Institute of Information Systems and Marketing (IISM)
Kaiserstraße 89-93
76133 Karlsruhe – Germany
http://iism.kit.edu

Karlsruhe Service Research Institute (KSRI)
Kaiserstraße 89
76133 Karlsruhe – Germany
http://ksri.kit.edu

WE SEE WE DISAGREE: INSIGHTS FROM DESIGNING A COOPERATIVE REQUIREMENTS PRIORITIZATION SYSTEM

Research in Progress

Rietz, Tim, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, tim.rietz@kit.edu

Schneider, Franziska, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, franziska.ma.schneider@gmail.com

Abstract

Information systems development is driven by a variety of stakeholders – each with specific requirements. Modern agile development methods, like Scrum, allocate the vital step of prioritizing requirements to dedicated roles like the product owner. However, this can create a bottleneck and may lead to misunderstandings and conflicts between stakeholders. Enabling stakeholders to cooperatively prioritize requirements frees up product owners while involving stakeholders more closely in a crucial development step, strengthening their ties to the final system. Therefore, stakeholders must form a mutual and shared understanding of requirements. This research in progress utilizes the design science research methodology to propose design principles for a cooperative requirements prioritization system using the MuSCoW method – which classifies requirements in four categories. By transferring the theory of shared understanding to the field of requirements prioritization, we derive design principles for cooperative requirements prioritization systems. We enable a group of heterogeneous stakeholders to identify and discuss differences and come to a mutually-agreed prioritization decision. We introduce design features that instantiate these principles and present promising results of an initial pre-test. This research in progress contribute to information systems development by introducing principles for the under-researched, but important class of requirements prioritization systems.

Keywords: Requirements Prioritization, Cooperative System, MuSCoW, Design Science Research

1 Introduction

The prioritization of software requirements plays a crucial role in successfully developing information systems (IS) (Babar et al., 2015; Perini, Susi and Avesani, 2013). Limited budgets, time, and human resources constrain software projects, making it crucial to identify the core requirements of stakeholders (Duan et al., 2009; Perini, Susi and Avesani, 2013). In agile development projects, usually, a single person performs the requirements prioritization process, e.g., a product owner, who then involves further stakeholders if necessary (Rubin, 2012). Being responsible for the prioritization as part of the release planning process, these persons quickly become bottlenecks, as involving diverse stakeholders – developers, customers, and end-users – is time-consuming and yields high coordination and transaction costs (Robertson and Robertson, 2013; Rubin, 2012; Kukreja, 2013). As such, one of the most common problems in agile teams is the limited time of product owners (Bergsmann, 2014; Ramesh et al., 2010). Enabling stakeholder to discuss their perception and prioritize requirements in a cooperative process, without the need to involve a product owner in the process, has the potential to open up this bottleneck. At the same time, involving stakeholders more closely into the decision-making process might create a democratization of the process by forming stronger bonds between decisions (i.e., the prioritization), and deciders (i.e., the stakeholders) (Kaner et al., 2014; Wilson, 2003). These issues have been observed, e.g., in a software development company, where the product owner (PO), responsible for several teams,

worked part-time. The stakeholders, such as the PO, developers, management, and customers, lacked a mutual understanding of the requirements, as customers were only involved selectively in the prioritization. At the same time, the advantage-seeking behavior of the company-internal stakeholders drove most decisions. Furthermore, stakeholders were physically separated by multiple locations, resulting in a lack of personal contact and misunderstandings of requirements.

In such cases, a mutual and shared understanding of requirements between stakeholders is vital to successful cooperation (Tan, 2015; Hoffmann, Bittner and Leimeister, 2013). While stakeholders usually do not need to be experts in each other's fields, they must be able to share and integrate their knowledge bases (Kleinsmann, Buijs and Valkenburg, 2010). When involving stakeholders with various backgrounds, building a shared understanding remains challenging (Hoffmann et al., 2013), due to different roles, responsibilities, vocabulary, and expertise (Busetta et al., 2017; Hoffmann et al., 2013; Duan et al., 2009). With this study, we follow the call of Hoffmann et al. (2013) for further research on shared understanding while transferring insights from previous research to the field of requirements prioritization. Following the design science research (DSR) approach, we formulate the following research question (RQ):

RQ: *Which design principles should a cooperative requirements prioritization system follow to increase shared understanding among stakeholders?*

This research in progress contributes requirements (R) and design principles (DPs) for a cooperative requirements prioritization system (CRPS), enabling stakeholders with various perspectives and roles, personal knowledge, and expertise to cooperatively discuss and decide on a mutually-agreed prioritization. The system's focus lies in increasing the shared understanding among stakeholders to reduce disagreement and reach a consensus on requirement priorities. We derive these principles from requirements that are extracted from issues (I) as reported in relevant literature in the domains of firstly, requirements prioritization and upon identifying it as a central issue, shared understanding. Furthermore, we present results from a pre-test of our prototype in a laboratory with 24 participants, meant to evaluate DPs initially as well as to test the experiment procedure.

