

P. Jönsson and C. Wohlin, "A Study on Prioritisation of Impact Analysis Issues: A Comparison between Perspectives", Software Engineering Research and Practice in Sweden, Västerås, Sweden, 2005.

# A Study on Prioritisation of Impact Analysis Issues: A Comparison Between Perspectives

Per Jönsson<sup>\*</sup>  
per.jonsson@bth.se

Claes Wohlin  
claes.wohlin@bth.se

School of Engineering  
Blekinge Institute of Technology  
PO-Box 520, SE-372 25, Ronneby, Sweden

## ABSTRACT

Impact analysis, which concerns the analysis of the impact of proposed changes to a system, is an important change management activity that previously has been studied mostly with respect to technical aspects. In this paper, we present results from a study where issues with impact analysis were prioritised with respect to criticality by professional software developers from an organisational perspective and a self-perspective. We visualise the prioritisation in a way that allows us to identify priority classes of issues and to discuss differences between the perspectives. Furthermore, we look at issue characteristics that relate to said differences, and identify a number of improvements that could help mitigate the issues. We conclude that looking at multiple perspectives is rewarding and entails certain benefits when dealing with software process improvement, but also that the prioritisation and visualisation approach seems to be good for optimising software process improvement efforts in general.

## Categories and Subject Descriptors

K.6.3 [Software Management]: Software process

## General Terms

Experimentation, Human Factors, Management

## Keywords

Impact Analysis, Issues, Perspective, Prioritisation, Software Process Improvement, Change Management

## 1. INTRODUCTION

Impact analysis plays a crucial role in change management, where it is used to determine the impact of a proposed change to the product. Bohner and Arnold have defined impact analysis to be the activity of identifying the potential consequences of a change or estimating what needs to

be modified to accomplish a change [2]. Pfleeger, on the other hand, has more management focus and defines impact analysis to be the evaluation of risks as well as effects on resources, effort and schedule [11]. These definitions go hand-in-hand, and point to the fact that impact analysis is, and needs to be seen as, a holistic activity.

Impact analysis typically contains a subjective element, in that the outcome of the analysis depends much on the performance of the analyst. Tools for supporting impact analysis often make use of traceability information, but, in our experience, it is common that the expertise, knowledge and gut feeling of the analyst control the analysis outcome. This means, however, that the analysis is subject to judgement errors. For example, previously when studying uses and issues of impact analysis, we found that underestimated system impact and overlooked affected parties were some of the top rated issues [6]. Also, Lindvall and Sandahl have reported on a study where professional developers underestimated the impact with a factor of three [10].

Traditionally, the main forum for impact analysis research has been the software maintenance field. However, impact analysis in pre-maintenance development offers additional challenges, such as the possible absence of source code and design models in early development, and requires therefore special attention. Furthermore, there is much focus on technical aspects of impact analysis, for example the creation of algorithms for calculating how changes propagate in a system. Considerably less attention has been devoted to studying organisational aspects of impact analysis, such as its role as a change management activity.

In this paper, we present a comparison of two perspectives with respect to how they affect the prioritisation of issues associated with impact analysis. In the underlying study, we originally looked at how issues as well as uses differed between three organisational levels. We subsequently published the results (see [6]), but did not pursue the fact that the participants of the study were asked to prioritise issues from two different perspectives. Here, we focus on the perspectives and discuss how they can be used to facilitate the selection of issues to deal with in process improvement, as well as on characteristics of issues that contribute to the differences between the perspectives. We also identify a number of improvements to help mitigate the most important issues.

<sup>\*</sup>Corresponding author.

The two perspectives are the *organisational* perspective and the *self*-perspective. When prioritising issues, the organisational perspective means to consider what is best for the organisation, while the self-perspective rather has to do with what is best for oneself. Ideally, these perspectives should be similar, but it can be argued that people, due to having personal agendas, may have interests that do not align entirely with those of the organisation.

The remainder of the paper is structured as follows. Section 1.1 details the research context, including research setting and research question. Section 2 presents related work with respect to both software development and prioritisation, while Section 3 describes the methodology used in the study. Results are presented in Section 4 and further analysed and discussed in Section 5. Finally, conclusions are given in Section 6.

## 1.1 Research Context

The study was conducted at a large Swedish software development company operating on a world market. The company is one of the leaders within its domain, and has several large international customers. The population we wish to generalise to is industrial/professional software developers<sup>1</sup> in general rather than just within the company. We believe this to be possible due to the fact that the company deals with large-scale software development and is ISO 9000 certified. It can be assumed that the challenges and characteristics of the development work exist in other companies of the same magnitude as well, and to some extent also in smaller companies.

