Eliciting Stakeholder Preferences for Requirements Prioritization

Alexander Felfernig
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
afelfern@ist.tugraz.at

Gerald Ninaus
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
gninaus@ist.tugraz.at

Florian Reinfrank
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b
8010 Graz, Austria
freinfra@ist.tugraz.at

ABSTRACT

Requirements engineering is a very critical phase in software development process. Requirements can be interpreted as basic decision alternatives which have to be negotiated by stakeholders. In this paper we present the results of an empirical study which focused on the analysis of key influence factors of successful requirements prioritization. This study has been conducted within the scope of software development projects at our university where development teams interacted with a requirements prioritization environment. The major result of our study is that anonymized preference elicitation can help to significantly improve the quality of requirements prioritization, for example, in terms of the degree of team consensus, prioritization diversity, and quality of the resulting software components.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Requirements Engineering; D.2.1 [**Requirements/Specifications**]: Requirements Negotiation; H.5 [**Information Interfaces and Presentation**]: Modeling Environments

General Terms

Human Factors, Experimentation

Keywords

Requirements Prioritization, Group Decision Making

1. INTRODUCTION

Requirements Engineering (RE) is the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems [14]. RE is considered as one of the most critical phases in software projects, and poorly implemented RE is one major risk for project failure [8]. Requirements are the basis for all subsequent phases in the development process and high quality requirements are a major precondition for the success of the project [4].

RecSys'12 9th - 13th September, 2012, Dublin, Ireland
Paper presented at the 2012 Decisions@RecSys workshop in conjunction with the 6th ACM conference on Recommender Systems.

Today's software projects still have a high probability to be canceled or at least to significantly exceed the available resources [13]. As stated by Firesmith [5], the phase of requirements engineering receives rarely more that 2-4% of the overall project efforts although more efforts in getting the requirements right result in significantly higher project success rates. A recent Gartner report [7] states that *requirements defects are the third source of product defects (following coding and design), but are the first source of delivered defects. The cost of fixing defects ranges from a low of approximately $70 (cost to fix a defect at the requirements phase) to a high of $14.000 (cost to fix a defect in production). Improving the requirements gathering process can reduce the overall cost of software and dramatically improve time to market.*

Requirements can be regarded as a representation of decision alternatives or commitments that concern the functionalities and qualities of the software or service [1]. Requirements engineering (RE) is then a complex task where stakeholders have to deal with various decisions [11]:

- *Quality decisions*, e.g., is the requirement non-redundant, concrete, and understandable?
- *Preference decisions*, e.g., which requirements should be considered for the next release?
- *Classification decisions*, e.g., to which topic does this requirement belong?
- *Property decisions*, e.g., is the effort estimation for this requirement realistic?

Stakeholders are often faced with a situation where the amount and complexity of requirements outstrips their capability to survey them and reach a decision [3]. The amount of knowledge and number of stakeholders involved in RE processes tend to increase as well. This makes individual as well as group decisions much more difficult.

The focus of this paper will be *preference decisions*, i.e., we want to support groups of stakeholders in the context of *prioritizing software requirements* for the next release. Typically, resource limitations in software projects are triggering the demand of a prioritization of the defined requirements [8]. Prioritizations support software project managers in the systematic definition of software releases and help to resolve existing preference conflicts among stakeholders.

Only a systematic prioritization can guarantee that the most essential functionalities of the software system are implemented in-time [12]. Typically, requirements prioritization is a collaborative task where stakeholders in a software project collaborate with the goal to achieve consensus regarding the prioritization of a given set of requirements. The earlier requirements are prioritized, the lower is the effort of implementing irrelevant requirements and the higher is the amount of available resources to implement the most relevant requirements [12].

Establishing consensus between stakeholders regarding the prioritization of a given set of requirements is challenging. Prioritizations do not only have to take into account business process related criteria but as well criteria which are related to technical aspects of the software. Especially in larger projects, stakeholders need a tool-supported prioritization approach which can help to reduce influences related to psychological and political factors [12]. Requirements prioritization is a specific type of group work which becomes increasingly important in organizations [10].