2 Conceptual Foundations

2.1 Requirements Prioritization

Requirements prioritization comprises the identification of core requirements to inform the development of a product release schedule (Bergsmann, 2014; Duan et al., 2009). Alkandari and Al.-Shammeri (2017) propose the following steps for a successful prioritization process: (1) determine stakeholders, (2) select a list of requirements that are subject to the prioritization, (3) define prioritization criteria, and (4) select a prioritization technique. However, decisions about priorities commonly cause conflicts between stakeholders, fed by various backgrounds, and incomplete information (Robertson and Robertson, 2013). Stakeholders often lack a shared understanding of the requirements (Ramesh et al., 2010), and pursue different goals, which complicates reaching an agreement (Robertson and Robertson, 2013). Nevertheless, it is crucial to include heterogeneous project stakeholders with different perspectives, skills, and roles (Busetta et al., 2017). The inclusion of all relevant stakeholders is especially crucial as unaccommodating vital stakeholders' interests could lead to project failure (Kolfschoten, Briggs and de Vreede, 2009). Amongst the numerous requirements prioritization methods, such as Ranking and Cumulative Voting (Ramzan et al., 2011), this study focuses on the MuSCoW method of prioritizing requirements through classification. While the "best" prioritization technique depends on the situation (Vestola, 2010), MuSCoW is a wide-spread method, judging by the number of citations and the reported utilization (Achimugu et al., 2014). MuSCoW classifies requirements into "MUST have" - requirements which are crucial for the success of the entire project and not negotiable; "SHOULD have" - preferable requirements; "COULD have" - requirements that would be desirable but are less important; and "WON'T have" - requirements which are not going to be implemented in the current project (Achimugu et al., 2014). Previous work on tool support for CRP investigated voting mechanisms (Park, Port and Boehm, 1999; Gruenbacher, 2000) and gamification (Busetta et al., 2017). However, the proposed tools regard

collaboration as a process where stakeholders only express preferences while a single decision-maker prioritizes. No real-time discussion or (re)negotiation is possible in these systems. Research that considers stakeholder discussion as part of the prioritization process argues that conflicting opinions concerning the importance of requirements should not be averaged out by a voting scheme that returns only one value (Cleland-Huang and Mobasher, 2008). With this study, we follow their argumentation by proposing a design for a CRPS that enables a synchronous discussion to establish a shared framework of understanding (Kaner et al., 2014; van den Bossche et al., 2011; Wilson, 2003). Therein, we extend the existing literature by exploring and evaluating designs for a system supporting synchronous requirements prioritization with multiple stakeholders. A real-time discussion has the advantage of not requiring averaging mechanisms to come to a mutual decision as a group and enables stakeholders to negotiate ratings.

2.2 Shared Understanding in Cooperative Decision-Making

People tend to support most what they helped to create – making it essential to involve stakeholders in the decision-making process to strengthen commitment and responsibility (Kaner et al., 2014; Wilson, 2003). A key challenge in this process is the lack of mutual and shared understanding among stakeholders, which creates disagreement about priorities (Kolfschoten et al., 2009; Ramesh et al., 2010). Differences in perspectives, knowledge, or vocabulary are barriers to forming a shared understanding (Yamaoka et al., 1998). For example, sharing knowledge between IT and business professionals, who are trained in different technical languages and interpret issues from their perspectives, quickly becomes very complicated (Pee et al., 2010). Hoffmann et al. (2013) specifically distinguish mutual understanding and shared understanding. Mutual understanding refers to an understanding of each other's disciplines and the recognition of differences in meaning or information. A shared understanding of a group, however, is described as the ability of multiple actors within a group to coordinate their behavior towards a common goal based on mutual knowledge, beliefs, and assumptions on the task, the group, the process or tools (Bittner and Leimeister, 2014). Thus, a *group* needs to build a common ground for *goals*, the work and group *processes* to achieve them, underlying *tasks*, and the knowledge, skills, and abilities other *team members* contribute (Gibson and Cohen, 2003; Windeler et al., 2015). Strongly related to forming shared understanding within groups is the theory of shared mental models (Bittner and Leimeister, 2014). Langan-Fox, Anglim and Wilson (2004) define a mental model as the individual internal (mental) representation of objects, actions, situations, or people. The theory of shared mental models extends this concept to the group level. Cannon-Bowers, Salas and Converse (1993) define shared mental models as knowledge structures held by team members that enable the formation of accurate explanations and expectations for a task, as well as the coordination of actions and reactions to changes and other team members. We distinguish two types of shared mental models. *Taskwork* mental models focus on performing tasks as a team and relate to understanding different goals, interdependencies, complexities, and procedures of accomplishing tasks processes (Cannon-Bowers et al., 1993; Mathieu et al., 2000; Yu and Petter, 2014). *Teamwork* mental models focus on the shared understanding of how a team interacts and relates to different communication patterns, roles, responsibilities as well as to background knowledge of team members like skills or abilities (Cannon-Bowers et al., 1993; Mathieu et al., 2000; Yu and Petter, 2014). Finally, it is necessary to distinguish perceived and measured shared understanding for evaluating cooperative systems (Bittner and Leimeister, 2014). Perceived shared understanding bears the risk of ignoring whether members are aware of incomplete information or conflicts. Measured shared understanding, on the other hand, requires members to articulate or demonstrate their perception, reducing that risk (Jentsch et al., 2014). Windeler et al. (2015) evaluate the technological intervention in distributed teams via self-reported (perceived) shared understanding while Berggren and Johansson (2010) measure shared understanding in command-and-control tasks. Previous work in the field of collaboration engineering has proposed tools that operationalize the theoretical findings on how to foster shared understanding and consensus-building. *MindMerger*, a collaboration process module that provides designers of collaborative work practices with a bundle of activities to solve clarification issues early in a collaborative process (Bittner and Leimeister, 2014). Tofan et al. (2016) propose a process named *GADGET*, aiming to increase consensus when making group architectural decisions (Tofan et

