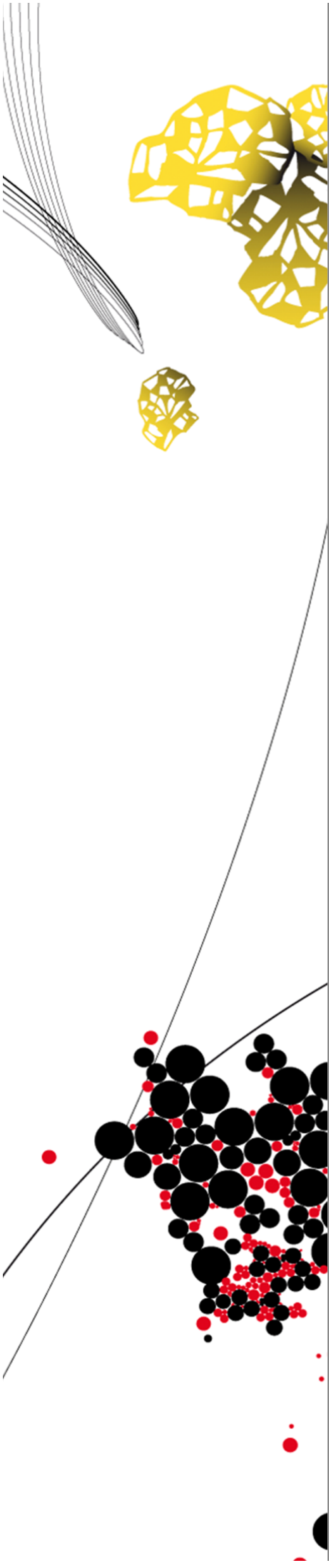


MASTER THESIS

What Project Management System does the Multidisciplinary Team want?

A requirement study to compare the users' wishes with two project management systems.

Renée de Wolf, BSc.

Utrecht, April 10 2017

ELITAC
FEEL THE INFORMATION

UNIVERSITY OF TWENTE.

What Project Management System does the Multidisciplinary Team want?

A requirement study to compare the users' wishes with two project management systems.

UNIVERSITY OF TWENTE.
FACULTY OF BEHAVIOURAL, MANAGEMENT AND SOCIAL SCIENCES (BMS).
HUMAN FACTORS AND ENGINEERING PSYCHOLOGY.

AUTHOR.
Renée de Wolf, BSc.
S1593013

SUPERVISORS UNIVERSITY OF TWENTE.
Dr. M. Schmettow
S.M. Vosslamber, MSc.

SUPERVISOR ELITAC B.V.
B.S. Bicknese, B.Eng.

Utrecht, April 10 2017

Abstract

This qualitative research study was conducted to evaluate the PM systems Jira and Taiga for the company Elitac. This was done by deriving the users' wishes regarding a PM system, giving answer to the research question: *"How should a Project Management System be designed for Elitac to achieve an efficient workflow?"* The study consisted of three parts, of which the first part elicited knowledge about the user needs by conducting nine interviews and a focus group. In total 62 user needs were found. The requirements were prioritized using the MoSCoW technique. Card sorting tests were conducted to uncover what navigation structure Elitac would want to use throughout the PM system. And lastly, a checklist was made used in the third part, helping in evaluating both PM systems Jira and Taiga. Three recommendations are presented, concerning staying with Jira, switching to Taiga, or looking for another PM system altogether. However, the results found in the current study would indicate that the PM system Jira would make a better fit in regards to the users' wishes for Elitac.

Table of Contents

1. Introduction 6
1.1 Project Management Approaches 6
1.2 Project Types 8
1.3 User Centered Design 9
1.4 Requirement Engineering 10
1.5 Company Background 13
1.6 Goal of this study 14
1.7 Structure 14
2. Method 15
2.1 Research Design 15
2.2 Participants 16
2.3 Part 1 Interviews and Focus Group 16
2.4 Part 2 Card Sorting 20
2.5 Part 3 System Evaluation and Comparison 21
3. Results 23
3.1 Part 1 Interviews and Focus Group 23
3.2 Part 2 Card Sorting 31
3.3 Part 3 System Evaluation and Comparison 34
4. Discussion 36
4.1 Reflection results 36
4.2 Implications for practice 39
4.3 Strengths and limitations 41
4.4 Reflection research design 42
4.5 Optional recommendations 44
5. References 46
Appendix A. Informed Consent 51
Appendix B. Interview Protocol 52
Appendix C. Focus Group Protocol 55
Appendix D. Mind Map V2 57
Appendix E. MoSCoW Cards 58

Appendix F. Python code for heatmap 59
Appendix G. Ontology: content-labels throughout the system 60
Appendix H. Checklist Requirements Jira 62
Appendix I. Checklist Requirements Taiga 67

What Project Management System does the Multidisciplinary Team want?
A requirement study to compare the users' wishes with two project management systems.

1. Introduction

The development of embedded systems is a complex endeavor. As Matzler and Hinterhuber (1998) described, one does not only need to meet customers' expectations and reach a high level of customer satisfaction. Also, time is essential, as time to market is becoming increasingly more important. Ensuring a good communication between different disciplines involved in the development helps to circumvent that resources are wasted. And lastly, it is critical to make sure the development process of product development is conducted systematically. Managing a project helps in making the development process more efficient and effective. This Project Management (PM) entails planning, organizing, directing, and controlling of company resources for a relatively short-term objective that has been established to complete specific goals and objectives (Kerzner, 2013). Note that 'short-term' is a definition that varies among different industries. The ISO 21500 by Zandhuis and Stellingwerf (2012) provides the following definition; "PM is the application of methods, tools, techniques and competences to a project". This includes the integration of various phases of the project life cycle, producing deliverables that are measurable, tangible outputs, such as hardware-, software-, or interim deliverables, accomplished through processes (Kerzner, 2013; Zandhuis & Stellingwerf, 2012).

But efficient PM requires more than good planning, like obtaining relevant information in a timely manner, and analyzing and reviewing this (Kerzner, 2013). PM software is a big support in this matter for it helps in managing the development process. But which one among the many different systems does one choose? Multiple factors that have to be taken into account are for example the choice in management approach, the type of project, and no less important; the users' wishes regarding such system.

1.1 Project Management Approaches

According to Špundak (2014), two opposite sides exist concerning project management approaches – the traditional and the agile project management approach. Table 1 shows an overview of the differences between the traditional and agile project management approach.

Table 1.

Main differences between traditional development and agile development (Dybå & Dingsøyr, 2008).

	Traditional development	**Agile development**
Fundamental assumption	Systems are fully specifiable, predictable, and are built through meticulous and extensive planning	High-quality adaptive software is developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Communication	Formal	Informal
Development model	Life-cycle model (waterfall, spiral or some variation)	The evolutionary-delivery model
Desired organizational form/structure	Mechanistic (bureaucratic with high formalization), aimed at large organizations	Organic (flexible and participative encouraging cooperative social action), aimed at small and medium sized organizations
Quality control	Heavy planning and strict control. Late, heave testing	Contentious control of requirements, design and solutions. Continuous testing

Traditional relies on specification-driven methods, such as the waterfall method, using extensive planning, codified processes, massive documentation, and rigorous reuse to make development an efficient and predictable activity (Boehm, 2002; Hoda & Murugesan, 2016). The customer is involved in the beginning and at the end of the project only (Castillo, 2016b).

In the 1990s, alternatives to the traditional approaches emerged, giving rise to agile approaches (Könnölä et al., 2016). These agile methods improve the process flexibility and transparency, making the development more efficient and productive. As case evidence in the study of Könnölä et al. (2016) showed, the interdependencies between work of each developer were better taken into account, product visibility was increased, and communication was improved. This is because in contrast to the traditional approaches, agile management approaches emphasizes continuous design, freezing design features as late as possible, embracing uncertainty and customer interaction, and a flexible scope by being able to respond quickly to changing business requirements, technologies, market conditions, and customer needs (Hoda & Murugesan, 2016; Petersen & Wohlin, 2010; Serrador & Pinto, 2015; Spath, Hermann, Peissner, & Sproll, 2012). Where the traditional approach is for a great deal planned upfront (Špundak, 2014), the agile software development approach is more flexible and can due to the continuous updated documentation adapt to changes throughout the project (Hoda & Murugesan, 2016). The idea is that a high-level framework plan is made, but details become more explicit during the development process (Collyer,

Warren, Hemsley, & Stevens, 2010), for example by developer-tester interaction accompanying the development (Bjarnason, Unterkalmsteiner, Borg, & Engström, 2016). According to Špundak (2014), this iterative approach should help in building a final project scope and a better customer satisfaction. Furthermore, this approach has a greater focus on informal communication (Collyer et al., 2010; Coram & Bohner, 2005). An agile approach example is the scrum method, which is according to Hoda and Murugesan (2016) the most popular agile method. Typically, one has a team of approximately four to 10 people with a 'product owner' (responsible for the return on investment), a 'scrum master' (one responsible for ensuring that the whole team is working well together and follows the agile-scrum methodology), and the developers (Castillo, 2016a). The idea is to iteratively develop a scope of what is to be done by using repetitive periods called 'sprints', and validating and prioritizing the outputs.

1.2 Project Types

Within an organization, each project is unique and differs e.g. in resources used, deliverables provided, etc., which makes that project management is practiced differently in different contexts (Besner & Hobbs, 2012; Zandhuis & Stellingwerf, 2012). The type of project deliverable itself is according to Besner and Hobbs (2012) considered as more representative of the patterns of variation in practices than the industry. Therefore they looked at differences in the context in which different types of projects are found.

Using data of a worldwide survey with 2,339 project management practitioners, Besner and Hobbs (2012) chose to compare the four biggest subgroups: 'business and financial services', 'engineering and construction', 'IT and telecommunication', and 'computer software development'. They describe that business and financial service projects often take place in smaller organizations. Engineering and construction projects would have very specific contexts, were likely to be larger in comparison, better defined, and executed for external customers. Moreover, Besner and Hobbs (2012) found that compared to the other project types, engineering and construction projects are carried out by smaller and more project-oriented organizations, with many different disciplines involved. On the other hand, IT and telecom projects take place in larger organizations. Software development is often engaged in smaller projects, and requires a smaller number of disciplines.

Next to the PM approach and project type, one needs to know what the users want and need to make a sound decision regarding a PM system. This basically is a question of *user-centered design*.

1.3 User Centered Design

A system should be consistent with the knowledge and experiences of the user, making it is easer to learn and use (Roske-Hofstrand & Paap, 1986). Bringing the user into the development process is called User Centered Design (UCD). The ISO 13407 describes that the idea of UCD is that one is engaged in an iterative development cycle, where the users' needs are taken into account during developing, as well as the needs of the owner and the developer (Earthy, Jones, & Bevan, 2012). The UCD processes deal with the total system, using a multi-disciplinary activity, and focusing on making the system usable (ISO, 1999). A somewhat similar holistic design approach is the User Experience Design (UXD) by Garrett (2006), shown in figure 1. However, where the UCD focuses on making the system usable, the UXD also incorporates a satisfying user experience.

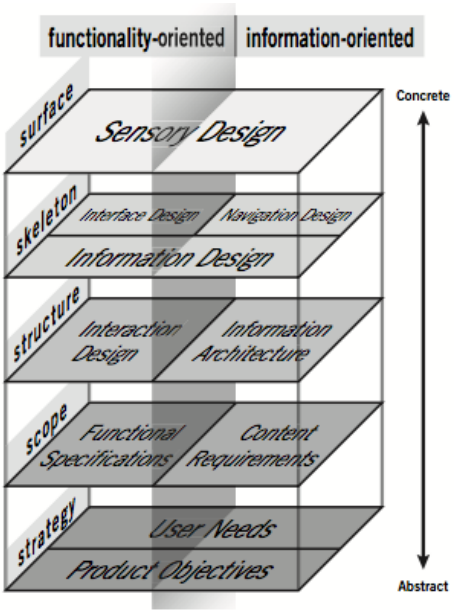


Figure 1. The Elements of User Experience Model (Garrett, 2006).

As with the UCD, UXD begins with a thorough understanding of the user needs and requirements. Through research and analysis, the interaction with the product is explored, accompanied by characteristics of the experience users desire. The next plane, 'scope', looks at the entire set of features the product will include, considering both functional and

informational aspects. With functional specifications, the set of operations the product will enable the user to perform are meant. The informational aspects, or content requirements, describe the information the product needs to communicate to the user.

Third, the structure is uncovered by looking at the information architecture. When developing systems, this is important because a system should be consistent with the knowledge and experiences of the user, making the system easier to learn and use (Roske-Hofstrand & Paap, 1986). But it also influences the emotional impact, making the product feel familiar and comfortable (Garrett, 2006). The last two steps entail more interaction and interface design related aspects, like arrangement and visual choices.

1.4 Requirement Engineering

For choosing a PM system, the users' wishes have to be clear. In this, requirement engineering is an important activity, for other phases in the development process depend on it. It can be defined as the process of seeking, uncovering, acquiring, and elaborating requirements (Zowghi & Coulin, 2005).

1.4.1 Eliciting Requirements

There are different elicitation techniques that could be used to gather requirements (Maguire & Bevan, 2002; Spath et al., 2012). Shams-Ul-Arif, Khan, and Gahyyur (2009) for example explain eighteen different tools for the elicitation part, giving advantages and disadvantages for each one. Also Zowghi and Coulin (2005) give a summary of twenty techniques with comparisons. The more applicable options for the current study, e.g. due to the number of participants or time constraints, are shown in table 2. The table summarizes methods to gather information, identify user needs once data has been collected, potential techniques that help with envisioning and evaluating the data, and lastly a technique that supports requirement specification. For each method or technique the benefits and drawbacks are written down.

