A Descriptive Classification for End User-Relevant Decisions of Large-Scale IT Projects

Ulrike Abelein, Barbara Paech
Institute of Computer Science, University of Heidelberg, Im Neuenheimer Feld 326, 69120 Heidelberg, Germany
{abelein, paech}@informatik.uni-heidelberg.de

Abstract—Large-scale IT projects with traditional development methods are still very common in practice. These projects mostly involve the end user in the beginning and at the end of the development. However, there are also user-relevant decisions in the phases between. Thus, it is important to investigate what decisions are made and which of them are user-relevant. Thus we suggested in our previous work a preliminary classification based on the TORE method to structure decisions.

In this paper, we validate this classification and collected exemplary user-relevant decisions by experts in large-scale IT projects. As part of our research in user-developer communication, we conducted an interview series with twelve experts. The interviews confirmed that our previously suggested classification is comprehensive and helpful to structure decisions and revealed several amendments. The examples given by the experts enabled us to collect a comprehensive list of end user-relevant decisions, and thus lead to our descriptive classification.

Index Terms— User-Developer Communication, User Participation, User Involvement, Software Development, Expert Interviews

I. INTRODUCTION

User participation and involvement (UPI) are widely studied and many empirical studies revealed that an increase in UPI in software (SW) development increases system success [1], [2]. One benefit of UPI is improved SW quality due to more precise requirements. Another advantage is to prevent the development of expensive features that users will not or cannot use. In addition, users have a more positive attitude towards the resulting system, which enables them to use the system more effectively. Furthermore, increased UPI in decisions about the SW leads to a more democratic organizational culture [1], [3]. Additionally, many methods, e.g. agile SW development, aiming at enhancing UPI suggest to increase the communication between users and developers [3], [4]. However, many large-scale IT projects still use traditional project management and SW development methods, such as the waterfall model, with a low level of UPI and communication [5]. The advantages of these traditional methods are high stability and clear agreements. Nevertheless, the drawbacks are waiting periods on the business side due to long development cycles starting after the requirement definition until the system validation. This waiting period can lead to different undesirable phenomena [6]. First, the users do not feel integrated in the project. Second, the end users do not recognize their requirements in the acceptance phase (either due to many transformations or a long time span). Third, the user will have a low acceptance of the system and a low motivation to participate in large-scale IT projects. We think that within projects using traditional SW development methods, many user-relevant decisions are made in the design and implementation phases and that it is important to enhance communication between users and IT personnel in those phases to prevent the mentioned phenomena. To ensure a helpful communication, we defined two research questions (RQ):

• RQ 1 – What decisions made by IT project members are relevant to end users?
• RQ 2 – How good is the suggested classification of user-relevant decisions?

We argue that a user-relevant decision becomes a trigger point to start communication. Therefore we will use the term trigger point for user-relevant decision within this paper. So far not many researchers studied what user-relevant decisions are being made during the design and implementation phases and when it is useful to trigger communication with users. Thus, in our previous work, we developed a classification for trigger points based on the Task-Oriented Requirement Engineering (TORE) method [7], [8]. We build on this approach and validate the classification with experts in the present work. To answer the research questions, we did an interview series, in which we conducted semi-structured interviews with twelve experts in large-scale IT projects. This paper presents the results of one part of the interviews. We collected a list of 81 exemplary decisions. The validated classification together with the exemplary decisions forms the descriptive classification. The paper is structured as follows. In Section II, we briefly present the classification developed in our previous work. In Section III, we explain the method of the interview series. In Section IV, we report on the results by presenting the descriptive classification and discuss the implications of the results. We conclude with future research plans in Section V.