al., 2016). *GADGET* showed promising results for increasing consensus. However, it may be improved by relying on other averaging mechanisms than a voting scheme. While insights from *MindMerger* and *GADGET*, amongst other methods and tools, inform the design principles proposed in this research in progress, we extend the body of knowledge by transferring insights to requirements prioritization, including evaluation in a controlled experimental setting, following the call for research of Bittner and Leimeister (2014).

3 Research Method

This study applied the Design Science Research (DSR) methodology to identify and implement design principles to develop a cooperative prioritization system by following Peffers et al. (2007). Figure 1 presents the larger overall DSR project and indicates steps taken in cycle II as the focus of this research-in-progress.

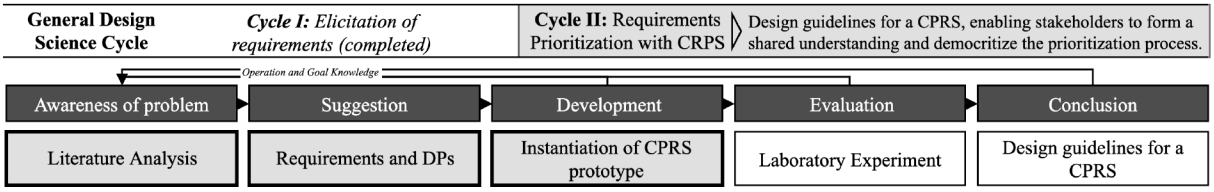

Figure 1. Overall DSR project including cycle I (completed) and cycle II (focus of this research-in-progress), visualization adapted from Kuechler and Vaishnavi (2008)

The prioritization of requirements is the focus of the second cycle of the overall DSR project. This research-in-progress reports initial results from cycle II by deriving four design principles, demonstrating a prototype and presenting the results of an initial evaluation. We build on the results of this study to implement a fully-functional CPRS prototype, which will be evaluated in a laboratory experiment to demonstrate the effects of the proposed design principles for a CPRS. Therefore, we will use the proposed CPRS to prioritize requirements that were collected as part of cycle I, where we conducted elicitation interviews with a wide audience using a chatbot (Rietz and Maedche, 2019).

4 Designing a Cooperative Requirements Prioritization System

4.1 From Awareness of Problem to Suggestion

Requirements prioritization is usually supported by stakeholders from different domains with different perspectives and objectives, resulting in a variety of potentially opposing opinions (Busetta et al., 2017; Duan et al., 2009). These differences between involved stakeholders make it complicated to reach an agreement in requirements prioritization **(I1)**. According to the theory of shared mental models, it is crucial to establish a shared understanding of cross-functional groups, especially when continuous communication is difficult, e.g., due to time pressure (Mathieu et al., 2000). Additionally, stakeholders follow different goals, which causes conflict in requirements prioritization (Robertson and Robertson, 2013), for instance, between developers and testers (Yu and Petter, 2014). Different objectives of stakeholders further aggravated these issues **(I2)**. The theory of shared mental models explains stakeholder conflict with a lack of *teamwork* mental models (Yu and Petter, 2014). Without establishing a shared understanding of teamwork amongst stakeholders, it is more difficult for involved stakeholders to understand their mutual views, perspectives and responsibilities (Kaner et al., 2014; Yu and Petter, 2014), which impedes communication and interaction (Mathieu et al., 2000).

R1: *When multiple stakeholders are involved in prioritization, a CRPS should provide stakeholders with the ability to share information (roles, goals, skills, responsibilities, ...) to build a shared understanding.*

Domain-specific languages or hidden individual knowledge frequently causes an insufficient understanding of each other's requirements (Hoffmann et al., 2013), which is one of the main limitations of existing prioritization techniques (Achimugu et al., 2014) (**I3**). The theory of *taskwork* mental models

stresses the relevance of a shared understanding of goals, interdependencies, and procedures related to the fulfillment of a task as a group (Mathieu et al., 2000). Furthermore, the theory of *shared* taskwork mental models suggests that not having a shared understanding of the meaning and implications of a requirement causes differences in taskwork mental models and thus disagreement (Kolfschoten et al., 2009). Lehtola, Kauppinen and Kujala (2004) observe that decision-makers and developers need to understand better *why* a requirement is vital to a customer or user **(I4)**.