Regardless of the research method, Mason (2010) explains that the sample size should be large enough to ensure that most or all important information is uncovered. This is according to the concept of saturation, describing the point on which new data does not shed any further light on the subject being researched. Another way to improve the saturation, one can use multiple iterations or methods to make sure as much information as possible is uncovered.

Table 2.

Methods that were reviewed and deemed applicable for the current study, accompanied by the benefits and drawbacks for each method.

Method	Benefits	Drawbacks
Information Gathering		
Task analysis	Provides a detailed understanding from which requirements can be discovered (Maguire & Bevan, 2002; Spath et al., 2012)	Focuses on very specific aspects in human work.
Semi structured interviews	Rich and detailed data, providing a more holistic view (Shams-Ul-Arif et al., 2009; Tong, Sainsbury, & Craig, 2007)	Takes a lot of effort and can be time consuming to collect data (Shams-Ul-Arif et al., 2009).
Contextual interviews	Detailed information and high external validity (Spath et al., 2012)	Whether all tasks are acquired is dependent on the domain and the professional, for the development process embodies a time span where some tasks might not occur in a certain time period. Therefore prone to produce incomplete information.
Identifying User Needs		
Focus groups	Discussion and exchange can enhance understanding, it can make that the acquired knowledge is greater than the sum of individual's domain knowledge, and it can increase the quality of the acquired knowledge (Darwish, Mohamed, & Abdelghany, 2016; Liou, 1992).	In contradiction with other research (Nijstad, Stroebe, & Lodewijkx, 2006). Forsyth (2009) speaks of 'process losses', where Straus, Parker, Bruce, and Dembosky (2009) explain the phenomenon called 'production blocking', saying that listening to other members and waiting for one's turn to speak can block the production of new ideas.
Persona Technique	Routinely used technique in human-computer interaction discipline (Cooper, Reinmann, & Cronin, 2007). Help in keeping the development process user-centered (Van Velsen, Wentzel, & Van Gemert-Pijnen, 2013).	May raise expectations too much and could over simplify the population (Maguire & Bevan, 2002).
Envisioning & Evaluating		
Card Sorting	Effective means for getting the optimal organizations of information as seen from the users' perspective (Wood & Wood, 2008).	Deep knowledge about the domain is required (Shams-Ul-Arif et al., 2009), indicating that this could be used after other elicitation techniques.
Prototyping	Provides detailed information and is especially useful when developing new systems and graphical user interfaces (Shams-Ul-Arif et al., 2009).	Users may become attached to the prototype and resistant to alternative systems or solutions (Zowghi & Coulin, 2005).
Requirement Specification		
MoSCoW	Robust against changes in cutoff scores (Beltman, Vosslamber, Molderink, & Noordzij, 2016).	Could lack clarity in distinguishing the priority of requirements (Krishnan, 2015).

1.4.2 Documentation

After the elicitation face, the output is analyzed and transformed into requirements, which are in turn documented. Requirements can express different areas, e.g., requirements representing user needs, or requirements concerning the design. Maguire and Bevan (2002) for example make a distinction between user requirements, usability requirements, and organizational requirements, while Hansen, Berente, and Lyytinen (2009) talk about design requirements. Van Velsen, Wentzel, and Van Gemert-Pijnen (2013) use functional and modality requirements, service requirements, organizational requirements, content requirements, usability and user experience requirements. Looking at the UXD of Garrett (2006), the following distinction can be made in requirement types: functional, content, structure, and more design related requirements. The distinction made can depend on the subject under investigation during the research study.

Requirements can be documented by writing down the requirement name, an identifier, and a description. In addition, Van Velsen et al. (2013) documented the requirement type, value, attribute, fit criteria, priority, and possible conflicts, where Ambler (2004) states it is also optional to include an example for each requirement, or the source to verify, related requirements, or revision history. One can also use scenarios and use cases that provide detailed and realistic examples to aid the understanding of requirements (Maguire & Bevan, 2002; Spath et al., 2012).

1.4.3 Verifying and Validating Requirements

A distinction can be made between verification and validation, where verification is about proving the requirements have been satisfied (Maalem & Zarour, 2016). Validation concerns the evaluation at the end of the development cycle (Boehm, 1984). Criteria for the verification and validating of requirements are about checking the consistency, completeness, feasibility, and testability of the elicited requirements (Boehm, 1984; Lee, In, & Kazman, 2014). Verifying outcomes with users offers a high degree of certainty of credibility (Rosenthal, 2016; Seale, 1999), which is in accordance with Austin and Sutton (2014), who state that checking the data enhances face validity and reliability. According to Shams-Ul-Arif et al. (2009), observation techniques are mostly used to verify and validate requirements, like for example prototyping (Boehm, 1984; Hansen et al., 2009; Spath et al., 2012).

1.5 Company Background

Elitac is a company engaged in product development, making vibration electronics in various wearable textiles. In order to realize the development of these products, they work with a multidisciplinary team of ten employees, including a project manager, scientists, software and electronics developers, designers, and professionals engaged in commercial aspects. This makes that knowledge is distributed among different persons. The product development process is therefore characterized by a demand for intensive exchange of knowledge between the professionals working in the different domains. This entails for example information about the progress, or knowledge for the engineering process, like design specifications, user and system requirements, etc. Their development entails hardware development like printed circuit board design (PCB) and textile wearable's, concurrent with software programming, using the agile-scrum methodology. Compared to the results from the study of Besner and Hobbs (2012), Elitac could be placed in the 'engineering and construction' project combined with 'software development'. Though, software development may be different from what participants reported in Besner and Hobbs' study, for software engineers engaged in hardware development spend a lot of time interacting with the hardware, e.g. loading and running the software, configuring hardware, debugging, etc. (Singer, Lethbridge, Vinson, & Anquetil, 1997).

In most cases, the projects Elitac works on embody a larger period of time (months to years), starting with an idea and investigation concerning the possibilities. When a more concord plan is developed, many iteration sessions for both software and hardware follow. Currently, professionals at Elitac mainly rely on face-to-face communication and documentation in Microsoft OneNote, making that information is scattered among different places. In order to manage the project process more effectively, Elitac would like to use a PM system in which all the stakeholders and development team members can check the progress, manage the development process, and find the information they need. Currently five of the nine professionals use the system 'Jira'. However, they looked into alternative PM platforms and think they want to start working with the system 'Taiga'; a platform for agile developers, designers, and project managers ("Taiga.io", 2016). However, no requirement analysis was done among the professionals working at Elitac upon choosing this system. So the question remains if this system fits the requirements of the professionals, stating what they would like and need.

1.6 Goal of this study

The main goal of this study is to evaluate the PM platforms Jira and Taiga, by deriving requirements concerning the design of such systems, and review if these are represented in the systems. The question guiding this goal is formulated as follows: "*How should a Project Management System be designed for Elitac to achieve an efficient workflow*?"

In keeping with the UXD, the first phase seeks to expand the understanding of the user needs. This is done by uncovering what goals the users have for interacting with the system, and what characteristics the users desire concerning the experience, leading to the question:

1. What are the goals and objectives users have for using a PM system?

The second step focuses on the functional and informational aspects of the system. Translating this into questions resulted into the following sub questions:

2. What functions (set of operations to enable the user to perform) do users want to see in a PM system?
3. What information do the users want to be available in the system?

To uncover the information architecture, a more structure related question concerns:

4. What mental model does the average user have of the content in the PM system?

And lastly, looking at the PM platforms 'Jira' and 'Taiga', the question remains:

5. How do the user requirements relate to the currently used system 'Jira', and the alternative chosen system 'Taiga'?

1.7 Structure

The next section outlines the methodology used, starting with an overview of the research design and explanation. Next, this section gives detailed information about the methods used to gather the information, and the chosen methods for the analysis. The section 'results', offers the outcomes generated in the elicitation phases. And lastly, the conclusions are presented, followed by a discussion.

2. Method

2.1 Research Design

For the current research, a qualitative approach was conducted, for qualitative research contributes to new knowledge and can give new perspectives (Tong et al., 2007). The research design for this study is shown in figure 2.

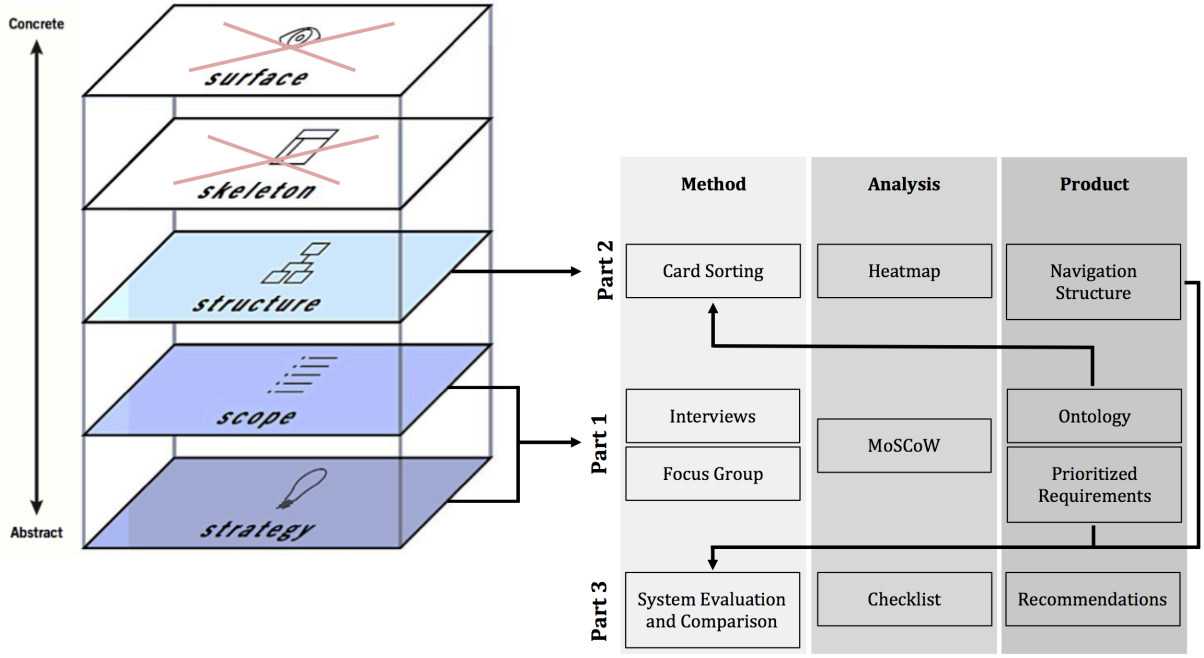


Figure 2. Tailor made model for current study, showing the three parts of this study, what methods are used to elicit information for the first three planes of the UXD model, and the chosen methods for the analysis, and the resulting products.

Part one seeks to answer the first three sub questions: 'what are the goals and objectives users have for using a PM system?', 'what functions (set of operations to enable the user to perform) do users want to see in a PM system?', and 'what information do the users want to be available in the system?'. Information is elicited by conducting semi-structured interviews. Even though all nine professionals working at the company Elitac were partaking in this study, it could be that important information was missed due to the small number of professionals representing the different disciplines. Therefore, the outcomes of the interviews were verified using a focus group. As mentioned before, this offers a high degree of certainty of credibility, enhancing face validity and reliability (Austin & Sutton, 2014; Rosenthal,

2016; Seale, 1999). Findings were transcribed, coded, translated to requirements, and prioritized using the MoSCoW method. The resulting prioritized requirement list and the ontology were used in other stages of the study.

Card sorting was a means in part two of the study. Looking at the information architecture, this technique was to answer the last sub question 'what mental model does the average user have of the content in the PM system?'. Using a heatmap to analyze the results, this gave rise to mental models.

The third part was to answer the fifth sub question: 'how do the user requirements relate to the currently used system 'Jira', and the alternative chosen system 'Taiga'?'. The products from the prior parts (prioritized requirements and mental model) provided for evaluating and making a comparison of how the outcomes related with the two PM systems.

2.2 Participants

All nine employees working at Elitac took part in this study, of whom seven males and two women, their age ranging from 21 to 38 years of age (M = 29.33; SD = 4.80). As mentioned before, the employees at Elitac work with a multidisciplinary team including a project manager, scientists, software and electronics developers, designers, and professionals engaged in commercial aspects. Prior to the study, participants received an informed consent, which had to be signed in order to participate. This informed consent can be found in appendix A. The methodology was approved by the BMS ethics committee of University Twente.

2.3 Part 1 Interviews and Focus Group

The first part entailed interviews and after analysis, a focus group to verify the results with all participants. The methodology is explained in more detail in the following part.

2.3.1 Interviews. Nine semi-structured interviews were conducted (face-to-face) with the participants separately.

Materials. In preparation, an interview protocol was composed, providing some demographic questions, and pre-made probe questions relating to the first three sub questions of the study. Themes that the protocol was divided in followed these questions, resulting in the following themes: 'goals and objectives', 'functional requirements', and 'content information'. The first version of the protocol was tested during a pilot to get feedback on the

timing of the session, activities, wording of questions, and to make the protocol and procedure better. The participant for the pilot did not work at Elitac. Afterwards alterations were made, resulting in a second version protocol used for the main study. However, after each interview also alterations were made so new information could be discussed in following interviews, improving the completeness. The last version of the protocol is included in appendix B. The interviews were voice-recorded, using a laptop, and a phone as backup.

Procedure. The researcher approached all participants individually at the office during working hours. After giving a short introduction of what the research was about and giving an indication of the duration or the interview (60 minutes), the researcher asked if the professional wanted to participate. If said yes, an appointment was scheduled for the first interview. The interviews were held in a conference room provided by Elitac. Each interview started with a little introduction to the research topic and goal. The participant received the informed consent on which the research topic and goal again were explained, as were the duration of the interview, and that participating was voluntary and that if the participant reconsidered, he or she was free to withdraw at any minute. Only when signed, the interview could continue, and the recordings could start.