The research question for this work is: *What difference does the perspective make in prioritisation of impact analysis issues?*

In the study, the participants prioritised issues according to criticality in two different ways:

- According to a self-perspective, i.e. such that the top-ranked issues were critical for the participants themselves.
- According to an organisational perspective, i.e. such that the top-ranked issues were perceived as critical for the organisation.

In this context, the perspectives can be expected to differ in that the issues critical from the self-perspective should concern the execution of specific tasks or activities, whereas issues critical from the organisational perspective should concern the bird's-eye view of the development work, including keeping deadlines and delivering the right product. If the perspectives differ, it means that personal priorities are not well aligned with organisational priorities. In an ideal situation, the perspectives should be equal, as that would mean that all personal priorities are in line with what is best for the organisation.

---

<sup>1</sup>With *software developers*, we mean everyone involved in the development work, including managers.

By looking at the perspectives, it is possible to divide the issues into priority classes from a process improvement point of view. Issues that are critical from both perspectives belong to a first-priority class, while issues that are critical from only one perspective belong to a second-priority class. There is also the low-priority class, which contains issues that are not critical from any of the perspectives (or only slightly critical).

## 2. RELATED WORK

Several researchers report on differences between managers and engineers in the context of software process improvement (SPI), for example concerning views of software quality [5], use of formal routines to transfer knowledge and experience [3] and how they rate factors affecting an SPI goal [8]. Even though both managers and engineers have personal agendas, our experience is that managers often have a more holistic view of the development work than engineers. Due to this, the personal priorities of managers are likely to align well with those of the organisation, whereas the personal priorities of engineers should be narrower. Thus, differences between managers and engineers could be attributed to differences between the organisational alignment of their default perspectives.

The notion of perspectives can be found in other research, for example on perspective-based reading (PBR). PBR is a scenario-based reading technique introduced by Basili et al., with the purpose of maximising the number of errors found when reading a document by reading it from several different perspectives [1]. The idea is that each perspective finds its own unique errors, and that the union of the perspectives provides a good coverage of the document being read. PBR is a good example of how to take advantage of differences between perspectives, although there is research that indicates that the perspectives need not differ with statistical significance [12].

Finkelstein and Sommerville discuss perspective in relation to viewpoints [4]. A viewpoint consists of an agent (i.e., actor) and a perspective. Large-scale software development requires, due to its complexity, multiple viewpoints and thus multiple perspectives. According to Finkelstein and Sommerville, differences in perspectives stem from the agents' different responsibilities and roles and thus variations in their goals and opinions. Furthermore, it is argued that multiple perspectives need to be integrated if there is a common or shared goal [4]. In SPI, the common goal is to create a successful improvement programme that addresses the organisation's software processes (see, for example, [15]). This motivates the relevance of looking at multiple perspectives when prioritising impact analysis issues.

Karlsson and Ryan describe a requirements prioritisation approach, in which requirements are prioritised from two different perspectives—a value (customer) perspective and a cost (developer) perspective [7]. Their visualisation of the prioritisations has inspired the visualisation approach used in this paper. A difference between our perspective comparison and their perspective comparison is that in their case there is a preferable situation, i.e. that the value/cost ratio for a requirement should be as high as possible. In our case, there is nothing that says that a high organisational/self

ratio is more preferable than a low one.

### 3. METHOD

In this section, we present the method used in the study. As the focus of this paper is on perspectives when prioritising impact analysis issues, the details concerning organisational levels have been excluded here. See [6] for a description of the method with this aspect included.

#### 3.1 Overview

The study consisted of three main steps: interviews, follow-up prioritisation, and workshop. In the interview step, 18 persons involved in the change management process were asked questions about impact analysis and change management. The roles of the participants ranged from programmers to product managers, but each one had in one way or another a relation to the impact analysis activity, as decision-maker, technical advisor, change request writer, implementer and so on.

The sampling of the participants can be characterised as *convenience sampling* [13], as no specific systemacy was used in selecting people to interview. Instead, people were selected mainly based on recommendation (and availability), although some were approached at random based on their role. As one of the initial objectives of the study was to look at organisational levels, the sampling of participants were largely controlled by to which extent the levels were thought to be represented.

In the follow-up prioritisation, the participants prioritised issues elicited during the interviews. We anticipated that each participant would only list a few issues, and thus not see the whole picture. This was the reason that the prioritisation was a separate step rather than included in the interviews.

The purpose of the workshop was to let the participants discuss the prioritised issues, and possibly come up with ideas how to mitigate them. Furthermore, as several roles were represented by the participants, the workshop could allow people to gain greater understanding of each other's views.