Prioritization decisions are typically taken in groups but this task is still ineffective due to reasons such as social blocking, censorship, and hidden agendas [10]. *One balancing strategy is to drop or defer low priority requirements to a later release* [12]. In a study conducted at the Graz University of Technology during the course *Object Oriented Analyses and Design*, the stakeholder part of the customer was impersonated by four course assistants. These assistants were not aware of the study settings and had to review the software functionality developed by the different teams. This evaluation did not include a code review. Rather it was supposed to assess the user experience of the product and which important functionality was supported. These important functions were partially defined by the exercise given in the course. The result of this evaluation is represented by a quality value between 0 and 30 credits. The major contribution of this paper is to show how *anonymity* in group decision processes can help to improve the quality of requirements prioritizations.

The remainder of this paper is organized as follows. In Section 2 we provide an overview of the basic functionalities of the INTELLIREQ requirements engineering environment developed at our university to collect preferences and decisions of stakeholders during the course *Object-oriented Analysis and Design* at the Graz University of Technology. In Section 3 we introduce the basic hypotheses that have been investigated within the scope of our empirical study; in this context we also provide details about the study design. In Section 4 we report the major results of our empirical study and show the effect of anonymity on the group consensus, the decision diversity and the output quality. With Section 5 we conclude the paper.

2. INTELLIREQ DECISION SUPPORT

INTELLIREQ is a group decision environment that supports computer science students at our university in deciding on which requirements should be implemented within the scope of their software projects. For this task 219 students enrolled in a course about *Object-Oriented Analyse and Design* at the Graz University of Technology had to form groups of 5–6 members. Unfortunately, it is not possible to evaluate the existing knowledge and experience of

Figure 1: INTELLIREQ Anonymous Preference Presentation: the preferences of users are anonymized by replacing the stakeholder names with the terms "User 1", "User 2", ... , "User n". The order of stakeholders and the assignment to these terms is randomly generated.

the students and the resulting groups. We therefore distributed the resulting groups randomly on the different evaluation pools and assume that the knowledge and experience is equally distributed on each pool. Each group had to implement a software system with an average effort of about 8 man months. Figure 1 shows the anonymized preference presentation of stakeholders in **IntelliReq**.

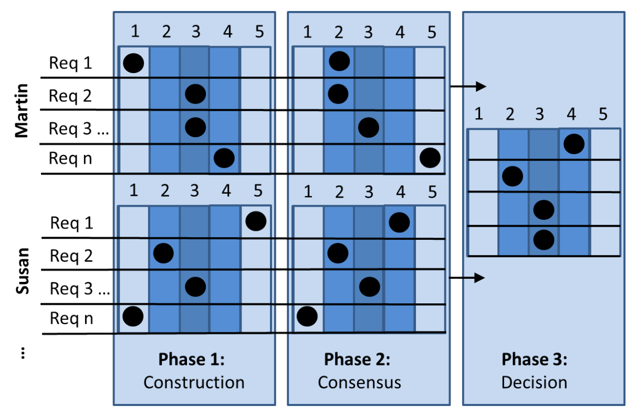

Figure 2: INTELLIREQ Prioritization (Decision) Process. Construction: stakeholders define their initial preferences; Consensus: stakeholders adapt their preferences on the basis of the knowledge about preferences of other stakeholders. Decision: project managers take the final group decision.

In our study, 39 software development teams had to define a set of requirements which in the following had to be implemented. These requirements had to be prioritized and the resulting prioritization served as a major criteria for evaluating the corresponding software components at the end of the project.

The requirements prioritization process consisted of three different phases (see Figure 2) denoted as *construction* (collection of individual stakeholder preferences), *consensus* (adaptation of own preferences, see Figure 3), and *decision* (group decision defined and explained by the project manager). This decision process structure results in about 15.000 stakeholder decisions and 798 corresponding group decisions taken by the team leaders (project managers). On the basis of this scenario we conducted an empirical evaluation with the goal to analyze the effects of supporting anonymized requirements prioritization. The basic settings of this study will be presented in the following section.