Following the themes of the interview protocol, the interview started with a few demographic questions like age, function at Elitac, and years working at Elitac, the three themes, and an ending of the interview. The first theme was about the goals and objectives of participants for using a PM system, and thus sought for answers concerning the first sub question. Examples of questions that were asked are ‘how is a project currently managed?’ and ‘what do you think could a PM system provide for you?’. The second theme consisted of questions regarding functional requirements, asking what would make the system more or less useful. Next, the third theme entailed questions about content information. Giving rise to questions like ‘what information do you need from others’, ‘what information do others need from you’, etc. As stated earlier, the interviews were semi-structured, meaning that during the interviews there was a defined line of objects and questions to discuss, but also the opportunity to go further into an answer given by the participant. Lastly, the participant was asked if there was something he/she wanted to mention regarding the subjects discussed or something he/she thought hadn’t come up during the interview. The participant was thanked for partaking in the interview, and the recordings were stopped and saved properly. The duration was approximately 60 minutes per interview.

Data analysis. The audio files of the conducted interviews were transcribed to written text for further analysis. This was done using the program F5 to play back the audio files in a lower speed. To aid the translation process from interview to requirements, quotes that captured something important in relation to the overall goal(s) were identified and coded, converting raw data from the interviews into usable data by identifying themes, concepts, or ideas and their connections (Austin & Sutton, 2014). This was done with an inductive approach, for codes emerged from the data. Assisting this process, the data managing program ATLAS.ti was used. Another researcher also coded two randomly chosen interviews in order to test the coding protocol and if needed revise it to ensure the coding scheme would be applied consistently. To later translate the raw data into requirements, part of the steps explained by Van Velsen et al. (2013) were used. As important quotes in the interviews were coded and grouped in the taxonomy, these represented attributes on which requirements could be formulated. To help analyze wishes regarding functionality of the system and look for relations etc., the program MindMaple Lite was used and made the results visual.

2.3.2 Focus Group. To check for redundancy and completeness regarding user needs, a focus group was carried out with all participants together.

Materials. As for the interviews, a protocol was made, providing some pre-made probe questions. The five themes discussed were based on the major functionality that became clear after analyzing the results of the interviews and creating the mindmap. The major functionality-themes in which the user needs could be divided were formulated as follows: ‘backlog’, ‘task level’, ‘sprint level’, ‘long term planning’, and ‘others. The protocol is included in appendix C. To further guide this focus group, a PowerPoint presentation was created. For this presentation, each PowerPoint slide showed one branch of the mindmap at a time, after which the whole mindmap was presented. Each branch represented a major functionality-theme. A TV screen was used to present the PowerPoint on, and a laptop and phone were available to record the focus group for later analysis.

Procedure. The focus group was conducted in a conference room at Elitac. A short introduction was given, stating the goal of the focus group and duration of approximately 60 minutes. The recordings started, after which the researcher shortly explained the five themes that would be discussed during the session. Each slide of the presentation showed one branch of the mindmap at a time and thus a major functionality-theme. The researcher summarized the slide and asked for each item on the mindmap some questions, i.e. ‘what do you think

about item number X?', 'do you miss functionality that should have been at display here?'. This was done for all the branches of the mindmap. The researcher guided the discussions that arose, asking the opinion of multiple professionals. When consensus arrived about the items that should be available for the major functionality, i.e. regarding the backlog, the researcher summarized what was discussed, asked if this was correct, and if so moved on to the next major functionality branch. After all themes were discussed, the whole mindmap was presented. Lastly, the researcher asked the participant if there was something they would like to add or discuss regarding the functionality. The recording was stopped and saved, and participants were thanked for participating.

Data analysis. The focus group was transcribed using the program F5. As for the analysis of the interviews, the focus group was coded with an inductive approach using ATLAS.ti. This was done because the focus group was meant to verify the results, but also uncover new and thus missed data. The same steps explained by Van Velsen et al. (2013) were used, coding the important quotes and clustering them in a taxonomy. For the new codes, new requirements were formulated and added to the already existing list. Also, the mindmap was altered, resulting in a second version, which is included in appendix D.

2.3.3 User requirements. Using the taxonomy and mind map, requirements were formulated. This was done following the guideline from Cooper et al. (2007), stating that a requirement consists of an action, object, and context, e.g. 'see (action) an overview of the deadlines (object) for each project (context)'. An independent analyst checked the taxonomy, mind map, and the formulated requirements, after which disagreements and suggestions were discussed. In the end, this made that on a small scale the taxonomy was altered incorporating the feedback from the other analyst. For each requirement the following information was written down: the requirement identifier (No.), and name. In order for the system evaluation and comparison, acceptance criteria were formulated for each requirement.

2.3.4 MoSCoW method. The user requirements were prioritized using the MoSCoW method, resulting in a list in which the requirements that presented a higher value to the user were placed closer to the top. This was done with all professionals together. Professionals were already acquainted with this method. They have cards, providing every participant with a set of cards representing a 'must', 'should', 'could', and 'won't' written on it. An example of the cards is attached in appendix E. The requirements were presented one at a time, in a

PowerPoint. For each presented requirement, all participants had to place one card on the table (placed upside down; the must/should/ could/won't facing the table). When all participants had a card on the table, the cards were turned and discussed until there was consensus about the priority of that requirement. When consensus arrived, the researcher wrote down the result, and the process started anew with another requirement. The results were later on written down in the fourth column in the requirement tables.

2.3.5 Ontology. The labels concerning content information were summarized in an ontology, providing a list of vocabulary that could be used throughout the system, accompanied with an explanation for each label. They can serve as a basis for technology's data structure, useful because consensus about this vocabulary can avoid miscommunication, misunderstanding, and inconsistencies (TNO & TUD, 2012). Therefore, the ontology was communicated with all professionals. This provided for checking if the information labels were correct, if there was content missing, and gave rise to universal label names that were understood by all participants. Checking the ontology was initially done face to face. However, after three participants it became clear that this was a very time consuming process. Therefor, the other six professionals received the list via mail and were asked to respond within a week. Based on the feedback, alterations were made.

2.4 Part 2 Card Sorting

Part two of the study sought to uncover the mental models participants had for the information in the PMS. This was done using the card sorting technique. For each participant separately, his or her mental model was explored. Using the labels summarized in the ontology, the card sorting task was done with an open, hierarchical sort.

Materials. Paper cards were made, on which content information was visible. The ontology constructed earlier was placed next to the participant on the table, presenting all the labels with a description for each item.

Microsoft Excel was used for making the matrixes and conducting the average of all data. Python was later on used to create the heatmap and cluster the items.

Procedure. The card sorting task took place in a room provided by Elitac, with all nine participants separately. Before starting the test, the participant was given instructions, what it entailed.

All cards were shuffled and the stack of cards was given to the participant. Next, the participant was asked to sort the items, with the possibility to sort them in different levels. This was done because the labels provided all content for different project phases, where some information relates more to each other. For in the PM system, it is interesting to see in what proximity certain information could be placed. For each pile, the participant was asked to name it. If they couldn't think of an applicable name, the pile remained unnamed.

Data analysis. After all the card sorting tasks were conducted with the professionals, the results were analyzed by making a similarity matrix. First, as done by Schmettow and Sommer (2016), a similarity measure for hierarchical card sorts was used; the *Jaccard coefficient*.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

For two items (A and B) the Jaccard coefficient is constructed by counting the number of groups both items are members of, which is divided by the number of groups at least one item is a member of. When this was done for all nine card sorts, the average for each item in the matrix was calculated. This was done in Microsoft Excel.

Next, the resulting average similarity matrix was loaded into Python. A similarity matrix consists of number ranging from 0 to 1, 1 meaning the items are very similar or equal, where 0 indicates the items never occurred in the same set, and thus no similarity. To visualize the average mental model, a heatmap was created. This is a graphical representation where the warmer or darker colors represent a stronger similarity. It was chosen to create a heatmap, because compared to dendograms these convey more information about the similarity. To cluster the items, it was chosen to compute the correlation distance between them in Python. The code used can be found in appendix F.

$$1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{||(u - \bar{u})||_2 ||(v - \bar{v})||_2}$$

2.5 Part 3 System Evaluation and Comparison

The third part of the study entailed an evaluation of both the PM systems Jira and Taiga.

Materials. As mentioned earlier, acceptance criteria were created for all requirements, serving as a checklist. These were formulated by stating each steps the system had to fulfill in order to mark a requirement as fulfilled. These were listed in the column in between the

requirement name and assigned priority. The checklist provided a fifth column that was added to state if the requirement was met (‘OK’) or failed (‘-‘).

Both programs were needed to test them. To check Jira without disturbing the workflow at Elitac, a free 7-days account was created. For Taiga there already was an account made which could be used.

Procedure. First the PM system Jira was tested. Each task related to a requirement was tested starting with the first requirement down. For each requirement that the researcher tried to perform in the system, the acceptance criteria had to be met in order to set the requirement to ‘OK’ or failed. This was the same for the PM system Taiga. Whenever the researcher wanted to double check or because it wasn’t sure if the requirement exactly did what it should do, the Jira/Taiga website was consulted.

Also for the content information the system was checked, by looking into the possibility to add multiple fields for information, if these could be giving different headings and if a description could be added to these fields. Also the order in which the fields could be added is checked. If the information could be arranged in the clusters and order the average mental model presented, this requirement was met.

Data analysis. When both systems were checked, the overall percentages of accepted requirements was calculated for each main functionality. This was also done for the ‘must haves’, ‘should haves’, and ‘could haves’ separately for each main functionality.

3. Results

In this study, the main question was formulated as follows: *"How should a Project Management System be designed for Elitac to achieve an efficient workflow?"* To answer this question, the research was divided into three parts. The first part entailed nine interviews and a focus group. The goal was to get more insight into the goal and objectives of the professionals for using a PM system. Also, the researcher asked about the users' wishes regarding the system. Based on this information, requirements were formulated and prioritized using the MoSCoW method. Lastly, the interviews provided for eliciting content information that would be used throughout the PM system. This was summarized in an ontology and later on used during the card sorting test. The second part sought to uncover mental models by conducting nine card sorting tests. The content labels summarized in the ontology were sorted, providing a navigation structure. This knowledge can be used by Elitac to customize the chosen PM system. And lastly, the third part of this study was to evaluate both PM systems Jira and Taiga, and comparing the results between the two.

3.1 Part 1 Interviews and Focus Group

Outcomes of the interviews and focus group are described in this section. Using the three sub-questions presented in the introduction, this section concerns the goals and objectives of professionals for using a PM system, their wishes regarding functionality, and what content information has to be available throughout the system.

3.1.1 Goals and objectives

During the interviews and focus group, the main goal and objective for using the PM system was expressed to be that it provides insight regarding the short term and long term planning, to document the progress, and because it could provide more structure to the process. The professionals currently working with a PM system Jira said they could see what tasks are scheduled for the upcoming two weeks. But not all professionals are working with this system. And even though the tasks for the near future are clear for most professionals, the overall picture is lost to them. In all nine interviews professionals referred to the current lack of insight into this process. "Ik vind het fijn voor mezelf om het grotere plaatje te hebben van wat we zouden willen bereiken binnen een project. Dat is nog niet altijd helemaal duidelijk" (r3), explains one of the professionals.

The executing professionals explained this means there is no overview of what tasks will come next, where for others this refers to the difficulty of managing the process. In general, one person manages and oversees the projects, keeping the overview of the progress, tasks for the (near) future, deadlines, and more. This is experienced as pleasant by most professionals, for one can work on a few tasks, and when finished ask what can be done next. However, the lack of knowledge of what tasks the future will bring has some consequences; "van de ene kant is mijn workload nu heel goed gemanaged door haar, maar van de andere kant kan ik niet altijd goed bedenken wat er handig zou zijn om nu te doen" (r4). Professionals cannot plan ahead, and cannot prepare or think about an upcoming task. Moreover, one professional mentions the risk of losing the overall goal of what they are doing. "Als je daar zelf ook geen overzicht over hebt dan weet je ook niet zo goed waar je het voor doet" (r6).

The lack of insight in the process is not only towards tasks that have to be executed, for seeing what other professionals are doing and working on is also preferred. Professionals want to know what other colleagues are working on, making professionals feel more involved. In addition, professionals knows with whom and when to communicate about tasks.

Relating this insight to the long term planning, several professionals talked about an overview of the overall project, upcoming deadlines, and the availability of professionals during the projects. Also, wanting to see in what phase one is working, i.e. the exploring phase, developing phase, evaluation, etc.

Next to insight and managing the development process, the goal of using a PM system is to document the process. In addition, almost all professionals thought and hoped the system could provide more structure to projects.

3.1.2 Functionality

Translating the information into requirements resulted in 62 requirements, which were prioritized using the MoSCoW technique. Of these requirements, 18 were deemed 'must haves', 31 'should haves', and the remaining 13 were labeled 'could haves'. None of the requirements was deemed a 'won't have'. Table 3 shows a summary of the most important requirements that came out of this study, the 'must haves'. The complete list with 62 requirements can be found in the appendix.

During the interviews and focus group, it became apparent that the requirements

Table 3.

The most important requirements ('must haves') in this study for each category.

