©2013 IEEE. Reprinted, with permission, from **Abelein, U, Paech B, A Descriptive Classification for End User-Relevant Decisions of Large-Scale IT Projects, Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2013), San Francisco (USA), May 25th, 2013, pp. 137-140**.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Heidelberg's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubspermissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

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

implicit decisions, which are grouped in four abstraction levels. We extended them by the project and the business process level. In addition, we developed a RACI (R–Responsible, A–Approved, C–Consulted, I–Informed) matrix (see [7]) for the involved roles in order to define who should be informed of these decisions. As there is no clear definition of *large-scale IT projects* in research, we use the following definition for this paper and the interviews sessions. Large-scale IT projects have to fulfill at least one of the following characteristics: large amount of users (over 1000 users), rollout of a system in multiple (min. of three) countries or business units, large budget (over 1 million EUR), project duration (min. 1 year).

III. Research Method

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 carriers 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 fixed. 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 I DESCRIPTIVE CLASSIFICATION

Abst. level	Project level			Business process level	Task level	Domain level			Interaction level			System level
Trig. points	Cost allocation (project, and operations)	Timing (project, operations)	Orga./ Skills	Business processes	Responsibility of the users	To-be activities	Features	Domain data	Workflow	User Interface	Syste m Int.	Technology
nterview examples from experts	• D: new system required new hardware I: increase of operating and hardware costs (B) • D: use new base technology, i.e. use of open source SW I: increase of project cost (especially in case of time & material contract), but decrease of license cost (C) • D: support of several application server platforms I: increased maintenance and operation cost (C) • C: travel booking system D: when is point of time to access external vacancy database, e.g. very early or only at the very end of order I: massive influence on operating costs (E) • C: CRM telecommunication contract system D: flexibility of promotions for contracts, i.e. adaptable from the business team or need to change system configuration I: large implementation cost implications (F) • D: no need for proprietary SW, e.g. use of open source enterprise service bus I: reduced license cost (H) • C: management information system D: fix serious performance issue in underlying systems due to missing definition of non-functional requirements I: required complete new infrastructure and thus large investments (L) • D: use of proprietary components, e.g. data bases or servers instead of open source I: changes in licenses and tools cost and cost of internal IT (J) • D: strategic decision from IT to use two vendor strategy to prevent dependence on one vendor I: additional cost but no implication for the end user (K)	• C: very customized HR of standard system D: prolongation of project plans (often foreseen by project team) kept hidden until the last moment before telling users I: System could not be tested before planned to go live (B) • C: large projects with waves of implementation D: IT project teams change orders of features or cut down features especially I: features are available at different times than expected by user (D) • D: no implementation of temporary access rights for proxies in case of vacation and illness, due to complexity I: lead to a delay of one year (D) • D: no detailed data checks, e.g. compulsory date field could have been empty in old system I: lead to serious project delays (D) • C: Telecommunication systems with critical time to market business opportunities D: decide for an interim "quick and dirty" solution I: reduced quality in feature and need to allocate resources for next phase (F) • D: support for old systems end I: prologues project as system needs to be replaced (G) • D: SW from third parties delayed I: system delays due to waiting time on third party system (H) • C: ERP system D: adaptations of original template for some business units I: changes in roll out alters the whole project plan (J) • D: architectural changes, e.g. refactoring, improvement of maintainability I: lead to delays of 3 to 4 months and hard to discuss with end users (K) • D: refactoring phase after first go live I: delays in roadmap (L)	• D: change of implement ation order of feature x and y I: changes in test schedule and thus for testers (i.e. end users) (C) • D: changes in timing with effect on testing and marketing I: require different business expertise skills for project and operations (E) • C: insurance industry (with seasonal business) D: changes in project schedule I: requires early communic ation to all stakeholde r (G) • D: delay in testing I: influences the need for business employees (L)	• C: standard ERP system D: use very efficient standard central dunning process I: change to central instead of local dunning process (B) • D: integrate content management system (CMS) function within larger system I: changed business process as user needed only to use one system instead of two (C) • D: use standard incoming payments process (incl. one chart of accounts) for all subsidiaries of a company I: subsidiaries with different payment process need to change it (E) • C: CRM system in telecommunication, that combined landline and mobile contracts D: missing definition of IT architecture I: business processes need to be altered to enable possibility to see both contracts from one customer (F) • C: information system in insurance industry D: use standard business processes e.g. what is a lead for an insurance sale I: process steps change after an object is a lead (G) • D: use standard process I: two or three departments required to adapt process (H) • D: use different IT architecture at different business units I: prevention of harmonization of companywide business processes (J) • C: processes-efficient industries, e.g. banking D: have two more clicks in process, through introduction of SEPA I: users denied change as this summed up to 2 additional employee in efforts (K) • C: insurance industry D: introduction of PostIdent I: lead to changes in business processes, as users need to include manual steps (L)	• D: centralized invoice process, I: took responsibility of printing and mailing of invoices from branch employees (B) • D: automatic validation checks from legacy health care system cannot be implemented in new