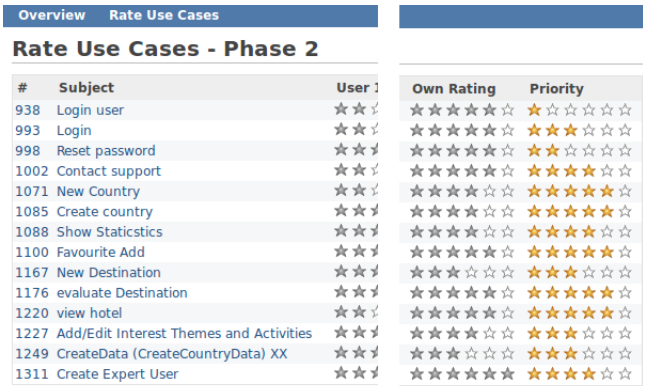

Figure 3: INTELLIREQ Preference Adjustment (Consensus): stakeholders can view their initial preferences and preferences of other stakeholder. With this information stakeholders can adjust their preferences to increase group consensus.

3. EMPIRICAL STUDY

Within the scope of our empirical study we wanted to investigate the impact of *anonymous preference elicitation* on the decision support quality of the INTELLIREQ environment. Consequently, each project team interacted with exactly one of two existing types of preference elicitation interface. One interface (*type 1: non-anonymous preference elicitation*) provided an overview of the personal preferences of team members (stakeholders) where each team member was represented by her/his name. In the second type of interface (*type 2: anonymous preference elicitation*) the preferences of team members were shown in anonymized form where the name of the individual team member was substituted with the terms "User 1", "User 2", etc (see Figure 1). The hypotheses (H1–H8) used to evaluate the decision process are summarized in Figure 4. These hypotheses were evaluated on the basis of the following observation variables.

Anonymous preference elicitation. This variable indicates with which type of prioritization interface the team members were confronted (either summarization of the preferences of the team members including the name of the team members or not including the name of the team members).

Consensus and Dissent. An indication to which extent the team members managed to achieve consensus (dissent) – see the second phase of the group decision process in Figure 2 – is provided by the corresponding variables. We measured the *consensus of a group* on the basis of the standard deviation derived from requirement-specific group decisions. Formula 1 can be used to determine the *dissent* of a group x which is defined in terms of the normalized sum of the standard deviations (sd) of the requirement-specific votings. The group *consensus* can then be interpreted as the counterpart of dissent (see Formula 2). As the consensus is the simple inversion of the dissent, we will only take into account the consensus in the remaining paper.

$$dissent(x) = \frac{\sum_{r \in Requirements} sd(r)}{|Requirements|} \quad (1)$$

$$consensus(x) = \frac{1}{dissent(x)} \quad (2)$$

Decision Diversity. The decision diversity of a group can be defined in terms of the average over the decision diversity of individual users in the consensus phase (see Figure 2). The latter is defined in terms of the standard deviation derived from the decision d_u of a user – a decision consists of the individual requirements prioritizations of the user.

$$diversity(x) = \frac{\sum_{u \in Users} sd(d_u)}{|Users|} \quad (3)$$

Output Quality. The output quality of the software projects conducted within the scope of our empirical study has been derived from the criteria such as degree of fulfillment of the specified requirements. We also weighted the requirements according to their defined priority in the prioritization task. E.g. not including a very high important requirement enormously decreases the quality value. On the opposite, low priority requirements will only have a small impact on the quality value. Therefore, defining a high priority for a requirement which is of minor importance has to be implemented anyway. On the other hand, each group has to implement all important requirements for the user experience and which are important for the functionality. Therefore, the requirements prioritization has a direct impact on the quality value. The quality of the project output has been determined by teaching assistants who did not know to which type of preference elicitation interface (anonymous vs. non-anonymous) the group has been assigned to. These assignments were randomized over all teaching assistants, i.e., each teaching assistant had to evaluate (on a scale of 0..30 credits) groups who interacted with an anonymous and a non-anonymous interface.

Within the scope of our study we wanted to evaluate the following hypotheses.

H1: Anonymous Preference Elicitation increases Consensus. The idea behind this hypothesis is that anonymous preference elicitation helps to decrease the commitment [2] related to an individual decision taken in the preference construction phase (see Figure 2), i.e., changing his/her mind is easier with an anonymous preference elicitation interface. Furthermore, anonymous preference elicitation increases the probability of detecting hidden profiles [6], i.e., increases the probability of exchanging decision-relevant information [9].