R2: *When differences in understanding requirements appear, a CRPS should provide stakeholders with the ability to identify the conflicting assumptions about the meaning and effect of a requirement.*

The terms 'requirements prioritization' and 'priority' often have subjective meanings and purposes, causing misunderstanding and confusion (Gupta and Gupta, 2018; Lehtola et al., 2004). Requirements prioritization can have ambiguous goals, such as long term benefits of a system/company or the identification of quick wins (Lehtola et al., 2004). Thus, a frequent cause of disagreement is an imprecise understanding of prioritization objectives and criteria in different phases of the project **(I5)**.

R3: *When stakeholders cooperate in requirements prioritization, a CRPS should be able to introduce stakeholders to the task and establish a precise understanding.*

Finally, stakeholders need to interact with each other to share their knowledge and discuss different meanings across domains (Kaner et al., 2014). While a voting based interface can be useful to quickly collect and compare opinions (Park, Port and Boehm 1999), it usually lacks ways of explaining choices, making it hard to come to a consensus. Especially in the context of requirements prioritization, forming consensus is critical, as involved parties commonly have a long-term collaborative interest (Kolfschoten et al., 2009).

R4: *When comparing different opinions in a prioritization process, a CRPS should provide stakeholders with the ability to discuss knowledge and opinions, negotiate the priority of requirements, and collaborate in making a decision.*

We established that shared understanding is influenced by being able to understand the background of team members (**R1**). Knowledge of roles, responsibilities, goals, and skills provides a context for arguments and assessments (Cannon-Bowers et al., 1993). Furthermore, we know that besides being introduced to the (prioritization) task at hand, a team should have a precise understanding of the underlying information (**R3**). Consequently, information about stakeholders would need to be collected and made accessible to all team members. The visualization of collected information can, for example, be based on profile pages (Windeler et al., 2015). Enabling a team to share and access information to understand relationships between each other enables members to spot overlaps or differences in motivation, responsibilities, or backgrounds, which may help to come to a mutually agreed-upon decision (Rosenkranz et al., 2014).

DP1: *Provide the CRPS with a team overview page for stakeholders that visualizes relationships in order for users to gather and mutually share and make sense of information (roles, goals, skills, responsibilities), given that stakeholders work together in a team and are incentivized to share information with team members.*

Secondly, we have established that stakeholders need to be able to spot conflicting assumptions about the meaning and effect of a requirement (**R2**). The transparency of differences in opinions, or in mental models, can help stakeholders to identify underlying problems that may inhibit the cooperative decision-making process (Kolfschoten et al., 2009). Stakeholders may be enabled to make more substantial cases for their arguments in a negotiation when they are aware of the personal views and prejudices of other stakeholders (Mathieu et al., 2000). Considering that stakeholders need to be able to discuss their opinions to fuel a negotiation (**R4**), this process can only come to meaningful results, with a chance of being accepted by as many stakeholders as possible, when the causes of conflicting perceptions are understood (Ramesh et al., 2010, Kaner et al., 2014).

DP2: *Provide the CRPS with an individualized evaluation of requirements and a visual comparison of individual perceptions in order for users to identify and manage conflicting perceptions, given that users do not agree on the meaning and effects of a requirement.*

Thirdly, we have established that stakeholders easily get lost in the process of requirements prioritization due to the misunderstanding of objectives (**R3**). As various criteria (time, pricy, innovativeness, …) drive the prioritization, it is crucial to converge on a set of mutual goals, especially when cooperating (Robertson and Robertson, 2013). The goals need to be introduced and explained to stakeholders to help them form a shared taskwork mental model (Yu and Petter, 2014). As an example, we can look at cooperative online gaming. Here, players receive detailed explanations, or tutorials, outlining how to work together to solve a problem or reach a goal (Kim et al., 2017).

DP3: *Provide the CRPS with a tutorial in order for users to understand the prioritization objectives and relevant criteria of the current project phase and reduce task-related misunderstanding, given that users do not have yet established a clear vision of the task.*

Finally, we have established that cooperative prioritization can only come to a mutual decision when stakeholders have the chance to discuss their opinions (**R4**). Compared to (multiple) rounds of voting, a chatroom as a space for free-form discussion allows participants to elaborate their perspectives and share knowledge, enabling the team to make the best possible decision given the underlying restrictions (Rosenkranz et al., 2014).

DP4: *Provide the CRPS with a shared space for discussion and decision in order for users to explain and negotiate perceptions of requirements, given that users face differences in individual perceptions.*

4.2 Development and Preliminary Evaluation

We defined two design features (DF) to satisfy DP2 and DP4, which were instantiated in a prototype for an initial pre-test, as shown in Figure 2. A CRPS needs to incorporate an additional process step for stakeholders to express an individual evaluation of requirements. Furthermore, a CRPS needs to visualize these individual assessments (**DF1**). Secondly, to allow stakeholders to discuss their assessments, a CRPS needs to include shared space, such as a chatroom, for discussion and cooperative decision-making (**DF2**). We include DF2 in both the control and treatment groups, as it is mandatory to enable the desired interaction. DP1 and DP3 are explicitly not included as part of the pre-test prototype, to reduce the number of treatments required for this initial evaluation of the experimental procedure.