	No.	Description	Acceptance Criteria
User story / Task level	1	Add a user story/task	• The title of the story/task can be inserted • A minimum of one text field is available to add information regarding the story/task
	5	Inset an indication of time to solve/execute the user story/task	• Professionals can insert an indication of time for each story/task • The time is described in points/minutes/hours
	7	Add dependencies to user stories/tasks	• A user stories or task that is dependent of another story/task can be linked
	8	See dependent user stories/tasks within the story/task	• For each story/task, dependent stories and/or tasks are visible for all professionals
	12	See which professional works on the user story/task	• Stories and tasks can be assigned to one or more professionals
Subtask level	17	See what professional works on the subtask	• Subtasks can be assigned to one or more professionals
Backlog	23	See an indication of time to solve/execute a story or task in the backlog	• For each story/task in the backlog, the assigned time is shown
	24	See a hierarchy of importance of user stories/tasks in the backlog	• Professionals can prioritize user stories/tasks in the backlog • The most important stories/tasks are shown at the top of the backlog
Sprint level	31	Add a workflow status to a story/task	• Not more than one status can be chosen for a story/task/subtask
	33	See the workflow for user stories/tasks within a sprint overview	• All professionals can see the assigned statuses of user stories/ tasks/subtasks within the sprint overview
	36	See what professionals are working on the subtasks within a sprint overview	• Professionals that are assigned to a subtasks are visible in the sprint overview
	37	See the progress of the sprint	• Professionals are able to see the progress of the overall sprint-status
Long term planning	41	See the progress in the long term planning (developments)	• The active sprint is displayed in the long term planning • Completed sprints are displayed in the planning • Professionals can see the sprints related to the near future
	42	Distinguish different projects in the long term planning	• Different projects can be displayed • The starting date is shown for each project • The ending date is shown for each project
	44	See dependencies in the long term planning	• Professionals can see what sprints are related to not yet finished sprints
	45	Add an important date to a project into the long term planning	• Professionals are able to add an important date like a deadline to a project
Others	55	Work with multiple professionals in the system at the same time	• All professionals can work in the system at the same time • All changes made in the system are carried through so the system is up to date
	57	Track changes made in the system	• Recently made changes are traceable • All professionals can see into recently made changes

mentioned by the participants could be divided into different groups. For example, a distinction could be made between requirements concerning a user story/tasks and requirements regarding the long term planning. Consequently, the requirements were divided into six topics; ‘user story and task level’, ‘subtask level’, ‘backlog’, ‘sprint level’, ‘long term planning’, and lastly the topic called ‘others’ referring to more general information that could not be placed into the former five topics.

User story and task level. On user story and task level, certain functionality is required, like adding additional information, giving indication of time, and seeing dependencies. All professionals mention that while adding a user story or task to the PM system, it has to be possible to give a title to the story/task. Starting a project, the professionals formulate user stories and tasks that have to be executed to realize a finished product. Additional information should be able to be inserted to give some context to the item: “het zou wel mooi zijn als er in ieder geval een basis idee neer zou kunnen zetten” (r2). Two management related professionals expressed that adding information should be done by following mandatory steps, providing more structure. However, another professional mentioned this could be too demanding, i.e. forcing one to add multiple fields of information. In the end, most of the professionals thought this to be a ‘should have’.

Another feature six of the professionals mentioned they would like to see is giving the item an indication of time for solving or executing it. This provides insight in how long or how difficult a task is expected to be. Also seeing what professional is working on the item, and what dependencies there exist between tasks are must have-features. “Ik denk dat het aller belangrijkste is om de afhankelijkheden tussen en binnen personen goed inzichtelijk te hebben” (r5).

Two requirements expressed to be less important are asking another person via the system to review a task, and adding feedback by the reviewer. Still, four of the professionals considered these must have-requirements.

Subtask level. For the subtask level, the functionality is similar to the features described above concerning user story and task level. However, functionality used on user storytask level is deemed more important, where most requirements on subtask level are said to be ‘should haves’. Only one requirement regarding seeing what professional is working on a subtask presented to be a must-have requirement for the PM system.

Backlog. Adding user stories and tasks to a PM system, a list emerges, embodying tasks that will be worked on in the (near) future. This list is the backlog. In agile PM, items

are prioritized, stating what items are most important at a certain time. Making this priority visible in the backlog makes it easier to state what will be done in the coming sprint: “dat je gelijk weet; oh die dingen moeten heel snel af zijn. En misschien ook wel een soort hiërarchie, dus wat bovenaan staat sneller af moet zijn” (r1). Also the time estimation is of importance here, for the upcoming sprint should contain not more items or a combined difficulty level than the professionals expect they can execute in the sprint.

One professional mentioned a ticket system could be useful, for newly added items are checked before being added to the backlog. Even though this requirement was mentioned by only one professional, and acquired only one vote to be a must have, the professional mentioning this requirement was the only one representing the discipline.

> “Dus nu is het zo dat … maakt 80 taken aan waarvan ik er misschien 30 helemaal niet zo belangrijk vind. Dus die schuif ik dan naar onder, maar eigenlijk wil ik die niet in de backlog want daar gaan we nooit aan toekomen want er komen altijd belangrijkere dingen tussendoor” (r5).

As a side note, the professional mentioned that declined items in this ticket system should not be deleted, but stored in the system with a reasoning for why it is declined for the moment.

Other ‘should have’ requirements relate to organizing the backlog by clustering the items, i.e. on subject or project. Initially, two professionals said it could be a nice feature to see the backlog visually instead of a list. In addition, seeing the dependencies between the items in the backlog could make it easier to plan a sprint. The team now prevents that dependent items are planned together in the same sprint, because this can result in a lot of delay.

Sprint level. Every two weeks a planning is made of what is to be done for the two-weekly period; called a ‘sprint’. Being able to see the overall progress of the sprint as a more detailed sprint overview was initially mentioned by six professionals, and later on deemed a must have for seven of the professionals.

> “Het scherm waar ik het meest naar kijk in Jira is het scherm dat de informatie verzameld wie met welke taak bezig is, welke taken er af zijn, welke taken er nog gedaan moeten worden op korte termijn. Dus binnen de sprint” (r9).

As the professional explains in the quote, the sprint overview is an important screen. In this overview, two of the professionals mentioned they wanted to see workflow stages. These give information about the status of each separate task. Examples of workflow stages are ‘to do’, ‘in progress’, ‘to review’, ‘blocked’, ‘done’. Being a ‘must have’ for the one, and a ‘should have’ for the other, this accounts for user stories and tasks as well as the subtasks. Another must have-requirement is to see what professionals are engaged in what tasks. Two professionals initially mentioned this during the interviews.

For the ‘should haves’, the visibility of the indicated time for each task is said to be useful. Also, the last feature professionals pointed out, was to see the consequences when the planning changes. So when an additional task is added to a sprint, what does this mean for the feasibility of the sprint, and what does this do to the next sprint.

Long term planning. Next to the short term planning also a long term planning separate for the different projects would be useful according to all nine professionals. Except for one, all professionals mentioned during the interviews they want to see the duration of a project, and important dates and deadlines accompanied by information about this date. In addition, six of them said to want to see the progress and dependencies within the long term planning.

> “Als je alle afhankelijke taken achter elkaar zet, dan kan je een pad tekenen, het afhankelijke pad…”, explained one professional, “…dat wil je eigenlijk inzichtelijk maken, wat dat is dat pad. Welke taken mogen absoluut niet gaan uitlopen want dan gaan we het einde van het project niet op tijd halen” (r3).

Two requirements regarding the long term planning that were deemed less important (‘could haves’) were about displaying what professionals is working on a project, and absences, i.e. vacation. The option to see this long term planning visually is mush preferred.

Others. Two requirements referring to more general functionality were thought to be ‘must haves’: the possibility for multiple professionals to work at the same time in the system, and being able to track all changes made in the system. “Het is heel belangrijk dat iedereen gewoon gelijker tijd aan die website kan rommelen” (r9).

More ‘should have’ related features are a dashboard when starting the system. Preferably, this dashboard can be customized for each professional separately. And one professional said being up to date on changes that are made while working in the system

could be useful. The professionals at Elitac active in selling the products pointed out that a progress of system versions should be available. They want to see what system version is worked on at the moment and when the deadline for this version is so they know when they can promote and sell it.

Less important functionality was in regards to other programs that are used next to the PM system. There is 'OneNote', 'SharePoint', 'Git', and 'WeWorked'. For all these programs almost all professionals communicated the same question, namely; could it be possible to integrate this in the PM system? "Zo'n systeem voor alle documenten zou ook mooi zijn, dus niet alleen voor code maar voor alle documenten" (r2). Still, this requirement was considered a 'could have' during prioritizing. Lastly, a phone application of the system was mentioned to be useful. However, only two of the professionals have the application for the current PM system, of which only one uses it frequently.

For the eighteen must have-requirements a discovery graph was created, stating on the x- axis how often a requirement was discovered over all the interviews and focus group, and the y-axis showing how frequent that was the case in the current study. For example, there were five requirements that were discovered two times over all the interviews and or focus group. The graph is shown in figure 3.

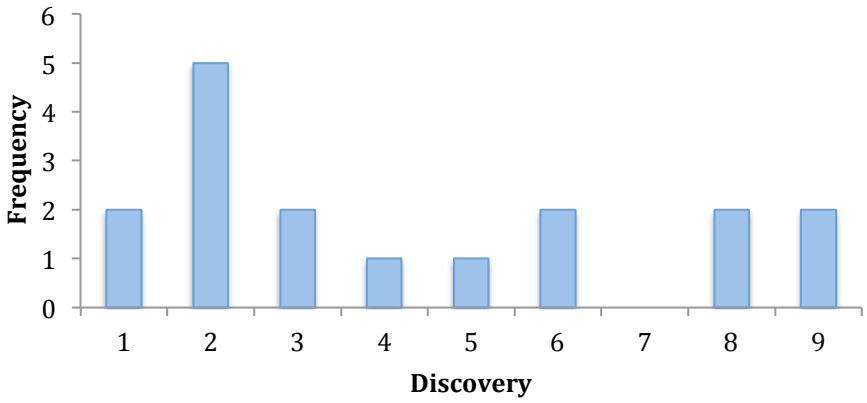


Figure 3. Graph presenting how frequent a requirement was discovered over all the interviews and focus group.

During the interviews, two of the eighteen must have-requirements were voiced by all of the professionals. In contrast two of the requirements were only mentioned by a single professional, i.e. number 55, "work with multiple professionals in the system at the same

time'. This was later on said to be an absolute must have, however, maybe so obvious to most that it was not thought of by more professionals.

During the interviews, it became salient that certain professionals mentioned the same requirements, seeming to divide the professionals in two groups. For example, requirements #31, #32, #33, and #34 were often named together by participants r2, r3, r4, r6 and r9. Other professionals (r5, r7, r8) mentioned requirements like #23, #48, #51 This difference could indicate there are different groups (roles). We'll get back to this in the discussion.

3.1.3 System content

When working with a PM system, one can add information to the system by adding text in fields or uploading documents. The professionals were asked what content they would want to use in the PM system. The results are described in this section, and later on the content was translated into 30 labels. These are summarized and alphabetically ordered in an ontology, accompanied by a description for each item separately (included in appendix G).

All professionals said that an added user story or task would be incomplete with only a title available. Adding some context, for example the goal and/or acceptance criteria could help in making a user story or task better understandable. Moreover, because the time between adding a user story or task and picking it up to execute varies a quit deal. As a professionals describes, some of the user stories/tasks without context will not be executed because no one remembers what the story/task was about: "dan vraag ik nog eens een keer van; joh waar gaat het eigenlijk over? En een deel weet al niemand meer waar het over gaat" (r9). Information is needed about why the item is added. So the reason/goal of the items is preferred. "Een background moet ik wel terug kunnen vinden of in ieder geval een link ernaartoe. Wat er precies moet gebeuren, waarom" (r3).

Additional information expressed by four professionals during the interviews was information concerning the client and target group, personas and/or user journeys. Two other professionals talked about some indication of what makes that a story or task is completed, like acceptance criteria, to be available at a story or task.

For some user stories and tasks research has to be conducted by the professionals, meaning they have to look into literature or test at what temperature the hardware is washable. Currently, the findings are documented in Microsoft OneNote, but all professionals would like to some of the findings in the PM system at the user story itself. Although the amount of detail directly available at the story or task is questionable and differs between

professionals, all want to know in short what methodology was used and what the concluding results were. “Ik wil helemaal niet de hele achtergrond weten, ik wil weten wat daar uit komt” (r8). Two professionals suggest that when one wants to look into the research in more detail, i.e. the whole methodology, he or she can go to the document in OneNote.

Next to this content, the development professionals talked about needing information in the PM system about the software, hardware, casing, and textile to be available in the PM system. For each of the disciplines, the constraints are required to be added to a user story or task. “Ik moet hun randvoorwaarden weten en zij moeten mijn randvoorwaarden weten, zodat we weten wat wel kan en wat niet kan” (r4). Measurements/size of the hardware is documented here also. For the software development, two professionals mentioned they wanted to know the Git flow ID accompanying the story/task. Like said earlier, Git is a version control system that helps a software team manage changes to source code over time. Because different versions and branches are created, the ID is useful to know in which file one worked. Other content for in the PM system is related to log data, bug reports, and looking from the graphical interface designer point of view, there are logos, icons, and images needed for making concepts.

It can be concluded that professionals need certain information to help in comprehending and making executing a user story or task more effective and efficient. Like said before, the time between adding a story or a tasks and executing it can vary. In the current situation, incomplete information sometimes results in a lack of understanding what was meant, and deleting the item altogether.

3.2 Part 2 Card Sorting

The content information elicited during the interviews and focus group gave rise to 30 content labels. These were summarized in the ontology and used during nine card sorting sessions. Uncovering the average mental model of the professionals provided for making a navigation structure. This could be used for structuring the content in the chosen PM system. The average mental model of the system content is presented in this section.