system I: end users now need to implement new process to check validity of input, e.g. double entry (D) • D: use own user management instead of single-sign-on solution I: business side needed to maintain 120.000 new users (D) • D: decide that shipping order can pnly be changed (after initial entry) by manager roles I: control of shipping order is no longer with standard user (E) • C: information system in insurance industry that is access by self-employed sales people D: all user have access right to additional customer data I: self-employed sales people need to give away their additional knowledge (G) • D: decide to no longer have the possibility to change document before printing I: user can no longer do final corrections (I) • D: implement new mandatory check in the system if material is really available I: blue collar workers in production need to check in new system (J) • D: define roles that have access right to insert changes in the FAQ part I: not all users can later FAQs (K) • C: Information system in insurance industry D: implement automated approval of risk assessment I: user has no longer responsibility for the check (L)	• D: no testing and communication of performance requirements (not defined in requirements phase) I: printing took 2 minutes instead of milliseconds before (B) • D: new data security regulations prohibited to send passwords via email I: new manual steps from business side was required (C) • C: CRM system in call center D: Automatic assignment of queues I: Employees no longer get to choose (even though it's a limited choice that is relevant for them and their motivation) (G) • D: which steps covered by the system vs. remain manual, e.g. calculations for tax evaluations I: algorithm calculations can have wrong output and require manual correction. This may take longer than initial manual input (H) • D: no implementation of automatic email distribution for mailing list I: activity no longer available for user (K) • D: include manual as dependencies to different back bone systems can lead to not maintainable system for IT, I:user needs to perform manual steps like printing (K)	• D: no implementation of feature for specific user group, e.g. deal calculation for researchers I: researcher cannot calculate deals (A) • C: ERP system in retail industry D: implement auto-disposition of orders to ensure better sales condition, instead of possibility to complete orders every day I: employees in shops were trained to order on specific days. The old feature needed to be implemented additionally (B) • C: ERP system in retail industry D: gift baskets could not be charged by commission, I: gift baskets can only be paid directly at a cashier (B) • D: support of all possible browsers is not realizable I: only use of supported browsers (C) • D: no implementation of transferability to other system e.g. contact in Microsoft Outlook and Apple iPads I: user has to maintain different contacts(E) • D: implementation of only one payment option for additional mobile packages abroad I: user can only use one payment option (F) • C: online web system for sales in insurance D: no implementation of offline functionality I: system not usable as sales employees sometimes have no internet access (G) • D: have direct data base input and no UI for workflow with only 10 cases p. a. I: difficult user experience (H) • D: no support of multi-language or currency support I: all users need to work in one language and currency (I) • C: User requirement requested to know who was at an ATM when it failed D: not possible in system failure I: feature neglected (K) • D: build feature only for online version I: features not available for internal clerk (L) • D: Support only one bank account in system I: End user could only change their main bank account instead of individual bank account per contract (L)	• D: SAP standard system did not allow 20-digites account numbers I: dismissal of SAP for accounting system (E) • D: extend order form with more data, e.g. price or conditions of contract I: more details available for end user (F) • D: input for system parameters (e.g. 'Sterbe-tafel' in Insurance systems) can only be filled with SQL statements I: changes of user experience (H) • C: New products for insurance system D: new data structure and input fields required I: user need to input more data (L)	• C: System in public sector, were users have been trained on new workflows D: roll out without real workflow and use direct entries in data base, as implementation time was too short I: users has different workflow (B) • Standard SAP solution required two clicks instead of one, which did have an influence on resource demand, if 100.000 service employees do the process 10 times a day (D) • C: travel booking system D: decide on order of transactions, e.g. 1. flight, 2. hotel, 3. rental car I: significant influence on rollback effort (E) • D: invoices of contract were included in online portal (instead of PDF attachment in emails) I: user can see whole contract data etc. (F) • D: Implement workflow for Germany power users with an efficient way for data entry I: not usable in the US, where users need a lot of drop down menus (J)	• D: Usage or non-usage of back-button in browser I: Different user experiences (C) • D: implement a smart phone app for payment abroad (instead of usage of normal website) I: better user experience (F) • D: Implement two UIs ond for power users, who want a quick possibility to enter data, and one for end users, e.g. citizens, that need an intuitive UI (H) I: better User experience • D: small UI changes I: need to be communic ated to marketing and internet sales, so they can adapted there business strategies immediate ly (L)	• D: all decisi ons that influe nce other system s in the IT landsc ape I: have implic ations for the busine ss side (E) • C: Call center D: decide how many system s are used in paralle l I: Call center emplo yees need to know how many system s (L)	• D: use end user devices for services employees with GPS tracking I: no communicatio n of that feature lead to the request of workers council to turn out function (B) • D: no support of iPads due to usage of standard calendar package I: user cannot use iPads (C) • D: Upgrade ORACLE systems I: fulfillment of end users' SLAs (F) • D: use specific front end technology e.g. HTML5 I: performance might significantly drop in case of slow internet connection (G) • D: implement data security requirement for tablet PCs I: usage of tablet PC possible (H) • D: usage of a light or a fat client as end user device I: different user experience (J) • D: Support of OSs and/or end user devices, e.g. tablets or switch from blackberry to android devices I: determines with devices can be used by user (L)
Count	9	10	4	9	9	6	12	4	5	4	2	7