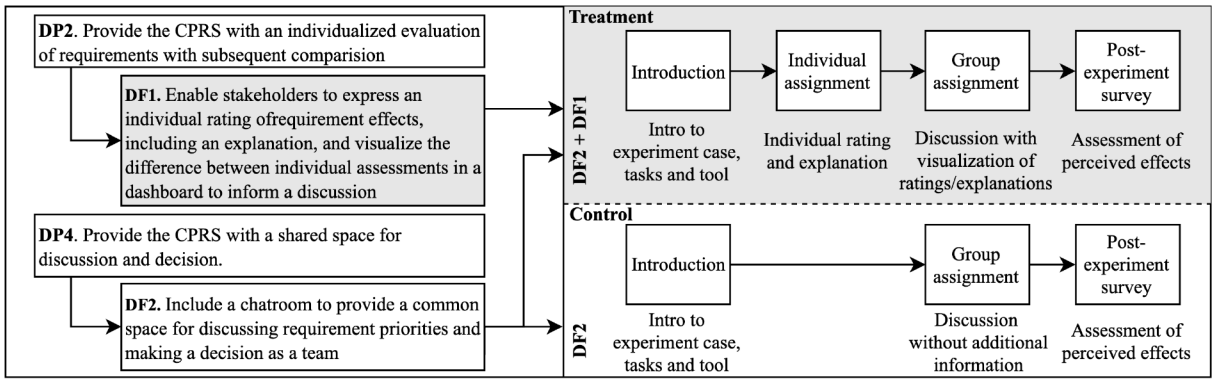

Figure 2. Abbreviated DP2 and DP4 and instantiation into design feature 1 and 2. The right side shows the experimental procedure for treatment and control groups and included design features.

We conducted a between-subjects experiment with 24 participants in two treatments. We randomly assigned participants to groups of three. The experiment contained three stages: An introduction, the interaction with the CPRS, and a post-experiment questionnaire. In the control treatment, we asked participants to prioritize requirements as a group, using a prototype implementing only DF2. We asked the treatment group to complete an individual rating of the requirements before the group phase, using a prototype implementing both DF1 and DF2. A printed description of the experiment case introduced the participants to five requirements for a new app for students. In the individual assignment, we asked participants to prioritize requirements according to their personal preferences. Each requirement had a value function, that determined the added benefit in case of its implementation. In the group assignment,

we asked participants to sort each requirement into one of three categories: Must (implementation of requirement is guaranteed), Should (50% chance of implementation) and Could (25% chance of implementation), each with limited capacity: must (1), should (3) and could (1). After the experiment, the participant's payoff was calculated by determining if a requirement would be met (based on each category's likelihood) multiplied by the value function. Value functions had a form such as *Requirement A has a 60% chance of adding 10 points and a 40% chance of adding 50 points*. Figure 3 shows the group assignment screen in the treatment group. Furthermore, we included multiple characteristics of prioritization in a professional environment to increase the generalizability of the results.

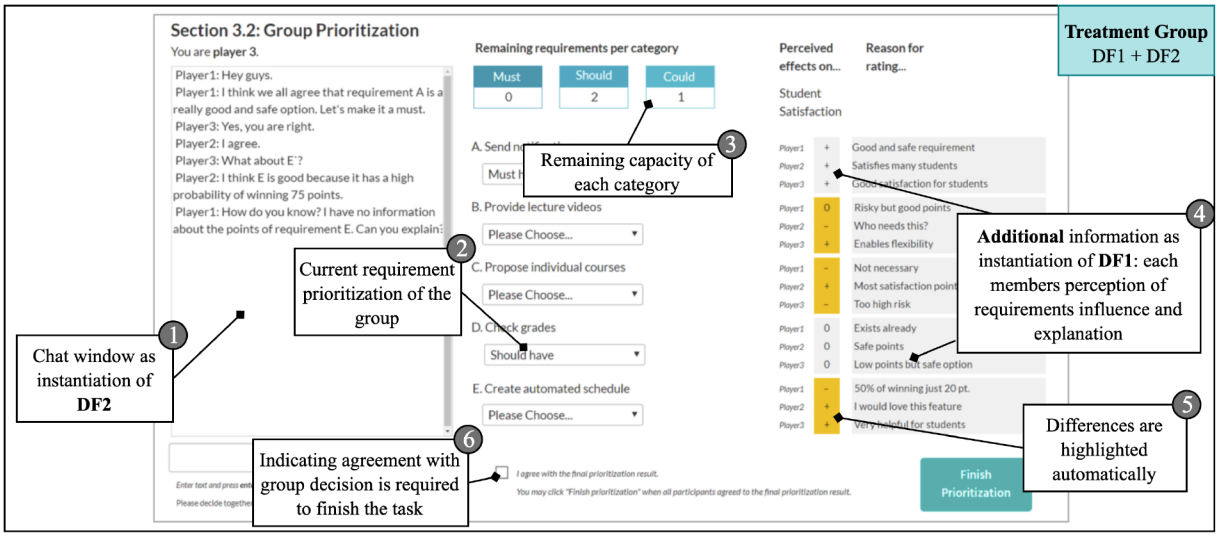

Figure 3. Screenshot of the group assignment in the treatment group. The visualization of the results of the individual assignment and resulting differences on the right side (4 and 5) are not shown in the control condition.