The following sections describe the steps of the study in more detail.

#### 3.2 Interview Design

The interview instrument contained seven main topics, of which each was associated with one or more open-ended questions. The questions are shown in Appendix A. For this paper, one of the topics is of primary interest: *potential issues*. This topic was associated with one question: *Which potential issues are associated with performing impact analysis?*

Note that we asked about potential issues rather than actual ones. The reason for this was to avoid limiting the generalisability of the results by extracting company-specific issues only. However, we had no means of verifying whether this condition was met or not.

The remaining topics were more qualitative in their nature, and were intended both for providing a context for and for collecting hidden or implicit knowledge about the issues. As discussed earlier, we did not intend for the participants to prioritise during the interviews, since we expected each of them to see only a subset of the possible issues.

In order to ensure the appropriateness and clarity of the questions, the interview instrument was developed in close cooperation with the company where the study was conducted. Furthermore, we performed a pilot interview in order to find inconsistencies and problems with the interview instrument. The pilot interview resulted in only minor changes to the interview instrument, which lead us to include the interview in the study.

The interviews were *semi-structured*, meaning that it was not necessary to follow the predefined question order strictly, and that the wording of questions was not seen as crucial for the outcome of the interviews [13]. The participants could speak rather freely, but we made sure that all questions were answered in one way or another.

The participants were asked if they would agree to join in a follow-up activity, where they should prioritise the issues elicited during the interviews. This was done in order to prepare the participants and increase their commitment towards the follow-up prioritisation.

A great variety of roles were covered in the interviews, including developer, tester, technical coordinator, manager (functional, product and project) and system architect. It should also be noted that the participants in general had been working at the company for a long time, and were thus familiar with processes and routines.

##### 3.2.1 Results Triangulation and Filtering

In order to obtain an as complete list as possible of issues, we made use of a triangulation scheme, consisting of three information sources: (1) the list generated in the interviews, (2) qualitative information from the interviews, and (3) information from the literature.

By using information from all interview topics, it would be possible to extract both explicit and implicit knowledge about issues. For example, it happened on several occasions that participants mentioned issues while talking about processes and tools. These issues were specifically tagged and in the triangulation step merged with the ones explicitly listed.

By collecting information from the literature, we would be able to add issues of which the participants were not aware.

The filtering part was intended to remove redundancies and inconsistencies in the issue list by merging similar issues together, and by discarding issues that were not directly related to impact analysis.

#### 3.3 Follow-up Prioritisation

In order to obtain information about the relative importance of all issues, a prioritisation step was designed as a follow-up to the interviews. When prioritising, the participants should assign weights to the issues, such that the weights

should sum to 1000. Thus, each weight could be seen as a certain percentage of the total criticality of all issues. This is also known as *cumulative voting* [9]. Advantages of this method are that it is easy to learn and use, and that the resulting weights are on a ratio scale, i.e. it is meaningful to discuss the weight ratio between two prioritised items.

To facilitate for the participants, an Excel-based prioritisation tool was created. The tool helped the participants keep track of the sum of weights, and also gave instant feedback by presenting the resulting list of prioritised issues after the prioritisation.

Since we were interested in potential issues only (see Section 3.2), we asked the participants to prioritise without regard to actual issues. In other words, we wanted to avoid priorities biased towards issues currently visible in the organisation.

Initially, we decided that the prioritisation of issues should be done according to criticality from an organisational perspective. Put differently, the participants should prioritise such that the issue with the highest priority would be the one most critical to the organisation if it existed. However, we were also interested in knowing if the participants would prioritise differently from a self-perspective. Thus, the prioritisation was extended to include both perspectives.

To account for the problem that the first prioritisation of issues could affect the second one (due to maturation problems), a two-group design was used in this step, such that half of the participants should prioritise from the organisational perspective first, and the other half from the self-perspective first.

### 3.4 Workshop

As mentioned earlier, the workshop was meant to provide a forum for the participants to discuss the results from the follow-up prioritisation. A second purpose was to allow the participants to obtain a better understanding of each other's views.

As not all the participants could meet at one single occasion, two separate workshops were arranged instead of only one. Still, less than a third of the participants were able to take part in the workshops. The workshop results were compiled and sent to all participants afterwards, both to spread the results to those who did not attend, and to merge the results from the two separate workshops. Furthermore, the participants could in this way control that the compiled information was correct.

## 4. RESULTS

The interviews resulted in 25 explicitly and implicitly mentioned issues after irrelevant issues had been removed, and similar issues had been merged. We found six issues mentioned in the literature, but these were already among the 25. Thus, the resulting list contained 25 issues in total.