H2: Anonymous Preference Elicitation decreases Dissent. Following the idea of hypothesis H1, non-anonymous preference elicitation increases commitment with regard to already taken (and published) decisions. It also decreases the probability of detecting hidden profiles [6] and thus also decreases the probability of high-quality decisions (see H3).

H3: Consensus increases Decision Diversity. As a direct consequence of an increased exchange of decision-relevant information (see Hypothesis H1), deep insights into major properties of the decision problem can be expected. As a consequence, the important differentiation between important, less important, and unimportant requirements with respect to the next release [3] can be achieved.

H4: Dissent decreases Decision Diversity. From less exchange of decision-relevant information we can expect a lower amount of globally available decision-relevant information. As a consequence, the differentiation between important, less important, and unimportant requirements is a bigger challenge for the engaged stakeholders.

H5: Consensus increases Output Quality. From Hypothesis H3 we assume a positive correlation between the degree of consensus and the diversity of the group decision. The diversity is an indicator for a meaningful triage [3] between important, less important, and unimportant requirements.

H6: Dissent decreases Output Quality. In contrary, dissent leads to a lower decision diversity and – as a consequence – to less meaningful results of requirements triage.

H7: Decision Diversity increases Output Quality. Group decision diversity is assumed to be a direct indicator for the quality of the group decision. With this hypothesis we want to analyze the direct interrelationship between prioritization diversity and the quality of the resulting software.

H8: Anonymous Preference Elicitation increases Output Quality. Finally, we want to explicitly analyze whether there exists a relationship between the type of preference elicitation and the corresponding output quality.

4. STUDY RESULTS

We analyzed the hypotheses (H1–H8) on the basis of the variables introduced in Section 3.[1] We used a Mann-Whitney U-test if the examined data set is not normal distributed (H1,H2) and the t-test if the data set is normal distributed (H8). The correlations (H3 – H7) are calculated with Pearson correlations (normal distribution) and with the Spearman's rank correlations (no normal distribution).

H1. The degree of group consensus in teams with anonymous preference elicitation is significantly higher compared to teams with non-anonymous preference elicitation (Mann-Whitney U-test, $p < 0.05$). An explanation model can be the reduction of commitment [2] and a higher probability of discovering hidden profile information which improves the overall knowledge level of the team.

H2. Group dissent is an inverse function of group consensus and – as a consequence – teams with non-anonymous preference elicitation have a significantly higher dissent (Mann-Whitney U-test, $p < 0.05$). In this context, non-anonymous preference elicitation can lead to higher commitment with regard to the orginially articulated preferences.

[1]We are aware of the fact that *dissent* is the inverse function of *consensus*, however, for reasons of understandability we decided to explicitly include *dissent* as a decision variable.

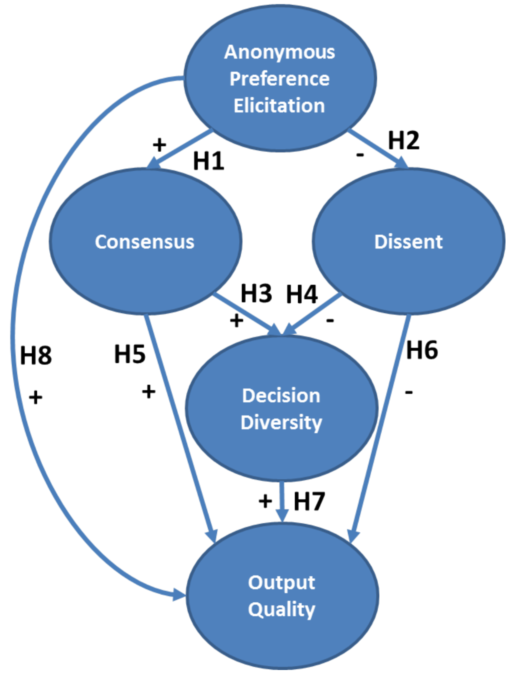

Figure 4: Hypotheses defined to evaluate the INTELLIREQ Decision Support.