Figure 4 shows the resulting mental model similarity matrix as a heatmap. As explained in the method section, the items in the similarity matrix were clustered computing

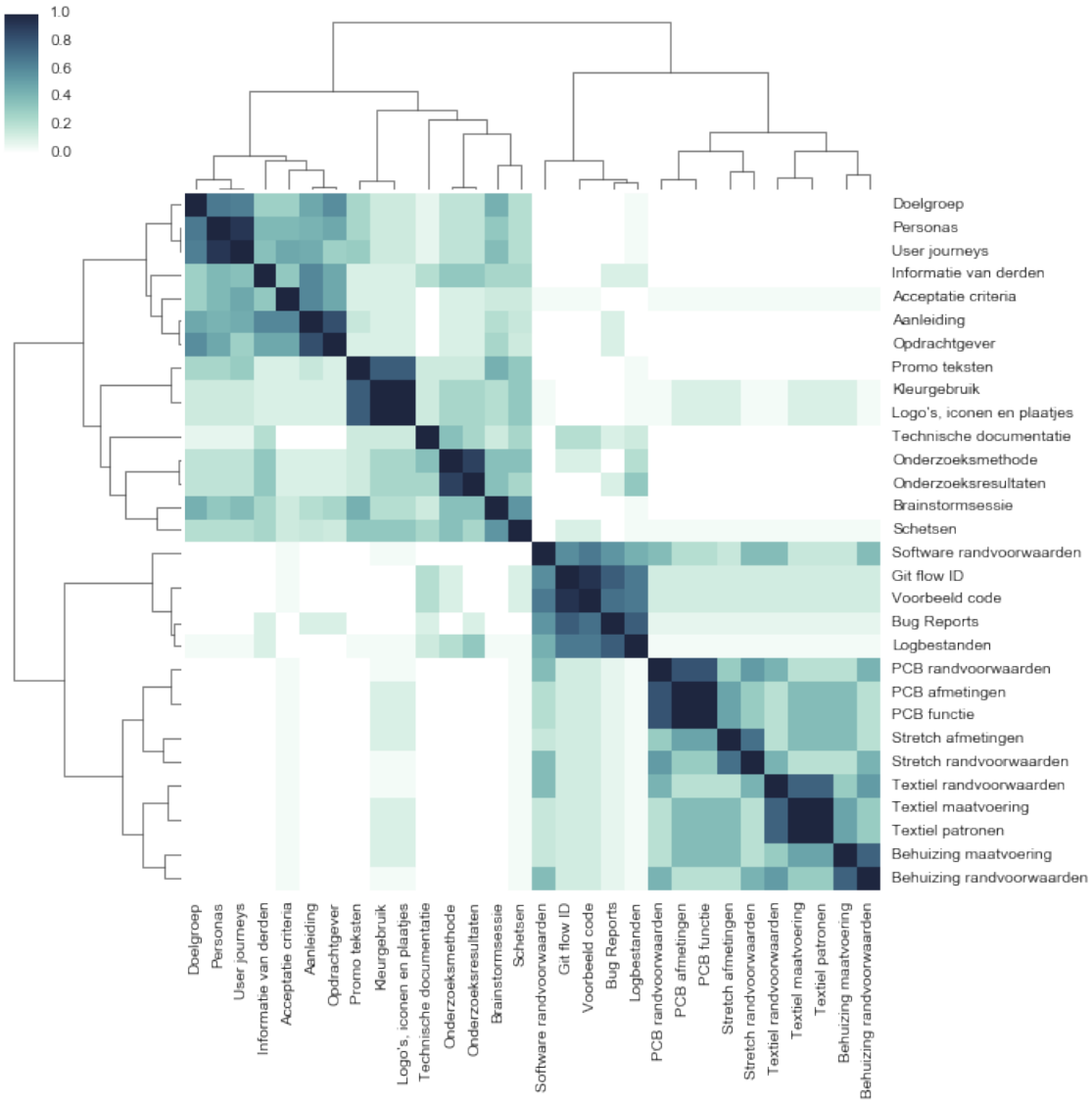


Figure 4. Mental model heatmap of the 30 content labels that will be used throughout the PM system.

the correlation distance between the items. Several groups become visible in the diagonal of the model. Two halves are visible, dividing the content into items concerning the information that is needed before developing the product, and more product developing related items. For the top half, the item ‘promo teksten’ was commonly grouped together with items such as ‘kleurgebruik’ and ‘logo’s iconen en plaatjes’. As were the items ‘doelgroep’, ‘personas’ and ‘user journeys’. According to the mental model, ‘doelgroep’, ‘personas’ and ‘user journeys’ should be put into a group with ‘informatie van derden’, ‘acceptatie criteria’, ‘aanleiding’ and ‘opdrachtgever’. In the lower half of the model for example, one can see five groups regarding ‘behuizing’, ‘textiel’, ‘stretch’, ‘PCB’, and ‘software. However, the prior four could be placed in a higher-level-group stating ‘hardware’ related items.

Other darker regions that are not on the diagonal indicate that an item is associated with a second group. For example ‘software randvoorwaarden’, which is grouped with other

software related subjects like ‘GIT flow ID’, but also with ‘PCB randvoorwaarden’ and ‘textiel randvoorwaarden’. As for the interviews, this difference in grouping during the card sorting, seemed to divide the participants into two groups, in particular in the lower half of the model. Participants r1, r2, r3, r4, r6, and r9 all grouped the cards together addressing the same discipline; ‘textiel randvoorwaarden’, ‘textiel maatvoering’, and ‘textiel patronen’. The same for all the ‘PCB’-labels, ‘behuizing’-labels, and software-related labels. However, participants r5, r7, and r8 grouped the ‘randvoorwaarden’ together, separate from the other content; ‘software randvoorwaarden’, ‘PCB randvoorwaarden’, ‘textiel randvoorwaarden’, and ‘behuizing randvoorwaarden’. This difference will be addressed in the discussion.

Roughly, the content can be clustered in three levels, resulting in a navigation structure presented in table 4. While performing the card sorting, professionals could group

Table 4. *Navigation structure based on the heatmap*

1st level	**2nd level**	**3rd level**
Achtergrond informatie / project basis	Aanleiding / doel	Doelgroep Personas User journeys
		Informatie van derden Acceptatie criteria Aanleiding Opdrachtgever
	Documentatie / onderzoek	Promo teksten Kleurgebruik Logo’s, iconen en plaatjes
		Technische documentatie Onderzoeksmethode Onderzoeksresultaten Brainstormsessie Schetsen
Eisen / implementatie / ontwerp / uitwerking binnen discipline	Software	Software randvoorwaarden Git flow ID Voorbeeld code Bug reports Logbestanden
	Vorm factor / hardware	PCB randvoorwaarden PCB afmetingen PCB functie Stretch afmetingen Stretch randvoorwaarden
		Textiel randvoorwaarden Textiel maatvoering Textiel patronen Behuizing maatvoering Behuizing randvoorwaarden

the cards hierarchically. This resulted in clustered groups, which they were asked to give a name to. Labels mentioned by the professionals to describe the different clusters ranged from very specific, detailed labels to labels for higher-level, clustered groups. For the above half in the heatmap, items were clustered using label names like ‘aanleiding’, ‘doel’, ‘achtergrond informatie’, ‘documentatie’, ‘onderzoek’, and ‘project basis’. For the lower half of the model, professionals gave more higher-level related clusters names like i.e. ‘eisen’, ‘implementatie’, ‘ontwerp’ and ‘uitwerking binnen discipline’. More detailed labels level entailed ‘software’, and ‘vorm factor’ or ‘hardware’. This information gave rise to the labels for the three levels in the navigation structure. Based on the average mental model, table 4 above proposes a navigation structure that could be used in a PM system.

3.3 Part 3 System Evaluation and Comparison

The 62 requirements together with the formulated acceptance criteria made a checklist that was used to evaluate both PM systems Jira and Taiga. Both checklists can be found in appendix H and I. This section presents the results of the evaluation.

Table 5 shows the difference in the number of requirements that could meet the acceptance criteria for both PM systems Jira and Taiga. Overall, Jira fulfills 87,10% of all the 62 professionals’ wishes regarding a PM system, where Taiga could fulfill 56,45%.

Table 5.

The numbers and percentages of accepted requirements for both PM systems Jira and Taiga.

Requirement category	Accepted requirements for Jira in numbers and percentages (54/62; *M* = 87,10%)		Accepted requirements for Taiga in numbers and percentages (35/62; *M* = 56,45%)	
User story/task level	12/13	92,31%	9/13	69,23%
Subtask level	8/9	88,89%	5/9	55,56%
Backlog	7/8	87,50%	6/8	75%
Sprint level	10/10	100%	8/10	80%
Long term planning	8/12	66,67%	0/12	0%
Others	9/10	90%	7/10	70%

Jira scored higher on all categories. For both Jira and Taiga, the highest percentages of requirements that were met by the PM system was at sprint level. The lowest scores were with 66,67% on the long term planning in Jira, and 0% in Taiga.

Figure 5 provides more depth regarding the numbers and percentage of accepted requirements, looking for the two PM systems at the ‘must/should/could have’ requirements.

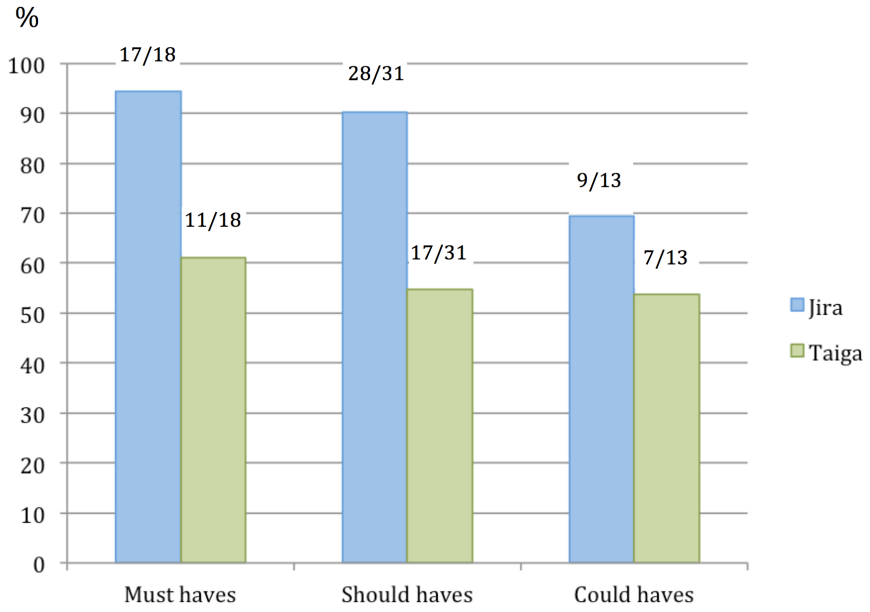


Figure 5. Graph illustrating the numbers and percentages of accepted requirements divided for 'must', 'should', and 'could haves' for both PM systems Jira and Taiga.

Looking at how many of the more important requirements could be fulfilled by the system, Jira could execute 17 out of the 18 'must haves', where Taiga could fulfill 11 of them. As for the must have-requirements, also more of the 'should' and 'could haves' could be met by the PM system Jira compared to Taiga.

Regarding the content, the system was checked by looking into the possibility to add multiple fields of information, giving different headings, providing a description to these fields, and changing the order of these fields. For both systems, multiple fields could be added, providing the ability to divide the information over several fields. Moreover, in both systems it was able to add different properties to these fields, like 'text' or 'url'. So far, no restrictions were encountered regarding the content that should be possible to add to the system while testing both PM systems.

4. Discussion

This research was conducted to evaluate the PM systems Jira and Taiga for the company Elitac, by deriving the users' wishes regarding a PM system. This gave rise to the research question: *"How should a Project Management System be designed for Elitac to achieve an efficient workflow?"* During the first part of the study, conducting nine interviews and a focus group, 62 user needs were found. The requirements were prioritized using the MoSCoW technique. Card sorting tests were conducted to uncover what navigation structure Elitac would want to use throughout the PM system. Both PM systems Jira and Taiga were reflected on using all requirements to evaluate the systems, taking into account that the must have-requirements were more important than the 'should' and 'could haves'. The results pointed out that the PM system Jira could execute more user requirements in comparison to Taiga, indicating that Jira would be a better fit for Elitac. The results are reflected below. After, possible recommendations are presented concerning the PM systems. And lastly, limitations are discussed, and recommendations for future research are presented.

4.1 Reflection results

In concordance with the definition given by Castillo (2016a), Collyer et al. (2010), and Hoda and Murugesan (2016), Elitac works with the most popular agile method, the scrum method. They start with a high-level framework plan, while details become more explicit during the development process. As professionals during the current study expressed, the team wants to add new user stories/tasks to the PM systems' backlog and prioritize these. Based on that information, a sprint is created, freezing the design features for the coming two weeks. Therefor Elitac needs a PM system in which the agile approach, moreover the scrum method, is supported. Both of the PM systems could support this, however, some differences in functionality remained.

Except for one, Jira could meet all 'must-have' requirements, where Taiga met eleven of the eighteen requirements. Looking in more detail to the users' wishes regarding a PM system, Jira and Taiga differ in how many requirements they could support. Requirements that could not be met in Jira could also not be executed by Taiga. In the opposite direction however, there were requirements Taiga could not support, while Jira could. For example, functionality regarding dependencies was frequently expressed by the professionals, which could unfortunately not be met by Taiga. Not only was this expressed as a requirement on

user story and task level, on subtask level and in the backlog, but in all those cases, this wish was claimed to be a must or should have. Looking at the Kano model, which captures the relationship between product performance and customer satisfaction, the must-be attribute similar to the must-have requirement has to be fulfilled (Matzler & Hinterhuber, 1998). According to this model, if the must-be attributes are not fulfilled, the customer will be extremely dissatisfied. So this is a mayor drawback for Taiga. And as a consequence, the wish to see what happens to related tasks on sprint level when the planning changes can also not be met. This could be a big issue for Elitac, because they would like to start using a PM system with different disciplines, and this could result in a lot of dependencies between tasks. Not only are hardware and software dependent of each other, but there are also dependencies between the different hardware components. As said before, this could not be supported by Taiga, but the developers of Taiga claim this is an issue they are going to tackle in the future (GitHub, 2017). If this changes in the (near) future, and thus meet these requirements, this could make Taiga a more attractive option than it is now when looking at the amount and importance of requirements the system could fulfill.