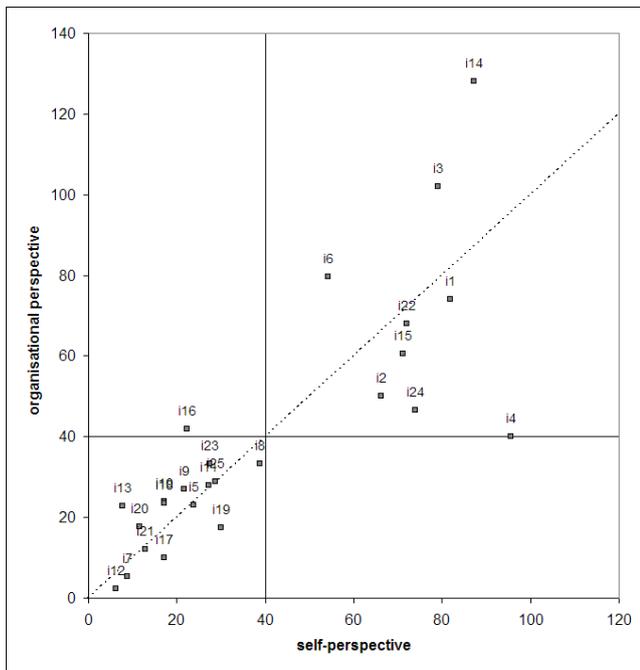
All issues can be seen in Table 1, labelled from i1 to i25. Note that the original issues were in Swedish and have been translated into English here. In the translation, we have strived to keep the main point of each issue.

id	Issue
i1	It is hard to get resources for performing impact analysis.
i2	There is not enough time for performing impact analysis.
i3	System impact is underestimated or overlooked.
i4	Change requests are unclear.
i5	Responsibility and product/project balance are difficult to handle for analyses that span several systems.
i6	Analyses are incomplete or delayed.
i7	Analyses require much expertise and experience.
i8	Analyses are too coarse or uncertain.
i9	It is difficult to handle conflicting and synergetic change requests.
i10	Analyses are not prevented from being disregarded.
i11	Existing traceability is manual and cumbersome.
i12	It is difficult to see trends and statistics for collective impact.
i13	Tools for supporting the analysis are missing.
i14	Affected parties are overlooked.
i15	Analyses are performed by the wrong persons.
i16	Change request decisions are based on interest.
i17	Requirements and baseline are missing for early change requests.
i18	Analyses and change implementation evoke stress.
i19	It is not possible to see the outcome of a change request.
i20	It is difficult to see status and updates for a change request.
i21	Different change request have different levels of complexity, and there is no single method for handling all levels.
i22	Cheap, short-term solutions win over good, long-term solutions.
i23	Solutions are specified with too much detail by high-level analysts.
i24	Hardware and protocol dependencies are difficult to handle for late change requests.
i25	Relevant structure and documentation to support the analysis are missing.

**Table 1: Issues elicited in the interviews**

Although a number of issues were removed from the initial list as they were not entirely relevant from an impact analysis perspective, the remaining issues vary in their closeness to impact analysis. For example, some issues concern change requests, some concern the actual analysis activity, some concern process details, and others concern the analysis results. This is of course less than optimal from a prioritisation point of view. The alternative would however be to divide the issues into groups and prioritise the groups separately, which would complicate the prioritisation further.

Figure 1 visualises the prioritisation of issues from the two different perspectives, with self-perspective on the x axis and organisational perspective on the y axis. The chart is divided into four boxes based on the average issue weight (25 issues with total weight sum 1000 means an average weight of 40). Thus, issues in the lower left box are less than average critical from both perspectives, while those



**Figure 1: Prioritised issues, comparison of perspectives**

in the upper right box are more than average critical from both perspectives. In the other two boxes, the issues are more than average critical from one perspective, and less than average critical from the other. The dotted line in the figure shows the one-to-one ratio, i.e. where issues are equally critical from both perspectives.

The vast majority of the issues found in this study were considered to be either of low criticality in general, or of high criticality in general. This is visible in Figure 1, where the issues are divided mainly between the lower left box and the upper right box. It can also be seen that the issues of high criticality in general (i.e., those in the upper right box) are spread out, meaning that the prioritisation definitively has been affected by the two perspectives. For example, there are four issues (i1, i3, i4 and i14) with priorities between 80 and 100 from the self-perspective, while their priorities range from 40 to 130 from the organisational perspective. It is interesting to note that the opposite pattern does not exist—there is no clear group of issues with very high criticality from the organisational perspective and a large variation in criticality from the self-perspective. This can be interpreted in two ways. Either there were no such issues, or the participants were reluctant to assign high criticality to an issue from the organisational perspective when the criticality was not also high from the self-perspective.