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 the 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

[1] M. A. Harris and H. R. Weistroffer, "A New Look at the Relationship between User Involvement in Systems Development and System Success Development and System Success," Communications of the Association for Information Systems, vol. 24, no. 1, pp. 739–756, 2009.

[2] T. McGill and J. Klobas, "User developed application success: sources and effects of involvement," Behaviour & Information Technology, vol. 27, no. 5, pp. 407–422, 2008.

[3] U. Abelein and B. Paech, "Understanding the Influence of User Participation and Involvement on System Success - a Systematic Mapping Study," submitted in Dec. 2012 to JESE

[4] K. L. Hope and E. Amdahl, "Configuring designers? Using one agile project management methodology to achieve user participation," New Technology, Work and Employment, vol. 26, no. 1, pp. 54–67, 2011.

[5] G. B. Alleman, "Agile project management methods for ERP : how to apply agile processes to complex COTS projects and live to tell about it," in Extreme Programming and Agile Methods: XP/Agile Universe, D. Wells and L. Williams, Eds. Springer Verlag, 2002, pp. 70–88.

[6] A. Al-Rawas, S. Easterbrook, U. S. N. Aeronautics, and S. Administration, "Communication problems in requirements engineering: a field study," Cognitives Science Research Paper - University of Suffex CSRP, pp. 1–2, 1996.

[7] U. Abelein and B. Paech, "A proposal for enhancing user-developer communication in large IT projects," in Cooperative and Human Aspects of Software Engineering (CHASE), 2012 5th Int. Workshop on, 2012.

[8] B. Paech and K. Kohler, "Task-driven requirements in object-oriented development," in Doorn, Jorge H. Perspectives on Software Requirements., Boston, MA: Kluwer Academic, 2004. Print., 2004, pp. 1–25.

[9] J. Iivari, H. Isomäki, and S. Pekkola, "The user - the great unknown of systems development: reasons, forms, challenges, experiences and intellectual contributions of user involvement," ISJ, vol. 20, no. 2, pp. 109–117, Mar. 2010.

[10] B. Paech and K. Kohler, "Task-Driven Requirements in Object-Oriented Development," in Perspectives on RE, J. Leite and J. Doorn, Eds. Kluwer Academic Publishers, 2003, pp. 1–25.

[11] M. D. Myers and M. Newman, "The qualitative interview in IS research: Examining the craft," Information and Organization, vol. 17, no. 1, pp. 2–26, 2007.