II. BACKGROUND

Most research on UPI focuses either on the early development phases, e.g. elicitation of user needs, or on the final stages of the project, e.g. on the user acceptance test [9]. We assume that in large-scale IT projects, using traditional development methods, there is a need for enhanced user-developer communication focusing on the translation process from user to system requirements. Thus, in our previous work, we presented ideas on how to enhance user-developer communication in large-scale IT projects [7]. We developed a classification of trigger points which is based on decisions defined in TORE [10]. TORE defines 16 different explicit or
We conducted an interview series with twelve experts in large-scale IT projects from Oct. until Dec. 2012. The first interview was used as a prototype interview, in order to refine the questionnaire and estimate a time frame. We used qualitative interviews, which is the most important data gathering tool in qualitative research [11]. The interviews were semi-structured, i.e. we used a questionnaire, but improvised and changed the order of questions if appropriate [11]. Four interviews were done in person; the other eight interviews were conducted via telephone. The average time for the complete interview was 90 minutes (min. 44, max. 125). All interviews were recorded with the permission of the interviewees and transcribed for analysis purposes. This paper reports on roughly a third of the comprehensive interview questions. The rest of the interviews will be used for other part of our research. Our interview partners are experts in large-scale IT projects. They classified themselves in the domains ‘business’ (1 expert), ‘business and IT’ (6 experts) or ‘IT’ (5 experts). In order to get a broad collection of examples, we chose interview partners with different backgrounds. Six experts are employed by IT or management consultancies, four experts work in internal IT departments of large organizations, and two experts work for SW or IT service providers. On average, the interview partners were involved in six large-scale IT projects (min. 2 projects, max. 15 projects) throughout their careers in various roles (e.g. developer, project manager, consultant), which ensures a wide expertise of all of them. Within the interviews we showed them our proposed classification and explained the abstraction levels and trigger points with the help of one example. From the questionnaire the following interview questions were analyzed in this paper:

1. Do you think the abstraction levels help to identify user-relevant decisions (i.e. trigger points)? Would you add/modify/delete any abstraction levels?
2. Would you add/modify/delete any trigger point category?
3. Do you have examples for trigger points in the categories?
4. Which of the following trigger points have you used in projects to initiate communication with the end user?
5. Which of the following trigger points would you rather not use to initiate communication with the end user?

We conducted the interviews in a semi-structured setup, therefore the questions 1 to 3 were asked explicitly. The questions 4 and 5 were answered indirectly. That means, if the interview partner was able to find an example for a trigger point question 4 was considered true otherwise we assume that question 5 was considered true.

IV. RESULTS AND DISCUSSION

The results presented in this paper and the interview series are part of our research on a method to enhance user-developer communication in large-scale IT projects. We started with a proposal for a method including a classification for trigger points [7]. In this section, we present the validation of the classification and its extension with examples of the interviews, which together build the descriptive classification.

A. Validation of the Classification

In this section, we report on the results of the interview questions 1 and 2. Overall, nine of the twelve experts (75%) clearly stated that they consider the classification valid and comprehensive. Of the remaining three, one did not comment on the classification, one expert suggested another structure, and the third interview partner had some issues with the project level. This was on account of this expert’s company in which project level decisions are targeted to a central department that is not connected with the users. Thus, we can conclude that a majority of experts validates the classification. With respect to the abstraction levels, one aspect, discussed in several interviews, was if it is reasonable or not to combine the business process and task level into one abstraction level. Four of twelve experts suggested combing them as these two levels are very closely connected. However, two experts argued strongly against it, reasoning that the business process level regards changes of business concerns and the task level represents the system’s perspective. As there was no consent, we decided to keep the original levels. One expert suggested combining the task, domain and interaction levels into one application level. As none of the other experts made a similar suggestion, we neglected this idea. Another expert commented that some decisions are not strictly confined to one level, but rather produce trigger points on several levels, e.g. which technology is used is important on the system level, but also on the project level as it influences costs and timing. Even though this observation is correct, we believe that is useful to have separate abstraction levels in order to support users, IT personnel and project management. Lastly, one interview partner suggested that decisions on the project level should not be communicated towards end user, but rather to a steering committee. This is addressed by our RACI matrix [7] which also suggests to have these decisions approved by the users’ managers. The same interview partner also thinks that trigger points on the system level should not be communicated as the tool stack (i.e. which frameworks and platform to build upon) should be decided. But he also said that this is highly specific to his company. Also the fact, that we identified seven examples in this category shows that there are user-relevant decisions. On the categories of trigger points, the interviews revealed three major suggestions that we integrated in the classification. First, trigger points regarding cost allocation and timing on the project level should not only cover project cost or go live dates, but also include operations cost and timing implications.
<table>
<thead>
<tr>
<th>Event level</th>
<th>Cost allocation (project, and operations)</th>
<th>Timing (project, operations)</th>
<th>Data/ Skills</th>
<th>Project level</th>
<th>Business process level</th>
<th>Task level</th>
<th>Domain level</th>
<th>Interaction level</th>
<th>Technology level</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
<tr>
<td>A: strategic decision from IT to use a two vendor strategy to prevent dependence on one vendor</td>
<td>A: Compensation</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
<td>B: required new hardware (such as new server)</td>
</tr>
</tbody>
</table>