Initially, we expected the perspectives to differ quite much for two reasons. First, we believed that the fact that people have personal agendas should result in a prioritisation where a few issues were given high weights from the self-perspective. Second, and related, we argued that certain issues ought to be more germane to the organisational perspective and some to the self-perspective. However, the fact

that most issues are in the two boxes that are closest to the one-to-one ratio line in Figure 1, shows that our expectations were not entirely met. A larger difference in perspectives should have resulted in issues occurring also in the boxes where one perspective is dominant with respect to criticality.

## 5. ANALYSIS AND DISCUSSION

This section contains an analysis and subsequent discussion of the results presented in the previous section. Note that no statistical analysis has been performed on the results—the analysis here is based purely on observation.

Section 5.1 analyses and discusses the issues elicited and prioritised in the study. Section 5.2 focuses on improvements to help mitigate issues, and finally Section 5.3 and Section 5.4 cover outliers and threats to validity, respectively.

### 5.1 Issues

A goal of process improvement must necessarily be to mitigate as many issues as possible. However, seeing to the cost aspects, it is of course more optimal to focus on issues that are of high criticality. Having said that, it is important to note that mitigating an issue has a positive effect on people experiencing the issue, regardless of the criticality they have assigned to it from either perspective. Differentiating between the perspectives is a way for us to make more informed decisions about which issues to deal with first.

The division of the issue graph in Figure 1 into four boxes allows us to consider four classes of issues. From a process improvement point of view, the issues that occur in the upper right box in the figure are most important to try to mitigate, as doing so could eliminate issues with “dual-perspective” criticality. Thus, they could be seen as first-priority issues. Similarly, the issues in the lower left box could be ignored or postponed, as they have low overall criticality. The expected benefits are limited, and trying to mitigate the issues may not be worth while with respect to cost. Thus, they could be seen as low-priority issues.

The issues in the upper left and lower right boxes are second-priority issues, from a process improvement perspective. As can be seen in Figure 1, there is only one second-priority issue (i16), while one (i4) is on the border between being second-priority and first-priority. We see that i4—change requests are unclear—is even slightly less critical from the organisational perspective than i16—change request decisions are based on interest—while it is the single most critical issue from the self-perspective. The reason for the difference in self-perspective criticality is most likely that i4 is an issue that affects a specific activity, i.e. the analysis of change impact, while i16 is of more general nature. However, i16 ought to be more critical from the organisational perspective, as it is in the best interest of the organisation that change request decisions are guided by architecture and strategies, not interests.

If there are second-priority issues of both kinds, there is the question of whether to deal with the issues critical from the organisational perspective first, or with the ones critical from the self-perspective first. Mitigating the organisational issues is of course in the best interest of the organisation,

while mitigating the self-issues should facilitate the daily work of the individual software developers. As we see it, the selection of issues to try to mitigate should include both criticality and cost. Three aspects of dealing with issues are important:

1. Focus on issues that have high criticality, preferably first-priority ones.
2. Also focus on issues that can be mitigated in inexpensive ways.
3. Finally, consider improvements that affect as many issues as possible (discussed later).

When looking at the first-priority issues, i.e. the ones in the upper right box in Figure 1, it is possible to discern three groups of three issues each:

- Issues i3, i6 and i14 all have an organisational/self ratio above one, i.e. they are more critical from the organisational perspective than from the self-perspective.
- Similarly, issues i2, i4 and i24 have a ratio below one, i.e. they are more critical from the self-perspective.
- Finally, issues i1, i15 and i22 are close to the one-to-one ratio line, meaning that their criticality does not depend on perspective.

The characteristics of the nine first-priority issues can be summarised as follows:

- Issue i1 (hard to get resources for performing impact analysis) affects the time plan of the project and delays the change management process.
- Issue i2 (not enough time for performing impact analysis) is a stress factor and possibly degrades the quality of the analysis.
- Issue i3 (system impact is underestimated or overlooked) will result in later problems when the change is implemented, and may thus delay the project and possibly destabilise the system as all implementation details are not thought through.
- Issue i4 (change requests are unclear) affects, as mentioned above, the activity of analysing impact, since the basis for the analysis has low quality.
- Issue i6 (analyses are incomplete or delayed) has severe impact on the change management process, since decisions may be made on loose terms, which in the end may affect the product negatively.
- Issue i14 (affected parties are overlooked) has organisation-wide impact, as all parties involved in the development work are crucial for the right product to be developed in the right way. Disregarding some parties disturbs the pace and lowers the quality of the development work.