For the long term planning, both Jira and Taiga could support fewer wishes than in the other categories, where Taiga couldn't meet any of the requirements. The developers of Taiga see a calendar function as a low priority (Alonso, 2015), whereas this is wished by the professionals working at Elitac. It is obvious that for managing a project, this is an important feature to plan ahead. The long term planning could be separated from the PM system. But having this integrated in the PM system would make it a lot more efficient for the managing professionals, for all the information concerning the sprint-planning, due date, etc. are already in the system. If the long term planning has to be managed in a separate system, the professionals have to manually add these dates to the planning-system, and oversee planning related changes in the PM system and update these manually in the planning program. The contradicting results in the current study to what Taiga has to offer, could be due to the fact that there are developers stating on the internet they consider the long term planning not necessarily to be in agile methodology (Taiga.io, 2016b). This is in concordance with Berteig (2013), who says that scrum assumes the team is committed to delivering value, without being concerned with time boxes. However, being a company that has to deliver products to customers and are tied to release dates, this is not possible. In addition, Elitac has multiple disciplines that are going to use the system, and for the sales related professionals, the long term planning would be more important than the detailed information the sprint level

provides. Moreover, also professionals engaged in developing mentioned they would like to have an overview of the long term planning, which would contradict that this is a feature only the management and sales related professionals would want this feature. So the fact that Taiga does not support this, while Jira does, is a big disadvantage.

As described, the results in this study showed that Jira could support more requirements concerning the PM system for Elitac than Taiga could. However, what has to be taken into account is that five of the professionals are already working with Jira and thus used to this system. For Elitac to choose this system, an issue could be that this experience can blind users, preventing them from recognizing possibilities for new functions (Carroll & Rosson, 1987). However, if Elitac would go for another PM system, this could result in loss aversion. This explains that the perceived loss is greater then the gain associated with receiving it (Tversky & Kahneman, 1991). And as Novemsky and Kahneman (2005) mention, a choice is often made looking at the changes relative to a reference point. So if the other PM system misses functionality that Jira provides, this could displease the professionals acquainted with Jira. Still, this would be the case for the five professionals already using a PM system, where the other four professionals have never worked with such system.

Next to the discovered needs and wishes regarding the PM system, there is a requirement that was not taken into account while comparing the systems; the pricing of a PM system. Where Taiga is for free, Jira brings costs to the table. Depending on how important this requirement is this could be a game-changer for Elitac. Especially because even though Jira fulfills more of the users' wishes, there are other ways to incorporate more user requirements. For example, the long term planning could be covered using a separate program to manage this planning. However, as said before, the downside is that these two programs will not be integrated, thus making changes in the PM system that affects the planning requires alterations in both programs. So it could be that choosing for Taiga makes it harder to manage projects because certain requirements could not be met, and it could be more time consuming for having to do double work. Thus, a balance will have to be struck between the ease of use that having one integrated program brings, against a cheaper solution that would require an additional program.

During the interviews and the card sorting tests, some group differences became salient. This could be due to the diversity in disciplines among the professionals working at Elitac. Looking at this a bit closer, it seemed that the differences in the results could be explained by dividing the professionals into two groups, whereas one represents the

developers and the other more management and sales related professionals. While evaluating both PM systems, this difference was not taken into account. This could result in a list with wishes that entails functionality often not used in a PM system. Still, Jira could support a lot of requirements, maybe because it has a lot of plugins available (meaning extra features that can be installed to the basic program, such as a calendar). Making that it is possible to integrate functionality into the systems is an advantage, for it could be that there are also plugins for features the professionals want to have in the future, or are currently not aware of they want or need. Because the needs and wishes of users are inclined to change over time, it would be a big advantage to have plugins available that can adept to these new needs. These features are according to the Kano model the most important in influencing how satisfied the user will be (Matzler & Hinterhuber, 1998).

4.2 Implications for practice

After reflecting on the results for the PM systems Jira and Taiga, three possible recommendations regarding the choice of PM system are explained in this section.

Staying with Jira

One of the recommendations could be to stay with the currently used PM system Jira. As said before, the results showed that of the 62 requirements, 54 could be executed in Jira. In addition, except for one, Jira could meet all of the most important requirements. The only one that could not be met is for the professionals to see dependencies within the long term planning. This makes information in the long term planning less complete. However, one can see which user stories/tasks are blocked in the backlog. And when for example a blocked requirement is dragged into the new two-week planning, the system will show that it is blocked. Nonetheless, the lack of this feature makes it harder to keep an overview of the long term planning.

Something else to take into account for Elitac if they chose to stay with Jira, is that still four of the professionals have to learn how to use the PM system. And because the system has a big variety in features and settings, and is less intuitive, this can overwhelm professionals that have never worked with a PM system before.

An advantage that Jira has to offer, is the availability of plugins that could support possible future wishes. This makes that whenever a new need arises, there is a chance a plugin can be installed to fulfill this need.

Going with Taiga

Another choice could be to switch to the PM system Taiga. This system was created to provide a simple and intuitive PM system, which would be easy and fast to learn. It acquires a lot of features the professionals would like to see, but there are some important features that could not be supported by this system. One of them is concerning adding and being able to see dependencies between user stories/tasks, and between subtasks. The absence of this feature makes it impossible to see what stories/tasks and subtasks need extra or immediate attention. For example, if the hardware components are not yet ordered, other tasks that address assembling the hardware are blocked and cannot be executed. So the dependencies help in visualizing what possible difficulties one might encounter, making this a big disadvantage in managing the process.

Another important feature that is lacking in Taiga is within the sprint-overview, for Taiga cannot preview the workflow for the user stories/tasks. Whenever a story contains subtasks, the progress of these subtasks are shown in the sprint overview. However, when there are no subtasks, one has to click on the user story before one can see the progress. This does not result in major consequences, but makes it nonetheless inefficient.

Furthermore, it is not possible to manage the long term planning in Taiga. To compensate, an additional program could be used to manage the long term planning, but this makes the whole process more difficult for the systems do not synchronize. So the additional program has to be edited manually if a change occurs in the PM system. This is not only really inefficient, but can also result in failures if a change in the PM system is overlooked or if the change was incorrectly made in the additional system. On the plus side, Taiga can be installed for free.

Altogether, choosing for Taiga would mean that all nine professionals have to invest time in learning a new system, the professionals have to learn how to work with an additional program to manage the long term planning, and the managing professionals have to oversee changes made in the PM system, and manually make changes to this additional program for the long term planning. In the end, even though the system is for free, it would seem that this system makes PM more time invasive.

Look for another PM system

The third possibility could be to look for another PM system using the knowledge acquired in this study. There are a lot of PM systems available for agile development, i.e. Trello, Microsoft Project. Nevertheless, the first step is to look into these other options and testing if there is a system that incorporates more requirements then Jira does. But this was already done by one of the professionals, and to do it more thoroughly would take time. And again, as for Taiga, there is the fact that all nine professionals have to learn this new system. However, if there is a system that better suits the needs and wishes of Elitac's professionals, it could be worth the investment of searching and testing PM systems, and learning how to use this system.

4.3 Strengths and limitations

The goal of this study was to derive the users' wishes regarding a PM system. A strength of this study was that multiple methods were used that made verifying the results possible. This should offer a higher degree of certainty of credibility, and enhance face validity and reliability (Austin & Sutton, 2014; Rosenthal, 2016; Seale, 1999).

Nevertheless, the completeness of the requirements is uncertain and thus a possible limitation. As Mason (2010) explains, a large enough sample size should helps to ensure most to all important information is uncovered. Elitac employs nine professionals. Even though this regards the whole population, the sample size is rather small. In addition, the professionals working at Elitac have different disciplines, representing different viewpoints. This resulted in a broad scope regarding user requirements. And for a discipline that is represented by just one professional, that one person has to express all user requirements in order to draw a complete picture. A consequence and thus issue could be that the user needs are not complete. Wishes regarding functionality could be missed or forgotten to mention by the professionals. According to the concept of saturation, eliciting information must go on until no new data is presented on the subject being researched (Mason, 2010). And as the last interview in the current study discussed new information, leading to new requirements, the level of saturation was not yet reached. This same issue about completeness could also be regarding the system content. Before the card sorting tests, the ontology was checked with three professionals face to face. To six of the nine professionals the list was send via mail, asking for feedback within a week's time. Unfortunately, not all participants took the time to look at it, resulting in feedback from only five of the nine participants in total. This could

lead to the possibility that content is missing or not given the correct (universal) name for all participants, where after this could have affected the results during the card sorting tests. Taking the possibility that the requirement or content information could not be complete, a problem could be that the evaluation of the systems Jira and Taiga is based on incomplete requirements. Maybe, due to the absence of uncovered, yet important requirements, less difference between the PM systems was visible, or another conclusion was drawn altogether.

Another possible limitation is that five of the professionals are already working with Jira and thus acquainted with the system. This could present an issue, for professionals could have described what they already knew and used, which could have biased the expressed needs for a new system. They could have been primed towards the system they knew, and in such a way resulting in the assimilation bias. This is in concordance with Carroll and Rosson (1987) who state that people interpret new situations based on what they already know. For example, when one is used to typing documents with a typewriter, they will seek functions in Microsoft Word based on what they know about working with the typewriter. This would reflect more on what currently is available for the user, instead of presenting what the user really wants.

4.4 Reflection research design

The model of User Experience Design (UXD) by Garrett (2006) was used as the foundation for the research design in this study. I find this model explains the steps one has to take in order to design a product really well. In this study however, only the first three planes of the UXD were followed, starting with identifying the user needs and product objectives. The second plane focused more on the functional and content requirements, where the goal of the third plane was to get insight in the information architecture. This third step was more design related, while the overall goal in this study was to derive requirements for a PM system and compare the two systems Jira and Taiga to see which of the two fit Elitac better. The requirements found provided for a checklist that made it clear what functionality a system has to incorporate. Furthermore, a clear navigation structure was researched, which is useful for Elitac to structure the content throughout the system. However, besides checking whether support was available to incorporate this structure, the actual result of the research into this structure was not used. In the end, this work was not strictly necessary to reach the same conclusions in this study. Though it could be used by Elitac to structure the content in the PM system.

Reflecting on the methods, overall I would say the ones used during the study were very effective. The interviews gave rise to a lot of information and I am really satisfied with the amount of knowledge it elicited. However, as mentioned before, because five of the professionals are already acquainted with Jira it could be that the results are biased towards Jira. So in hindsight, I could have framed some questions differently. Instead of asking about what they would like to see in a PM system, I could have focused more on the tasks that accompany product development, and the problems they encounter in their work.

The focus group made it possible to verify the results and gave rise to two requirements that were initially not mentioned during the interviews. In my opinion, this really complemented the interviews that were held separately. Also it provided more understanding of the requirements since professionals were asked to explain why certain requirements would be useful. The only thing I encountered during the focus group was that the professionals more experienced with using a PM system seemed to provide the most input, in addition to the more management related professionals. Based on this knowledge, the thing I would change if I would do this again, is conducting the focus group with smaller groups, maybe grouping them by the differences found during the interviews; the management related professionals and product development related professionals separate from each other.

Prior to the card sorting tests, all the content that has to be available in the PM system mentioned during the interviews was summarized in a list accompanied by a description for each label. The ontology was send to all participants for feedback. This provided for checking if the information labels were correct, if there was content missing, and gave rise to universal label names that were understood by all participants. Checking the ontology was initially done face to face. However, after three participants it became clear that this was a very time consuming process. Therefor, the other six professionals received the list via mail and were asked to respond within a week. Unfortunately, this wasn’t done by all participants, resulting in feedback from five of the nine participants in total. This could add to the possibility that content is missing or not given the correct (universal) name for all participants. Alternatively, I could have discussed the ontology in a short session with all professionals together. This would have been less time consuming compared to checking with each professional separately, but you would still get input from all participants. The card sorting itself is an approach that resulted in clear output. I would certainly recommend this approach to others who want to uncover a navigation structure.

In the end, the resulted products for Elitac are a list stating all the requirements, an ontology regarding the content in the PM system, a navigation structure, and recommendations for which system to choose. Some of the results still have to be presented to Elitac. However, a few reactions are already expressed. For example, after showing the requirement list to one of the professionals, a few weeks later she commented that while looking at Taiga she had a better view of what the system could provide and what was missing. Moreover, the requirement-list makes for a clear overview of what Elitac is looking for, and what is of more importance among these needs. The results that came after testing and comparing the two systems were discussed with the three related professionals. Even thought they initially thought Taiga could still be a suitable option, the results made it clear there were very important features Taiga could not meet. Ultimately, they have chosen to stay with Jira and customize the system so the needs and wishes are accounted for in the PM system.

4.5 Optional recommendations

In this section some optional recommendations to optimize working with the PM system are presented. The first item is that it could be of use to look at on how the system should be customized, such as visual aspects. For instance what colors should be used or how many and what names should be given to workstages. The importance of customizing the workstages is for example that the system becomes more consistent with the knowledge professionals have for the development process, making it is easer to use (Roske-Hofstrand & Paap, 1986). And looking at what colors to use and choosing the ones that are deemed most wished by the users could for example help in making that working with the system becomes more intuitive (Kennedy, 2017). This could be done by conducting usability tests: formulating tasks that the professionals have to perform, and look at difficulties or inconsistencies they encounter.

