in the rightmost three columns in Table 2, further confirms that there are no statistical significances in the results. This means that the null hypothesis cannot be rejected for the infrastructure questions.

Table 2. Kruskal-Wallis (left) and Chi-Square (right) outcome, all questions

	H (K-W)	p (df=7)	X^2	df	p
Exists	5.02	0.66	31.04	28	0.32
Form	4.38	0.73	12.46	14	0.57
Quality	3.62	0.82	16.41	14	0.29
Update	9.83	0.20	24.14	21	0.29
Match	5.88	0.55	17.56	21	0.68
Insight	4.12	0.77	14.91	21	0.83
Improve	N/A	N/A	47.71	28	0.046

5.2 Improvement Question

Because the data collected from the improvement question are nominal, we apply only the Chi-Square test and not the Kruskal-Wallis test. The outcome, presented in the bottom row in Table 2, shows that there is a statistically significant difference among the roles, as p is less than 0.05. Thus, the null hypothesis can be rejected in favour of the alternative hypothesis for the improvement question.

The overall Chi-Square does not pinpoint the differences. In other words, it does not identify exactly which roles that differ from others. To find the exact locations of the differences, the significances of all the partitions in the data are calculated. Each partition represents one pair of role and purpose (i.e., most important improvement), and has one degree of freedom. Simply speaking, the partition significance for a particular role-purpose pair is calculated based on the score of the pair and all scores to the left and above it. Consequently, the a priori order of the rows and columns affects the outcome of the partition significances [23]. We deal with this problem by performing an exhaustive calculation where all possible permutations of row and column order are evaluated. We argue that the pairs of role and purpose that are significant in more than 50% of the permutations can be considered significant overall. Table 3 shows the resulting pairs of role and purpose.

Table 3. Significant role-purpose pairs

Role	Purpose	Significance frequency (%)
Product manager	Risk analysis	53.5
Product manager	Cost analysis	54.7
Designer	Change impact analysis	59.1

It can be seen that product managers differ from other roles in that they more frequently chose risk analysis and cost analysis. Moreover, designers differ in that they more frequently chose change impact analysis than other roles. We see that this aligns well with some of the observations made in Section 4.2.

6 Discussion

In Section 4.1, we have seen that there seems to be much agreement among the roles regarding the infrastructure questions. This is supported by the statistical analysis in the previous section, which shows that there are no statistical differences among the roles for these questions. In other words, the respondents have fairly similar viewpoints. Our interpretation of this is that the architecture documentation process is well established at the company and that everyone has a joint understanding of how the architecture is documented and what the state of the documentation is. This is an important starting point when doing architecture-related process improvement, because it makes it easier to obtain a baseline of the current process. If the viewpoints had differed among roles, it could be difficult to find a common baseline.

Looking at individual roles, the results in Section 4.1 shows that some roles have more internal disagreement than other roles. This is true for architects and designers on the question about to what extent architecture documentation exists, and also for architects on the question about to what extent the documentation matches the system. The reason that there is disagreement within both roles may be that people with these roles work closer to the documentation and are therefore more sensitive to variations in its state. Moreover, internal disagreement is also noticeable for project managers on the question about to what extent the documentation is updated. A reason may be that project managers are more dependent on documentation update frequency when working with time and resource allocation. Finally, programmers and testers have larger internal disagreement than other roles for the question about how easy it is to gain insight into the system through the documentation. An explanation for this can be that these roles use the documentation for gaining system insight more than the other roles, and are therefore more sensitive to its ability to provide insight.

The results in Section 4.2 clearly show differences among the roles for the improvement question. System insight is the top improvement for all roles except designers and product managers. For these roles, change impact analysis and cost analysis are most important, respectively. The product manager role also stands out because it is the only role with strong focus on risk and cost analysis, and no focus at all on system insight. The reason that the product manager role differs on several accounts may be that this role has more market focus and less development focus than the other roles. The reason that the designer role differs may be that designers depend on the information in the architecture documentation more than other roles, and that this requires a strong focus on change impact analysis in order to handle ever-changing requirements.

The statistical analysis of the improvement question supports the differences among the roles outlined above. More specifically, designers' focus on change impact analysis and product managers' focus on cost and risk analysis are statistically significant. The fact that product managers do not consider system insight an important improvement is not significant, however. The existence of differences among the roles for the improvement question indicates that the priorities of respondents are different. This means that it becomes more difficult to create an improvement plan that satisfies all process users, since the plan needs to have a wider scope.

7 Conclusions

In this paper, we have presented results from a case study where we used a questionnaire for investigating process users' viewpoints and priorities regarding the architecture documentation process. The study was conducted at an office of Ericsson AB, Sweden. The objective of the study was to prepare improvement of the architecture documentation process at the company by examining process users' viewpoints and priorities with respect to their roles.

A large number of employees were asked questions about the current state of the architecture documentation and possible improvements of it. As explained in Section 1.2, we asked about the product (architecture documentation) rather than the process in order to have more tangible questions. The process users were divided into groups according to their software engineering roles. In order to analyse the results statistically, the following hypotheses were evaluated:

- H_0: *There are no differences among roles*.
- H_A: *There is a difference among roles*.