Different understanding of requirements. Stakeholders often lack a shared understanding of requirements leading to disagreement on their importance (Ramesh et al., 2010). In the experiment, differences in understanding and disagreement depend on (1) the participants' perception of a requirements content and its importance for the university app, (2) participants' risk aversion and their perception of probabilities regarding a requirement's value function, and (3) incomplete information (Homan et al., 2007).

Incomplete information. Participants received incomplete information about the value of requirements, as each group member received the value functions of only three of the five requirements. The participants are not explicitly informed about their incomplete information. However, the two remaining value functions are marked with a question mark (?).

Decision with incomplete information. In practice, stakeholders from different domains prioritize requirements according to their knowledge (Busetta et al., 2017). Therefore, within the experiment, participants are asked to select their favorite requirement based on each requirement's value function.

Favoring own requirements. Stakeholders tend to give higher weight to their requirements to improve their benefit (Windeler et al., 2015). Participants receive a bonus to their payoff if they manage to put their favorite requirement into the *must category*, which is limited to one.

Dependence on the outcome. In practice, stakeholders are usually affected by the result of the prioritization. As such, the cooperative prioritization decisions primarily determined each participant's payoff.

Incentivization. We used monetary incentives for participants to create an additional source of conflict. Participants final payoff depended on the group decision as well as the implementation of the favorite requirement (however, only with an influence of 5% on average on the total payoff) that participants chose before entering the group stage.

As measurements, we used a post-experiment survey with four constructs (enjoyment, satisfaction, perceived shared understanding and task conflict), measured on a five-point Likert scale. Furthermore, we

analyzed discussion chatlogs in a double-blinded review process with three reviewers to evaluate shared understanding in a measured form. Therefore, we asked randomly selected reviewers to rate groups in the following categories: (C1) group talked about the content of requirements, (C2) group discussed the value functions of requirements, and (C3) group indicated that they recognized their incomplete information. We utilized ratings for each group to compare perceived and measured shared understanding (cf. Homan et al., 2007).

5 Initial Results and Next Steps

On average, participants answered 7.83 of 9 control questions correctly, indicating a sufficient understanding of the experimental case. Given the pre-tests small sample size (n = 24), conclusions based on a statistical analysis of the data are not reliable, but may serve as a first indicator. For all of these constructs, Cronbach's α and CR were larger than 0.7, demonstrating high internal consistency (Hair et al., 2017). However, using an independent t-test, we did not observe any significant effect of our treatment compared with the control groups based on the self-reported measures. To summarize, the results showed minor differences in enjoyment between control ($M_c = 3.83$, $SD_c = 0.26$) and treatment groups ($M_t = 3.42$, $SD_t = 0.33$), $t(20.53) = 1.33$, $p = 0.2$, $r = .28$; minor differences in satisfaction between control ($M_c = 3.9$, $SD_c = 0.88$) and treatment groups ($M_t = 3.67$, $SD_t = 0.81$), $t(21.98) = 1.19$, $p = 0.24$, $r = .25$; marginal differences in shared understanding between control ($M_c = 3.97$, $SD_c = 0.33$) and treatment groups ($M_t = 3.65$, $SD_t = 0.32$), $t(20.83) = 0.99$, $p = 0.35$, $r = .21$; and minor differences in task conflict between control ($M_c = 2.72$, $SD_c = 1.11$) and treatment groups ($M_t = 3.11$, $SD_t = 0.57$), $t(20.85) = -1.3$, $p = 0.21$, $r = .27$. When analyzing the objective measures however, we observe a lower number of chat messages in control ($M_c = 34$, $SD_c = 21.4$) compared to treatment groups ($M_t = 50$, $SD_t = 22.89$), $t(21.9) = -1.77$, $p = 0.09$, $r = .35$. These findings are in line with the shorter time that participants in control groups ($M_c = 442.02$, $SD_c = 164.89$) took for the task, compared to treatment groups ($M_t = 827.91$, $SD_t = 372.26$).