- Issue i15 (analyses are performed by the wrong persons) means that the resulting analyses may be of low quality, which could affect both the end product and generate extra work.
- Issue i22 (cheap, short-term solutions win over good, long-term solutions) means that the changes introduced in the system do not necessarily hold for the future. This may both destabilise the system and generate extra work later on.
- Issue i24 (hardware and protocol dependencies are difficult to handle for late change requests) generates much work as the dependencies need to be dealt with and are typically very problematic in the late phases of a project.

Based on the summary, it is not surprising to see i3, i6 and i14 forming a group with a high organisational/self ratio. These three issues affect process, project and end product aspects, and should therefore be critical from an organisational perspective. Issues i3 and i14 are also seen as critical from the self-perspective, which could be explained by the fact that they will reflect back on the implementation of the change, in that they will require additional individual effort.

Issues i2, i4 and i24, which form the group with a low organisational/self ratio, are all issues that are closely related to the execution of the analysis task. Thus, it is apparent why these were considered to be critical from the self-perspective.

Finally, issues i15 and i22 are related to both the bird's-eye development aspects and to the execution of the analysis task, whereas issue i1 seems to be mostly a bird's-eye issue. Thus, i1 ought to have a higher organisational/self ratio than it has. An explanation to why it was seen as equally critical from both perspectives may be that it not only affects time plans, but also the resource planning activity, which means additional effort for those responsible for planning resources.

## 5.2 Improvements

During the workshops, a number of possible improvements to help mitigate the issues at hand were discussed. They are presented here as proposals only, i.e. they have not been evaluated in any way.

Regarding improvements, it is of course desirable to select those that can mitigate as many issues as possible at the same time, for a reasonable cost. As many of the issues are related in one way or another (e.g., i3 and i14 both have to do with missed impact, albeit of different types), improvements are likely to have multi-issue impact. However, a multi-issue improvement may be very expensive, in which case other, possibly narrower, improvements have to be considered instead.

The following list shows the improvements that were identified during the workshops:

- S1** Limit the number of outstanding change requests at any time and use a selection process for taking in change requests.

- S2** Introduce different ways of handling different types of change requests. For example, use different change request “tracks” (e.g., based on change request priority).
- S3** Involve the design organisation more in the requirements specification work to avoid change requests related to requirements problems.
- S4** Plan early for the fact that there will be many change requests in the project.
- S5** Introduce a database for storing old impact analysis results to be used as a knowledge base for future analyses.
- S6** Introduce meetings where different development teams and subprojects discuss joint impact analysis efforts.
- S7** Involve the support organisation more in the impact analysis work to ensure focus on post-delivery aspects.
- S8** Missionise the target architecture more within the project to make sure that everyone knows about it.
- S9** Introduce tool and method support (such as checklists) to help individuals perform better and more robust analyses.
- S10** Act more towards the customer—meet the customer more often to discuss solutions. Listen to several customers instead of only one to avoid solutions that just a few need/want.
- S11** Anchor release intent in the design organisation to make sure that decisions go towards a common goal.
- S12** Do not allow change requests late in the project (close the change control board as early as possible).

Each improvement was associated with one or more issues. Put differently, an improvement package was formed for each issue, and most improvements were shared among several packages. Table 2 shows the mapping between issues and improvements. An x in a cell means that the improvement in the current row should be effective in mitigating the issue in the current column. It should be noted that the mapping of improvements to issues mainly took place during the workshops, although it was reviewed and in some cases completed afterwards.

As pointed out, improvements affecting several issues should be favoured over more localised improvements. It can be seen in Table 2 that improvement S1 affects six of the first-priority issues. All of these except i6 are more critical from a self-perspective than from an organisational perspective (see Figure 1). Furthermore, improvement S8 affects the remaining three first-priority issues, meaning that improvements S1 and S8 together form a good start when conducting process improvement. However, one single improvement is unlikely to completely resolve an issue—it is better to use a combination of multiple improvements.

In addition to the ones presented above, a number of improvements were tailored specifically for the studied company. As these are not generic, they have been left out here.

	i1	i2	i3	i4	i6	i14	i15	i22	i24
<b>S1</b>	x	x		x	x		x		x
<b>S2</b>	x	x			x		x		
<b>S3</b>	x	x							
<b>S4</b>	x	x			x		x		
<b>S5</b>			x		x	x	x		
<b>S6</b>			x			x			
<b>S7</b>			x			x			
<b>S8</b>			x			x		x	
<b>S9</b>			x			x	x		
<b>S10</b>							x		
<b>S11</b>							x		
<b>S12</b>								x	

**Table 2: Mapping of identified improvements to first-priority issues**

### 5.3 Treatment of Outliers

A participant who has prioritised completely different than other participants, for example by giving extremely high weights to certain issues, could be seen as an outlier. Outliers disturb statistical measures, such as the mean value, and may have severe impact on statistical tests. However, as we perform no statistical analysis based on the results, little attention has been devoted to outliers. In fact, none of the participants were removed in the presentation and analysis of the data.