The two types of questions (current state and improvement) in the questionnaire were analysed separately. When analysing the results from the questions about the current state of the process, the null hypothesis could not be rejected. Our interpretation of this is that the company has succeeded in spreading knowledge about the process to the process users. However, some roles have larger internal disagreement compared to other roles for some of the question. This indicates that there may be underlying factors, other than role, that affect viewpoints.

When analysing the results from the improvement question, on the other hand, the null hypothesis could be rejected in favour of the alternative hypothesis. By performing a more in-depth statistical test and investigating the results from the improvement question directly, we found the following:

- System insight is not considered important to improve by the product managers. It is, however, the most important improvement for all other roles except the designers.
- Cost analysis and risk analysis are significantly more important to improve for product managers than for other roles. The reason may be a stronger market focus for this role than for the other roles.
- Change impact analysis is significantly more important to improve for designers than for other roles. The reason may be that designers use the documentation more than other roles when determining change impact.

Initially, we expected differences among roles both for viewpoints and priorities, since stakeholders of architecture documentation often are considered to have different needs and knowledge. Our expectations were however only fulfilled for priorities, as these clearly were different among the roles. It could be argued that it is “common knowledge” that priorities differ, which is why our expectations were set as they were. However, it remains difficult to foresee exactly how roles differ, which is important to know in process improvement. Furthermore, we had expected larger differences than were actually found. In any case, the fact that viewpoints, contrary to our expectations, did not differ while priorities did, leads us to conclude the following:

- It is important to cover process users’ viewpoints and priorities in process improvement work, because they may not coincide with the prevalent expectations.
- Further research should investigate in which situations roles are likely to differ and in which they are likely to be similar.

8 Acknowledgements

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the project “Blekinge - Engineering Software Qualities (BESQ)” (http://www.bth.se/besq).

References

1. Aaen, I.: Software Process Improvement: Blueprints versus Recipes. IEEE Software 20 (2003) 86-92
2. Baddoo, N., Hall, T.: De-motivators for software process improvement: an analysis of practitioners’ views. Journal of Systems and Software 66 (2003) 23-33

3. Basili, V., Green, S.: Software Process Evolution at the SEL. IEEE Software 11 (1994) 58-66
4. Berander, P., Wohlin, C.: Differences in Views between Development Roles in Software Process Improvement – A Quantitative Comparison. Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE'04), Edinburgh Scotland (2004) 57-66
5. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford J.: Documenting Software Architectures: Views and Beyond. Addison-Wesley (2003)
6. Conradi, R., Dybå, T.: An Empirical Study on the Utility of Formal Routines to Transfer Knowledge and Experience. ESEC/SIGSOFT FSE (2001) 268-276
7. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons (1973)
8. Fowler, M.: Who Needs an Architect? IEEE Software 20 (2003) 11-13
9. Garson, G. D.: PA 765 Statnotes: An Online Textbook. North Carolina State University, http://www2.chass.ncsu.edu/garson/pa765/statnote.htm, last checked March 3 (2005)
10. Hall, T., Wilson, D.: Views of software quality: a field report. IEE Proceedings on Software Engineering 144 (1997) 111-118
11. Hofmeister, C., Nord, R., Soni, D.: Applied Software Architecture. Addison-Wesley (2000)
12. Humphrey, W.: Managing the Software Process. Addison-Wesley (1989)
13. Jönsson, P., Wohlin, C.: An Evaluation of k-Nearest Neighbour Imputation Using Likert Data. Proceedings of the 10th International Software Metrics Symposium (METRICS'04), Chicago IL USA (2004) 108-118
14. Karlström, D., Runeson, P., Wohlin, C.: Aggregating Viewpoints for Strategic Software Process Improvement – a Method and a Case Study. IEE Proceedings on Software Engineering 149 (2002) 143-152
15. Klappholz, D., Bernstein, L., Port, D.: Assessing Attitude Towards, Knowledge of, and Ability to Apply, Software Development Process. Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03), Madrid Spain (2003) 268-278
16. Lehmann, D.R., Winer, R.S.: Product Management. 3rd edn. McGraw-Hill (2002)
17. McFeeley, B.: IDEALSM: A User's Guide for Software Process Improvement. Handbook CMU/SEI-96-HB-001. Software Engineering Institute, Carnegie Mellon University (1996)
18. Nicholas, J.M.: Project Management for Business and Technology: Principles and Practise. 2nd edn. Prentice Hall (2001)
19. Pfleeger, S.L.: Software Engineering: Theory and Practise. Intl. edn. Prentice Hall (1998)
20. PROFES User Manual – Final Version 1999. http://www.vtt.fi/ele/profes/PUMv10.pdf, last checked March 3 (2005)
21. Rainer, A., Hall, T.: Key Success Factors for Implementing Software Process Improvement: a Maturity-Based Analysis. Journal of Systems and Software 62 (2002) 71-84
22. Robson, C.: Real World Research. 2nd edn. Blackwell Publishing (2002)
23. Siegel, S., Castellan Jr., N.J.: Nonparametric Statistics for the Behavioral Sciences. Intl. edn. McGraw-Hill (1988)
24. Sommerville, I.: Software Engineering. 6th edn. Addison-Wesley (2001)
25. Svahnberg, M.: A Study on Agreement Between Participants in an Architecture Assessment. Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03), Rome Italy (2003) 61-71
26. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers (2000)
27. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley (1998)