**Table 1: Descriptive Classification**

<table>
<thead>
<tr>
<th>Abst.</th>
<th>Project level</th>
<th>Business process level</th>
<th>Task level</th>
<th>Domain level</th>
<th>Interaction level</th>
<th>Technology level</th>
</tr>
</thead>
<tbody>
<tr>
<td>L: technical experience</td>
<td>B: proposed project teams (often foreseeing their own role in the solution) until they are at a level where they are used to handle defects</td>
<td>A: strate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Second, two experts suggested including a third category organization/skills in the project level. They reported that often decisions have an influence on the organization. Third, two experts suggested to include system interfaces in the interaction level.

B. List of Examples

Interview question 3 was used to collect examples of decisions (i.e. trigger points) from practice. All examples were classified with respect to their abstraction levels and trigger point category thus enabled us to create a descriptive classification, which is presented in Table I. In total, we collected 81 examples from our interviews. We formatted the examples in a schema with context (C) (if named in interview), decision (D) and impact (I). This was used to classify the example in the category where the decision had the highest impact. We assigned the letters A-L to our twelve interview partners and assigned each example to the corresponding letter. The number of examples varies from 2 to 12 per trigger point category. A more detailed look at the examples revealed seven reoccurring topics and discussions:

- **License cost** including the tradeoff discussion open source vs. proprietary SW was named three times.
- **Staffing for tests** is a common discussion topic in projects.
- **Standard central processes** have been named four times.
- **Access rights and automation of approvals** were named by four experts.
- **Manual vs. automated activities** is a common topic to be discussed with end users.
- **Unfeasible user requirements** due to complexity were mentioned four times, and should be discussed with users.
- **Support of end user devices** was named three times.

As described in the research method section, we answered the fourth and fifth interview question indirectly by the fact if an expert was able to name an example or not. A summary of the amount of examples is given in Table I. On the project level, cost allocation was used by nine experts and timing was used by ten. Four experts gave examples for the organization and skills category, but this can be explained as this is a newly integrated category. On the business process and task level nine experts were able to name an example. For the domain level, the feature category is used by all experts and thus seems to be the most important one. The other two categories (to-be activities and domain data) seem to be less important as only six respectively four experts named examples. On the interaction level, decisions on workflows seem to be common, as five examples were named. Surprising to us, the UI category revealed only four examples. Two experts discussed the new category ‘system interfaces’ and therefore it has two examples. Lastly, for the technology level has seven examples.

V. CONCLUSION

In this paper, we reported on a descriptive classification of trigger points. We proposed a classification based on TORE in our previous work [7]. We conducted an interview series with twelve experts to find out what decisions in large-scale IT projects are user-relevant and to extend our classification to a descriptive classification. The expert interviews enabled us to collect 81 examples of trigger points. From these examples we derived seven common discussion topics with end users (see Section IV). Eight of twelve experts considered the suggested classification as valid. The remaining four did not have strong arguments against it, but rather suggested changes. The analysis showed that most trigger points were used by experts, as they were able to name an example. This paper is part of our research on user-developer communication. In the interviews, we discussed the expert’s experience of communication setups in large-scale IT projects as well possible solutions to close the communication gap between end user and developer. In the future, we will report on the other results of the interviews and will devise a method to enhance user-developer communication. Within this method the descriptive classification for user-relevant decisions will be integrated.

ACKNOWLEDGMENT

We would like to thank all experts for their support.

REFERENCES