Subsequently, we continued with the evaluation of the reviewer's ratings for each group in categories 1 - 3. Interestingly enough, in the control treatment, none of the four groups completed C1 - C3. Only one group realized their incomplete information (C3), despite not discussing the requirements as such. Of the four treatment groups, two fulfilled C1 - C3, while all groups at least talked about the content of requirements (C1), despite not realizing the underlying incomplete information (C2). These results are an essential takeaway from the pre-test: While participants in the control group report a higher (in absolute terms) perceived shared understanding, it appears as if these groups primarily did not realize that their understanding was, in fact, different from each other. Every group in the treatment discussed the content of requirements, with half of the groups understanding the reasons for their different perceptions in the individual ranking. Of the control groups, only one group identified the differences, as the team guessed that they might not have the same information from the start. Our results indicate that while team members may not perceive their shared understanding to be better in the treatment group, they demonstrate a better measured shared understanding. However, these findings are limited by the size and profession (students) of our sample. Thus, the results may only serve as an initial indication of the final evaluation. However, by introducing the characteristics of prioritization in a professional environment mentioned above, we attempted to counter the limitations of involving only students in the pre-test. Based on the presented results, our next steps are twofold: firstly, we aim to adjust our experiment design to focus more strongly on the spread in perceived and measured shared understanding. Therefore, we may increase the duration of the conversation and introduce a qualitative interview at the end of each session to improve our understanding of participants' perceptions of the discussion and the results. Additionally, we will include further objective measures to better outline differences in perceived and measured shared understanding between groups, to reduce our reliance on qualitative (chatlog) data (Berggren and Johansson, 2010). Secondly, we will conduct a second pre-test, focusing on DP1 and DP3, to prepare for a large scale laboratory evaluation, in which we compare all four design principles with a more diverse mix of stakeholders. Until then, this research in progress contributes to the body of knowledge by presenting and initially evaluating DPs for a synchronous CRPS.

References

Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. (2014). A systematic literature review of software requirements prioritization research. Information and Software Technology, 56(6), 568–585.

Alkandari, M., & Al.-Shammeri, A. (2017). Enhancing the Process of Requirements Prioritization in Agile Software Development - A Proposed Model. Journal of Software, 12(6), 439–453.

Babar, M. I., Ghazali, M., Jawawi, D. N.A., Shamsuddin, S. M., & Ibrahim, N. (2015). PHandler: An expert system for a scalable software requirements prioritization process. Knowledge-Based Systems (84), 179–202.

Bergsmann, J. (2014). Requirements Engineering für die agile Softwareentwicklung: Methoden, Techniken und Strategien zur erfolgreichen Umsetzung (1. Aufl., neue Ausg). Heidelberg, Neckar: dpunkt.

Berggren, P., & Johansson, B. J. (2010). Developing an instrument for measuring shared understanding. In Proceedings of the 7th International ISCRAM Conference, Seattle, USA.

Bittner, E. A. C., & Leimeister, J. M. (2014). Creating Shared Understanding in heterogeneous work groups - Why it matters and how to achieve it. In Journal of Management Information Systems (JMIS), 31(1), 111–143.

Busetta, P., Kifetew, F. M., Munante, D., Perini, A., Siena, A., & Susi, A. (2017). Tool-Supported Collaborative Requirements Prioritisation. In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), 180–189.

Cannon-Bowers, J. A., Salas, E., & Converse, S. (1993). Shared mental models in expert team decision making. In N. J. Castellan (Ed.), Individual and group decision making: Current issues, 221–246. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

Cleland-Huang, J., & Mobasher, B. (2008). Using Data Mining and Recommender Systems to Scale Up the Requirements Process. In: ULSSIS ’08, Proceedings of the 2nd International Workshop on Ultra-large-scale Software-intensive Systems, 3–6.

Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. Requirements Engineering, 14(2), 73–89.

Gruenbacher, P. (2000). Collaborative requirements negotiation with EasyWinWin. In Proceedings 11th International Workshop on Database and Expert Systems Applications, 954–958.

Gibson, C. B., & Cohen, S. G. (2003). Virtual Teams That Work: Creating Conditions for Virtual Team Effectiveness. Hoboken: Wiley.

Gupta, A., & Gupta, C. (2018). CDBR: A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach. Journal of King Saud University - Computer and Information Sciences.

Hair, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2017). A primer on partial least squares structural equation modeling (PLS-SEM) (Second edition). Los Angeles, London, New Delhi, Singapore, Washington DC, Melbourne: SAGE Publications Inc.

Hoffmann, A., Bittner, E. A. C., & Leimeister, J. M. (2013). The Emergence of Mutual and Shared Understanding in the System Development Process. In J. Doerr & A. L. Opdahl (Eds.), Requirements Engineering: Foundation for Software Quality, 174–189. Springer Berlin Heidelberg.

Homan, A. C., van Knippenberg, D., van Kleef, G. A., & Dreu, C. K. W. de. (2007). Bridging faultlines by valuing diversity: Diversity beliefs, information elaboration, and performance in diverse work groups. Journal of Applied Psychology, 92(5), 1189–1199.

Jentsch, C., Beimborn, D., Jungnickl, C. P., & Renner, G. S. (2014). How to Measure Shared Understanding among Business and IT. In Academy of Management Proceedings, 2014(1), Briarcliff Manor, NY 10510: Academy of Management.

Kaner, S., Toldi, C., Berger, D., Lind, L., & Fisk, S. (2014). Facilitator's guide to participatory decision-making (Third edition). San Francisco, CA: Jossey-Bass a Wiley Brand.