H3. There is a positive correlation between the group consensus and the corresponding decision diversity (correlation 0.523, $p < 0.01$). More group discussions can lead to a higher level of relevant knowledge about the decision problem. In the following this can lead to a development of a deeper understanding of the need of requirements triage [3] which leads to a higher degree of decision diversity.

H4. Dissent is an inverse function of group consensus – the higher the dissent, the lower the corresponding decision diversity (correlation -0.523, $p < 0.01$). A lower degree of group decision diversity (prioritization diversity) can be explained by a lower degree of decision-relevant knowledge.

H5. Consensus in group decision making increases the output quality (correlation 0.399, $p < 0.01$). An overlap in the personal stakeholder preferences can be interpreted as an indicator of a common understanding of the underlying set of requirements. This leads to a better prioritization and a higher quality of the resulting software components.

H6. The hypothesis can be confirmed (correlation -0.399, $p < 0.01$), i.e., there is a negative correlation between group dissent and the corresponding output quality.

H7. In our analysis we could detect a positive correlation between group decision diversity (diversity of prioritization) and the corresponding output quality (correlation 0.311, $p < 0.01$). Decision diversity can be seen as an indicator of a reasonable triage process and reasonable prioritizations result in higher-quality software components.

H8. Groups with anonymous preference elicitation performed significantly better compared to groups with a non-anonymous preference elicitation (independent two-sample t-test, $p < 0.05$).

5. CONCLUSIONS

Requirements prioritization is an important task in software development processes. In this paper we motivated the application of requirements prioritization and discussed issues related to the aspect of anonymizing group decision processes in requirements prioritization. The results of our empirical study clearly show the advantages of applying anonymized preference elicitation, for example in terms of higher-quality software components, and can be seen as a step towards a more in-depth integration of decision-oriented research in requirements engineering processes.

6. REFERENCES

[1] A. Aurum and C. Wohlin. The fundamental nature of requirements engineering activities as a decision-making process. *Information and Software Technology*, 45(14):945–954, 2003.

[2] R. Cialdini. The science of persuasion. *Scientific American*, 284:76–81, 2001.

[3] A. Davis. The art of requirements triage. *IEEE Computer*, 36(3):42–49, 2003.

[4] A. Felfernig, C. Zehentner, G. Ninaus, H. Grabner, W. Maaleij, D. Pagano, L. Weninger, and F. Reinfrank. Group decision support for requirements negotiation. In *Advances in User Modeling*, pages 105–116, 2012.

[5] D. Firesmith. Prioritizing requirements. *Journal of Object Technology*, 3(8):35–47, 2004.

[6] T. Greitemeyer and S. Schulz-Hardt. Preference-consistent evaluation of information in the hidden profile paradigm: Beyond group-level explanations for the dominance of shared information in group decisions. *Journal of Personality & Social Psychology*, 84(2):332–339, 2003.

[7] G. Group. Hype cycle for application development: Requirements elicitation and simulation. 2011.

[8] H. Hofmann and F. Lehner. Requirements engineering as a success factor in software projects. *IEEE Software*, 18(4):58–66, 2001.

[9] A. Mojzisch and S. Schulz-Hardt. Knowing other's preferences degrades the quality of group decisions. *Journal of Personality & Social Psychology*, 98(5):794–808, 2010.

[10] A. Pinsonneault and N. Heppel. Anonymity in group support systems research: A new conceptualization, measure, and contingency framework. *Journal of Management Information Systems*, 14:89–108, 1997.

[11] B. Regnell, B. Paech, C. Aurum, C. Wohlin, A. Dutoit, and J. ochDag. Requirements means decision! In *1st Swedish Conf. on Software Engineering and Practice (SERP'01)*, pages 49–52, Innsbruck, Austria, 2001.

[12] K. Wiegers. First things first: Prioritizing requirements. *Software Development*, 1999.

[13] D. Yang, D. Wu, S. Koolmanojwong, A. Brown, and B. Boehm. Wikiwinwin: A wiki based system for collaborative requirements negotiation. In *HICCS 2008*, page 24, Waikoloa, Big Island, Hawaii, 2008.

[14] P. Zave. Classification of research efforts in requirements engineering. *ACM Computing Surveys*, 29(4):315–321, 1997.