Furthermore, we argue that dealing with outliers is of particular importance when the data consists of measurements of performance or effort, in which case outliers could be seen as the result of measurement or sampling problems. In our case, however, each data point is a subjective opinion, meaning that removing outliers would disqualify uncommon opinions.

A way of dealing with outliers resulting from uncommon or extreme opinions is to go back to the source and discuss the reason behind the outlier. This approach has not been pursued in this study.

### 5.4 Threats to Validity

The validity of a study is often divided into four types. *Construct validity* is concerned with the design of the main study instrument and that it measures what it is intended to measure [13]. A study has *internal validity* if there is a causal relationship between the treatment and the outcome [13]. *External validity* is concerned with the generalisability of the results [14]. Finally, *conclusion validity* has to do with being able to draw correct conclusions about how the treatment has affected the outcome [14].

The small sample size and the fact that we sampled based on convenience are threats to both internal and external validity, as the participants may not have been representative for the population. However, as the participants were selected mainly based on recommendations by several persons, we believe they were good representatives both of their roles and of professional software developers in general.

Furthermore, the fact that we focused on potential issues

rather than actual ones should increase the external validity. Also, the participants covered all issues from the literature, which indicates that their views of impact analysis were not company-specific.

A threat to internal validity is maturation, in our case that the first prioritisation of issues could have affected how the same issues were prioritised a second time (although from a different perspective). We tried to counter this threat by not letting all participants prioritise the perspectives in the same order (see Section 3.3).

Another threat to internal validity is instrumentation, i.e. that the instruments used in the study could have been badly designed. This threat affects conclusion validity as well, and possibly construct validity. We believe the threat is reduced due to the facts that we conducted a pilot interview to test the interview instrument (see Section 3.2), and that the prioritisation tool (see Section 3.3) was designed to alleviate the prioritisation effort of the participants as much as possible.

A threat to construct validity is that the participants may not have had the desired mindset when prioritising issues. As stated in Section 3.3, we asked the participants to prioritise as if the issues currently were not present, but we could not verify if they adhered to our request.

Another threat to construct validity is that the participants may not have been entirely neutral when prioritising from an organisational perspective. It is probably difficult to push the self-perspective aside, so there is a risk that the self-perspective has coloured the organisational perspective in the prioritisations. If this is the case, it is likely that the difference between the perspectives as reported in this paper is in reality larger. A solution for reducing this threat would have been to let some participants prioritise from one perspective only, and the others from the other perspective only. However, we chose not to do this due to the small sample size.

A threat to conclusion validity is that the participants may have been disturbed when doing the prioritisation. As they prioritised individually in their work place, we could not control this threat. We do not believe it to be very significant, though.

## 6. CONCLUSIONS

In this paper, we have presented results from a study where impact analysis issues were prioritised by professional software developers from two different perspectives. The *organisational* perspective was used to get an idea of which issues were critical for the organisation, while the *self*-perspective was used to find issues critical for the software developers themselves.

We have compared the two perspectives, and shown a multi-perspective visualisation of issues that allowed us to discuss three classes of issues: first-priority, second-priority and low-priority. The priority classes serve as a means to optimising a process improvement effort, as focus should be put mainly on the first-priority issues. These are issues that are critical both from an organisational perspective (i.e., with respect to

what is best for the organisation) and from a self-perspective (i.e., with respect to what is best for an individual software developer).

Furthermore, visualising issues in a multi-perspective chart (see Figure 1) gives quick feedback on the relative importance of issues, i.e. how the issues relate to each other with respect to criticality.

We have seen that the characteristics of an issue seem to affect how it is prioritised from the two different perspectives—issues that have a holistic impact on the organisation, for example regarding the process or the development of the right product, are typically perceived as critical from the organisational perspective. Conversely, issues that are related to the execution of specific tasks or activities, for example analysis of change impact, are seen as critical from the self-perspective.

A number of possible improvements for mitigating the issues that were seen as first-priority issues in the study have been identified. In an improvement effort, these improvements should be combined, although the cost aspect needs to be taken into consideration, to maximise the outcome with respect to mitigation of existing issues. The issues and improvements presented in this paper are of generic nature, and should apply to other companies than the studied one. Thus, they could be used in future research about impact analysis as a change management activity. We want to stress, however, that the improvements are only proposals at the moment—they have not been evaluated in any way.