Kim, Y. J., D. Engel, A. W. Woolley, J. Y. T. Lin, N. McArthur and T. W. Malone. (2017). “What makes a strong team? Using collective intelligence to predict team performance in League of Legends.” In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW, 2316–2329.

Kleinsmann, M., Buijs, J., & Valkenburg, R. (2010). Understanding the complexity of knowledge integration in collaborative new product development teams: A case study. Journal of Engineering and Technology Management, 27(1-2), 20–32.

Kolfschoten, G., Briggs, R., & de Vreede, G.-J. (2009). A Diagnostic to Identify and Resolve Different Sources of Disagreement in Collaborative Requirements Engineering. In Proceedings of the International Conference on Group Decision and Negotiation, 100–113.

Kuechler, B. and V. Vaishnavi. (2008). "On theory development in design science research: anatomy of a research project." European Journal of Information Systems, 17(5), 489-504.

Kukreja, N. (2013). Decision theoretic requirements prioritization: A two-step approach for sliding towards value realization. In 2013 35th International Conference on Software Engineering (ICSE), 1465–1467.

Langan-Fox, J., Anglim, J., & Wilson, J. R. (2004). Mental models, team mental models, and performance: Process, development, and future directions. Human Factors and Ergonomics in Manufacturing, 14(4), 331–352.

Lehtola, L., Kauppinen, M., & Kujala, S. (2004). Requirements Prioritization Challenges in Practice. In T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, O. Nierstrasz,. . . H. Iida (Eds.), Lecture Notes in Computer Science. Product Focused Software Process Improvement (Vol. 3009, pp. 497–508).

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. Journal of Applied Psychology, 85(2), 273–283.

Park, J.-W., Port, D., & Boehm, B. (1999). Supporting Distributed Collaborative Prioritization for Win-Win Requirements Capture and Negotiations. In Proc. Int'l Third World Multiconf. Systemics, Cybernetics and Informatics, 578–584.

Pee, L. G., Kankanhalli, A., & Kim, H.-W. (2010). Knowledge Sharing in Information Systems Development: A Social Interdependence Perspective. Journal of the Association for Information Systems, 11(10), 550–575.

Peffers, K., T. Tuunanen, M. A. Rothenberger and S. Chatterjee. (2007). "A Design Science Research Methodology for Information Systems Research." Journal of Management Information Systems, 24(3), 45–77.

Perini, A., Susi, A., & Avesani, P. (2013). A Machine Learning Approach to Software Requirements Prioritization. IEEE Transactions on Software Engineering, 39(4), 445–461.

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. Information Systems Journal, 20, 449–480.

Ramzan, M., Jaffar, A., & Ali Shahid, A. (2011). Value based intelligent requirement prioritization (Virp): Expert driven fuzzy logic based prioritization technique. International Journal of Innovative Computing, Information and Control, 7(3), 1017–1038.

Rietz, T., & Maedche, A. (2019). LadderBot: A Requirements Self-Elicitation System. In Proceedings of the IEEE 27th International Requirements Engineering Conference (RE), pp. 357–362.

Robertson, S., & Robertson, J. (2013). Mastering the requirements process: Getting requirements right (3rd ed.). Upper Saddle River, N.J: Addison-Wesley.

Rosenkranz, C., Vranešić, H., & Holten, R. (2014). Boundary Interactions and Motors of Change in Requirements Elicitation: A Dynamic Perspective on Knowledge Sharing. Journal of the Association for Information Systems, 15(6), 306–345.

Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular agile process. Upper Saddle River, NJ: Addison-Wesley.

Tan, M. (2015). Establishing Mutual Understanding in Systems Design: An Empirical Study. Journal of Management Information Systems, 10(4), 159–182.

Tofan, D., Galster, M., Lytra, I., Avgeriou, P., Zdun, U., Fouche, M. A., de Boer, R., & Solms, F. (2016). Empirical evaluation of a process to increase consensus in group architectural decision making. Information and Software Technology, 72, 31-47.

Van den Bossche, P., Gijselaers, W., Segers, M., Woltjer, G., & Kirschner, P. (2011). Team learning: Building shared mental models. Instructional Science, 39(3), 283–301.

Vestola, M. (2010). "A Comparison of Nine Basic Techniques for Requirements Prioritization." Helsinki University of Technology, 1–8.

Wilson, M. A. (2003). Collaborative decision making: building consensus group decisions for project success. In PMI Global Congress. North America, Baltimore, MD.

Windeler, J., Maruping, L., Robert, L., & Riemenschneider, C. (2015). E-profiles, Conflict, and Shared Understanding in Distributed Teams. Journal of the Association for Information Systems, 16(7), 608–645.

Yamaoka, T., Tsujino, K., Yoshida, T., & Nishida, S. (1998). Supporting mutual understanding in collaborative design project. In Proceedings. 3rd Asia Pacific Computer Human Interaction, 132–137.

Yu, X., & Petter, S. (2014). Understanding agile software development practices using shared mental models theory. Information and Software Technology. (56), 911–921.