Another recommendation is to look how and where accompanying information should be documented. During this study it became clear that professionals encounter difficulties finding the right information. In a lot of cases they didn't know where to look for the information because of the navigation structure. Also, within a document, the findings often seem to be documented without a specific template. So it is proposed to look into possible navigation structures for the accompanying programs Microsoft OneNote and SharePoint, and a template to structure the documented information. In both cases this could be done by first looking at what distinction there could be made in project and document content.

Afterwards, an open card sorting technique could be used to discover the navigation structure as explained by Wood and Wood (2008).

In the end the chosen methods with the UXD as a guideline, provided a scientific basis for this study to uncover the users' wishes regarding a PM system. Three recommendations are presented, concerning staying with Jira, switching to Taiga, or looking for another PM system altogether. However, the results found in the current study would indicate that the PM system Jira would make a better fit in regards to the users' wishes for Elitac.

5. References

Alonso, A. (2015). Taiga issue #2199 Calendar. Retrieved from https://tree.taiga.io/project/taiga/issue/2199

Ambler, S. W. (2004). *The Object Primer. Agile Model-Driven Development with UML 2.0* (3th ed.). New York: Cambridge University Press.

Austin, Z., & Sutton, J. (2014). Qualitative research: Getting started. *Canadian Journal of Hospital Pharmacy*, *67*(6), 436–440. http://doi.org/10.4212/cjhp.v67i6.1406

Beltman, S., Vosslamber, S. M., Molderink, A., & Noordzij, M. L. (2016). Toward the Design of an Energy Consumption Feedback System. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *24*(3), 9–16. http://doi.org/10.1177/1064804615611283

Berteig, M. (2013). The rules of scrum: I do not track my hours or my "actual" time on tasks. Retrieved from http://www.agileadvice.com/2013/07/29/referenceinformation/the-rules-of-scrum-i-do-not-track-my-hours-or-my-actual-time-on-tasks/

Besner, C., & Hobbs, B. (2012). An Empirical Identification of Project Management Toolsets and a Comparison Among Project Types. *Project Management Journal*, *43*(5), 24–46. http://doi.org/10.1002/pmj.21292

Bjarnason, E., Unterkalmsteiner, M., Borg, M., & Engström, E. (2016). Multi-case study of agile requirements engineering and the use of test cases as requirements. *Information and Software Technology*, *77*, 61–79. http://doi.org/10.1016/j.infsof.2016.03.008

Boehm, B. W. (1984). Verifying Validating Requirements Specifications. *IEEE Software*, *1*(1), 75–88.

Boehm, B. W. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64–69. http://doi.org/10.1109/2.976920

Carroll, J. M., & Rosson, M. B. (1987). Paradox of the active user. *In J.M. Carroll, Ed., Interfacing Thought. MIT Press*, 80–111.

Castillo, F. (2016a). Agile-Scrum Project Management. In *Managing Information Technology* (1st ed., pp. 199–210). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-38891-5_8

Castillo, F. (2016b). Managing Projects. In *Managing Information Technology* (1st ed., pp. 85–146). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-

38891-5_5

Collyer, S., Warren, C., Hemsley, B., & Stevens, C. (2010). Aim, fire, aim-Project planning styles in dynamic environments. *Project Management Journal*, *41*(4), 108–121. http://doi.org/10.1002/pmj.20199

Cooper, A., Reinmann, R., & Cronin, D. (2007). *About Face 3: The essentials of interaction design*. *Information Visualization*. Indianapolis, IN: Wiley Publishing, Inc. http://doi.org/10.1057/palgrave.ivs.9500066

Coram, M., & Bohner, S. (2005). The Impact of Agile Methods on Software Project Management. In *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)* (pp. 363–370). IEEE. http://doi.org/10.1109/ECBS.2005.68

Darwish, N. R., Mohamed, A. A., & Abdelghany, A. S. (2016). A Hybrid Machine Learning Model for Selecting Suitable Requirements Elicitation Techniques. *(IJCSIS) International Journal of Computer Science and Infroamtion Security*, *14*(6), 380–391.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. http://doi.org/10.1016/j.infsof.2008.01.006

Earthy, J., Jones, B. S., & Bevan, N. (2012). ISO Standards for User-Centered Design and the Specification of Usability. In *Usability in Government Systems* (pp. 267–283). Elsevier. http://doi.org/10.1016/B978-0-12-391063-9.00049-3

Forsyth, D. R. (2009). *Group Dynamics* (5th ed.). Wadsworth: Cengage Learning.

Garrett, J. J. (2006). Customer Loyalty and the Elements of User Experience. *Design Management Review*, *17*(1), 35–39. http://doi.org/10.1111/j.1948-7169.2006.tb00027.x

GitHub, I. (2017). Task/issue/user story dependencies #154. Retrieved from https://github.com/taigaio/taiga-back/issues/154

Hansen, S., Berente, N., & Lyytinen, K. (2009). Requirements in the 21st Century: Current Practice and Emerging Trends. In ... *Requirements Engineering: A Ten- ...* (Vol. 14, pp. 44–87). http://doi.org/10.1007/978-3-540-92966-6_3

Hoda, R., & Murugesan, L. K. (2016). Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, *117*, 245–257. http://doi.org/10.1016/j.jss.2016.02.049

ISO. (1999). *ISO 13407: Human Centred Design Process for Interactive Systems*. Retrieved from http://www.ash-consulting.com/ISO13407.pdf

Kahneman, D., Higgins, E., Amir, O., Ariely, D., Dhar, R., Frederick, S., … Werten-, K. (2005). The Boundaries of Loss Aversion. *Journal of Marketing Research*, *XLII*, 119–128. http://doi.org/10.1509/jmkr.42.2.119.62292

Kennedy, E. D. (2017). Color in UI design: A (practical) framework. Retrieved from https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e#.cg735upfp

Kerzner, H. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (10th ed.). Hoboken, New Jersey: John Wiley & Sons Inc.

Könnölä, K., Suomi, S., Mäkilä, T., Jokela, T., Rantala, V., & Lehtonen, T. (2016). Agile methods in embedded system development: Multiple-case study of three industrial cases. *Journal of Systems and Software*, *118*, 134–150. http://doi.org/10.1016/j.jss.2016.05.001

Krishnan, A. (2015). Techniques to Prioritize Requirements.

Lee, Y. K., In, H. P., & Kazman, R. (2014). Customer requirements validation method based on mental models. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, *1*, 199–206. http://doi.org/10.1109/APSEC.2014.39

Liou, Y. I. (1992). Collaborative Knowledge Acquisition. *Expert Systems with Applications*, *5*(1–2), 1–13.

Maalem, S., & Zarour, N. (2016). Challenge of validation in requirements engineering. *Journal of Innovation in Digital Ecosystems*, *3*(1), 15–21. http://doi.org/10.1016/j.jides.2016.05.001

Maguire, M., & Bevan, N. (2002). User requirements analysis: A review of supporting methods. In *Proceedings of IFIP 17th World Computer Congres* (pp. 133–148). http://doi.org/10.1007/978-0-387-35610-5_9

Mason, M. (2010). Sample Size and Saturation in PhD Studies Using Qualitative Interviews. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *11*(3). http://doi.org/ISSN 1438-5627

Matzler, K., & Hinterhuber, H. H. (1998). How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. *Technovation*, *18*(1), 25–38. http://doi.org/10.1016/S0166-4972(97)00072-2

Nijstad, B. A., Stroebe, W., & Lodewijkx, H. F. M. (2006). The illusion of group productivity: A reduction of failures explanation. *European Journal of Social*

Psychology, *36*(1), 31–48. http://doi.org/10.1002/ejsp.295

Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study. *Empirical Software Engineering*, *15*(6), 654–693. http://doi.org/10.1007/s10664-010-9136-6

Rosenthal, M. (2016). Qualitative research methods: Why, when, and how to conduct interviews and focus groups in pharmacy research. *Currents in Pharmacy Teaching and Learning*, *8*(4), 509–516. http://doi.org/10.1016/j.cptl.2016.03.021

Roske-Hofstrand, R. J., & Paap, K. R. (1986). Cognitive networks as a guide to menu organization: An application in the automated cockpit. *Ergonomics*, *29*(11), 1301–1311. http://doi.org/10.1080/00140138608967247

Schmettow, M., & Sommer, J. (2016). Linking card sorting to browsing performance – are congruent municipal websites more efficient to use? *Behaviour & Information Technology*, *35*(6), 452–470. http://doi.org/10.1080/0144929X.2016.1157207

Seale, C. (1999). Quality in qualitative research. *Qualitative Inquiry*, *5*(4), 465–478.

Serrador, P., & Pinto, J. K. (2015). Does Agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051. http://doi.org/10.1016/j.ijproman.2015.01.006

Shams-Ul-Arif, M. R., Khan, M. R. Q., & Gahyyur, S. A. K. (2009). Requirements engineering processes, tools/technologies, & methodologies. *International Journal of Reviews in Computing*, 41–56.

Singer, J., Lethbridge, T. C., Vinson, N., & Anquetil, N. (1997). An examination of software engineering work practices. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative Research, November 10-13* (pp. 174–188). Toronto, Ontario, Canada: IBM Press. http://doi.org/10.1145/782010.782031

Spath, D., Hermann, F., Peissner, M., & Sproll, S. (2012). User Requirements Collection and Analysis. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (4th ed., pp. 1311–1322). Hoboken, New Jersey: John Wiley & Sons Inc.

Špundak, M. (2014). Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? *Procedia - Social and Behavioral Sciences*, *119*, 939–948. http://doi.org/10.1016/j.sbspro.2014.03.105

Straus, S. G., Parker, A. M., & Bruce, J. B. (2011). The group matters: A review of processes and outcomes in intelligence analysis. *Group Dynamics: Theory, Research, and*

Practice, *15*(2), 128–146. http://doi.org/10.1037/a0022734

Taiga.io. (2016a). Retrieved November 2, 2016, from https://taiga.io

Taiga.io. (2016b). Why is there no time tracking? Retrieved from https://tree.taiga.io/support/frequently-asked-questions/why-is-there-no-time-tracking/

TNO, & TUD. Applying the Situated Cognitive Engineering Method - A Comprehensive Guide (2012). Retrieved from http://scetool.nl/sce_manual_v1.0.pdf

Tong, A., Sainsbury, P., & Craig, J. (2007). Consolidated criterio for reporting qualitative research (COREQ): a 32- item checklist for interviews and focus group. *International Journal of Qualitative in Health Care*, *19*(6), 349–357. http://doi.org/10.1093/intqhc/mzm042

Tversky, A., & Kahneman, D. (1991). Loss Aversion in Riskless Choice : A Reference-Dependent Model Author (s): Amos Tversky and Daniel Kahneman Published by : Oxford University Press. *The Quarterly Journal of Economics*, *106*(4), 1039–1061. Retrieved from http://www.jstor.org/stable/2937956?seq=1#page_scan_tab_contents%5Cnhttp://qje.oxfordjournals.org/content/106/4/1039.full.pdf+html

Van Velsen, L., Wentzel, J., & Van Gemert-Pijnen, J. E. W. C. (2013). Designing ehealth that matters via a multidisciplinary requirements development approach. *Journal of Medical Internet Research*, *15*(6). http://doi.org/10.2196/resprot.2547

Wood, J. R., & Wood, L. E. (2008). Card Sorting: Current practices and beyond. *Journal of Usability Studies*, *4*(1), 1–6. Retrieved from http://usabilityprofessionals.org/upa_publications/jus/2008november/JUS_Wood_Nov2008.pdf

Zandhuis, A., & Stellingwerf, R. (2012). *ISO21500: Guidance on project managemeng - A Pocket Guide*. Zaltbommel: Van Haren Publishing.

Zowghi, D., & Coulin, C. (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements* (pp. 19–46). Berlin/Heidelberg: Springer-Verlag. http://doi.org/10.1007/3-540-28244-0_2

Appendix A. Informed Consent

Je bent gevraagd om deel te nemen aan dit onderzoek. Voordat je besluit wel of niet deel te nemen wordt er informatie gegeven over het doel van het onderzoek en wat het zal inhouden. Neem de tijd om de volgende informatie door te nemen, en weet dat mocht je besluiten niet deel te willen nemen (ook tijdens het interview) je dit op elk moment kan aangeven.

Reden van het onderzoek: Bram heeft gekeken naar verschillende Project Management systemen die gebruikt kunnen worden tijdens een project. Het proces van het project kan worden gevolgd, en informatie die gedeeld moet worden kan bij het desbetreffende project worden opgeslagen. Nu wordt 'Jira' gebruikt. Het systeem dat er veelbelovend uitzag is 'Taiga'.

Het doel van dit onderzoek is om te kijken of dit systeem goed bij Elitac zou passen. Dit wordt gedaan door middel van deze interviews en een focus groep waarbij er gekeken wordt wat de wensen zijn vanuit de professionals werkend bij Elitac.

Het gesprek zal worden opgenomen om verder te kunnen analyseren. Het interview duurt ongeveer 60 minuten. Met de gegevens zal vertrouwelijk worden omgegaan.

Mocht je na het interview nog vragen hebben kan je mij altijd aanspreken op kantoor, of mailen naar renee@elitac.nl. Heel erg bedankt!

Datum

_____

Geboortedatum participant

Handtekening participant

_____ _____

Appendix B. Interview Protocol

Voorbereiding:

- Reserveren van vergaderruimte
- Print informed consent en een pen meenemen
- Batterijen van opneemmateriaal opladen (laptop en gsm)

Start interview:

- Geef informatie over het onderzoek:

Bedankt dat je tijd hebt om mee te doen aan het onderzoek. Ik zal beginnen met een korte uitleg met het doel van dit onderzoek. Binnenkort zullen jullie gaan werken met een Project Management systeem, waarin alle nodige informatie per project opgeslagen kan worden, en waar je het proces van een project kan volgen. Nu gebruiken jullie 'Jira'. Bram heeft gekeken naar systemen die gebruikt zouden kunnen worden, en uiteindelijk heeft hij voor 'Taiga' gekozen. Na dit interview zullen in een later stadium nog andere methoden bijdragen aan de resultaten van dit onderzoek. Uiteindelijk zal er gekeken worden of het gekozen systeem voldoet aan de wensen van iedereen werkzaam bij Elitac. Dit zal besproken worden met Carien en Bram. Met het interview is het doel om verwachtingen en doelen in kaart te brengen rondom een PM systeem, evenals wensen rondom de functionaliteit en de inhoud. Dit zal ongeveer een uur duren. Het is vrijwillig, dus mocht je je bedenken en besluiten niet mee te willen doen kan je dit altijd aangeven.