In summary, we want to focus on two results:

(1) Studying issues from multiple perspectives is rewarding and entails certain benefits from a process improvement point of view. To answer our research question, it is clear that the perspective does affect the prioritisation of issues with impact analysis, though in our case less than we initially had expected. Furthermore, the spread of issues between the perspectives allows us to get an idea of to what extent people's personal agendas align with what is best for the organisation.

(2) The division (and associated visualisation) of any items into priority classes, based on their weights in a multi-perspective prioritisation, seems to be a good way of optimising a process improvement effort. The approach enables a certain amount of streamlining, as focus can be put primarily on the items in the high-priority class.

## 7. 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>).

## 8. REFERENCES

- [1] V. R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørumgård, and M. V. Zelkowitz. The empirical investigation of perspective-based reading. *Empirical Software Engineering: An International Journal*, 1(2):133–164, 1996.

- [2] S. A. Bohner and R. S. Arnold. *Software Change Impact Analysis*. IEEE Computer Society Press, 1996.
- [3] R. Conradi and T. Dybå. An empirical study on the utility of formal routines to transfer knowledge and experience. In *ESEC/FSE-9: Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 268–276, New York, NY, USA, 2001. ACM Press.
- [4] A. Finkelstein and I. Sommerville. The viewpoints FAQ. *Software Engineering Journal*, 11(1):2–4, 1996.
- [5] T. Hall and D. Wilson. Views of software quality: a field report. *IEE Proceedings Software*, 144(2):111–118, 1997.
- [6] P. Jönsson and C. Wohlin. Understanding impact analysis: An empirical study to capture knowledge on different organisational levels. In *Proceedings of the Intl. Conf. on Software Engineering and Knowledge Engineering*, pages 707–712, Taipei, Taiwan, July 2005.
- [7] J. Karlsson and K. Ryan. A cost-value approach for prioritizing requirements. *IEEE Software*, 14(5):67–74, 1997.
- [8] D. Karlström, P. Runeson, and C. Wohlin. Aggregating viewpoints for strategic software process improvement—a method and a case study. *IEE Proceedings Software*, 149(5):143–152, 2002.
- [9] D. Leffingwell and D. Widrig. *Managing Software Requirements—A Unified Approach*. Addison Wesley, 1999.
- [10] M. Lindvall and K. Sandahl. How well do experienced software developers predict software change? *Journal of Systems and Software*, 43:19–27, 1998.
- [11] S. L. Pfleeger. *Software Engineering—Theory and Practice*. Prentice-Hall, intl. edition, 1998.
- [12] B. Regnell, P. Runeson, and T. Thelin. Are the perspectives really different?—further experimentation on scenario-based reading of requirements. *Empirical Software Engineering*, 5(4):331–356, 2000.
- [13] C. Robson. *Real World Research*. Blackwell Publishing, 2002.
- [14] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, 2000.
- [15] S. Zahran. *Software process improvement: practical guidelines for business success*. Addison-Wesley, 1998.

## APPENDIX

### A. INTERVIEW QUESTIONS

The list below is a summary of the high-level topics and associated questions that made up the interview instrument. Text within square parentheses is provided for some questions to exemplify their scope/direction.

Note that the questions have been translated from Swedish. We have, however, tried to preserve the main point of the original Swedish phrasing.

It should also be mentioned that, while some questions may seem very generic, they have an implied “at your company” part. For instance, the question about impact analysis accuracy refers to the accuracy of analyses in the company where the interviews were conducted.

- Impact analysis process
  - What do the change request and impact analysis processes look like?
  - When is impact analysis performed; when do change requests appear in the project?
  - What is impact analysis based on? [*models, knowledge, traceability etc.*]
  - Which are the consequences of improper impact analysis?
  - What is the perspective when doing impact analysis? [*short-term/long-term*]
- Organisational support
  - Is there a clear process description? [*change request process/impact analysis process*]
  - How is it decided when an analysis is good enough?
  - Is impact analysis supported by tools or methods?
  - Is impact analysis mainly a project or a product activity?
- Uses
  - Which potential uses does impact analysis have?
- Issues
  - Which potential issues are associated with performing impact analysis?
- Measurement/feedback
  - Is it possible to measure the quality of the impact analysis process?
  - Is it possible to measure the quality of a specific analysis?
  - How accurate do you believe impact analysis generally is?
- Traceability
  - Which artefacts is it necessary to trace between?
  - Which artefacts would it be convenient to trace between?
  - Between which artefacts does the process prescribe traceability?