- Geef informed consent aan de participant en vraag of hij/zij dit zou willen doorlezen en wanneer overeengekomen om te tekenen

→ Wanneer ondertekend, start de opnamen en ga door met interview

Tijdens interview:

- Demografische vragen:
    - Wat is je leeftijd?
    - Wat is je functie binnen Elitac?
    - Hoeveel uur per week werk je bij Elitac?
- Thema 1: Goals and objectives → voor het beantwoorden van sub question 1 (What are the goals and objectives users have for using a PM system?)
    - Huidige situatie:

        - Kan je een beschrijving geven van je dagelijkse werkzaamheden?
        - Kan je een project noemen waar je aan gewerkt hebt, of momenteel aan werkt, en kan je wat informatie over dit project geven?
        - Kan het voorkomen dat je aan meerdere projecten werkt op een bepaald moment?
        - Hoe lang duurt een project gemiddeld?
        - Kan je een project in bepaalde fasen indelen?
        - Welke eisen/afhankelijkheden zijn er voor jou tijdens een project?
        - Hoe wordt een project momenteel gemanaged (proces, informatie)?
        - Wat zijn de gevolgen van de manier waarop het nu gemanaged wordt (positief en negatieve)?
    - PM systeem:
        - Wat is volgens jou het doel van het gebruiken van een PM systeem?
        - Wat denk je dat dit voor jou kan opbrengen?
        - Wat denk je dat dit voor het team kan opbrengen?
        - Wat denk je dat dit voor het gehele project kan opbrengen?
        - Doorvraag op bovenstaande vragen:
            - Kan je aangeven waarom?
- Thema 2: Functional requirements → voor het beantwoorden van sub question 2 (What functions (set of operations to enable the user to perform) do users want to see in a PM system?)
    - Welke functionaliteit zou je willen dat het systeem heeft?
        - Waar zou je het systeem voor willen gebruiken?
        - Hoe zou het systeem dat moeten doen?
        - Welke functionaliteit gebruik je nu graag?
        - Welke functionaliteit mis je aan het huidige systeem?
    - Wat maakt een systeem nuttig voor jou?
    - Wat maakt dat een systeem niet of minder nuttig zou zijn voor jou?
    - Doorvraag op bovenstaande vragen:
        - Kan je aangeven waarom?
- Thema 3: Content information → voor het beantwoorden van sub question 3 (What information do the users want to be available in the system?)
    - Welke informatie heb je van anderen nodig?

- Welke informatie hebben anderen van jou nodig?
- Doorvraag op bovenstaande vragen:
  - Wat voor een informatie moet er beschikbaar zijn?
  - Kan je je dagelijkse bezigheden beschrijven?
  - Wat voor een informatie zou je in het systeem willen opslaan?
  - Van wie ben je afhankelijk om je werkzaamheden uit te voeren?
  - Van wat ben je afhankelijk om je werkzaamheden uit te voeren (informatie van andere collega's, etc.)?

Einde van interview:

- Zou je een PM systeem willen gebruiken?
  - Waarom liever wel/niet?
- Laatste vraag: Is er iets wat je nog wil toevoegen of bespreken rondom dit onderwerp wat bijvoorbeeld nog niet besproken is?
- Bedank participant voor het meedoen
- Geef nogmaals aan dat de informatie besproken vertrouwelijk en anoniem is
- Stop opnamen en zorg dat deze goed zijn opgeslagen

Appendix C. Focus Group Protocol

Voorbereiding:

- Reserveren van vergaderruimte
- Batterijen van opneemmateriaal opladen (laptop en gsm)

Start interview:

- Geef informatie over het onderzoek:

Bedankt dat je tijd hebt om mee te doen aan het onderzoek. Ik zal beginnen met een korte uitleg met het doel van deze focus groep. Ik heb met iedereen van jullie een gesprek gehad waarin we bepaalde thema's hebben besproken. Deze gesprekken heb ik teruggeluisterd en geanalyseerd. De resultaten rondom de functionaliteit heb ik gesorteerd in vijf verschillende thema's, waarbij ik voor ieder thema de behoeftes heb genoteerd in een mindmap. Deze mindmap wil ik vandaag met jullie doornemen om te kijken of jullie je hierin vinden, of juist iets tegen komen wat waar jullie iets anders voor ogen zien of bijvoorbeeld nog functionaliteit missen. Ik heb hierbij een PowerPoint gemaakt. In totaal zal het ongeveer een uur duren. Het is vrijwillig, dus mocht iemand zich bedenken en besluiten niet mee te willen doen kan diegene dit altijd aangeven. Ook dit gesprek neem ik weer op om later terug te kunnen luisteren zodat ik niets over het hoofd zie.

- Start de opnamen en ga door met interview

Tijdens interview:

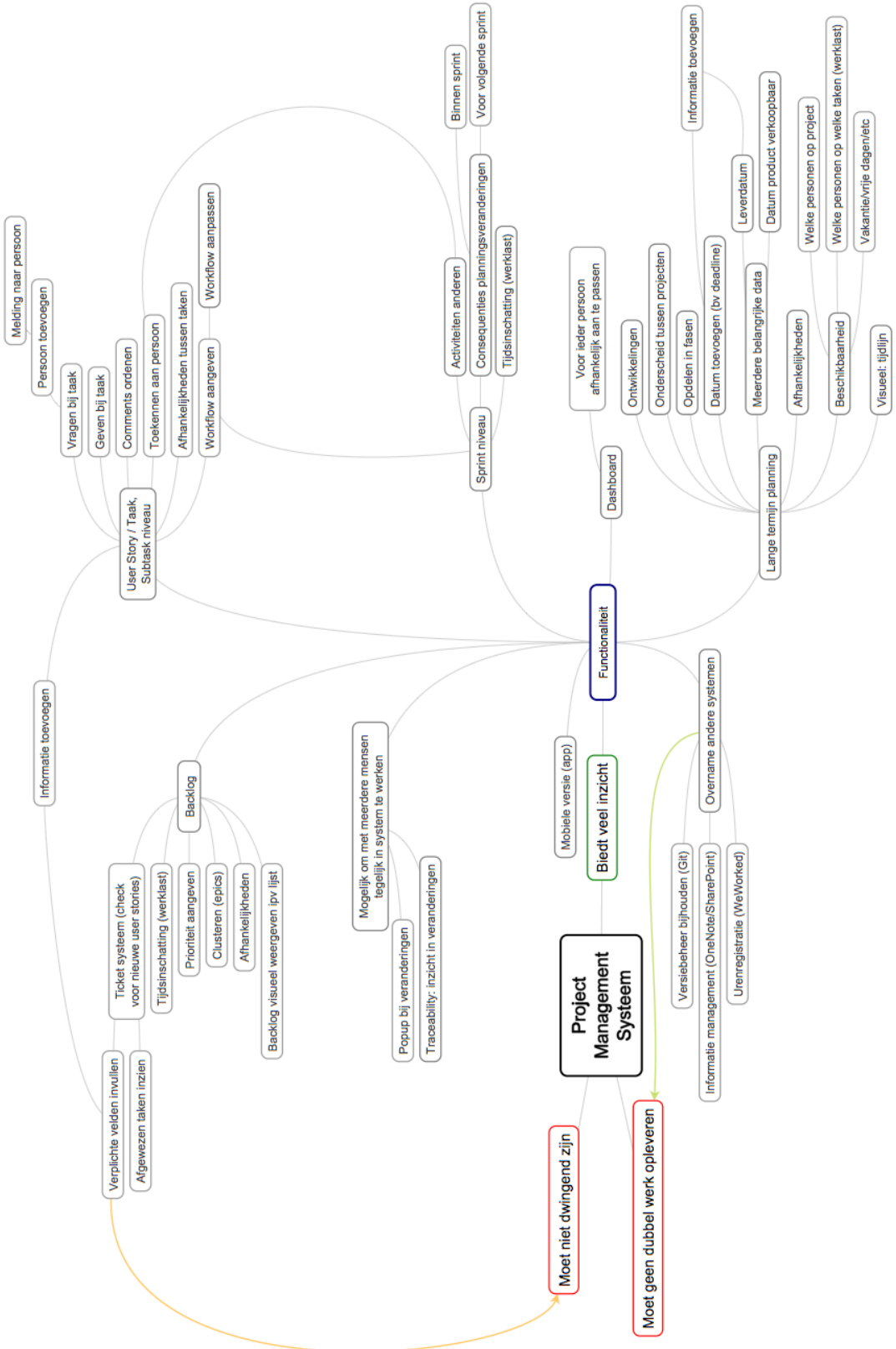
- Thema 1: Backlog
- Thema 2: Taak niveau
- Thema 3: Sprint niveau
- Thema 4: Lange termijn planning
- Thema 5: Overig
- Voor ieder thema:
    - Loop de functionaliteit door en geef uitleg per item
    - Wil er iemand reageren op de items?
    - Kijk rond hoe mensen reageren. Vraag bij opvallende reacties of diegene zijn/haar gedachten kenbaar wil maken

- Mist er informatie?
- Bij een discussie: vraag om verschillende meningen en zienswijze
- Vat alle informatie die besproken is samen

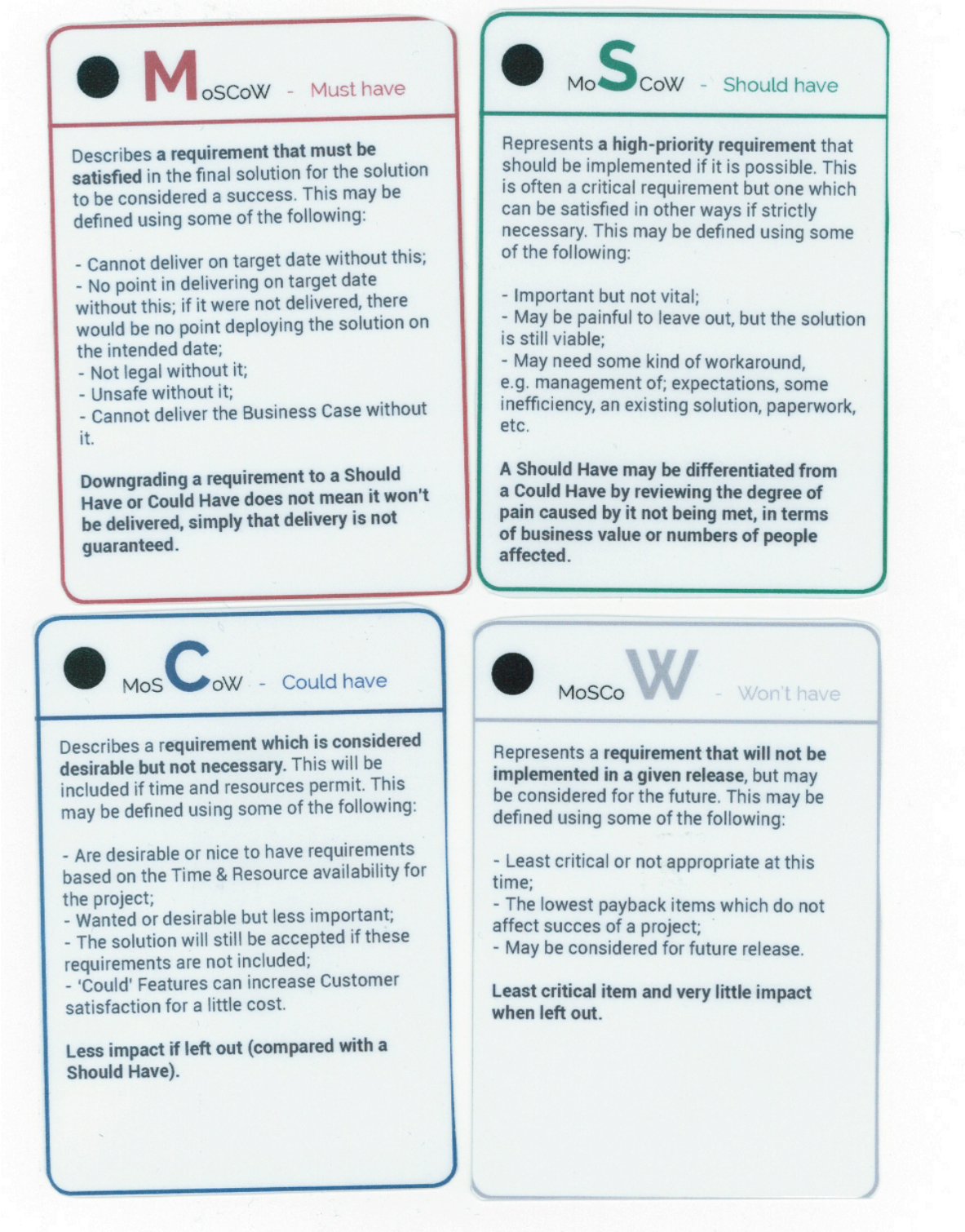
Einde van interview:

- Laatste vraag: Wil er nog iemand iets toevoegen of bespreken rondom dit onderwerp wat bijvoorbeeld nog niet besproken is?
- Bedank participanten voor het meedoen
- Stop opnamen en zorg dat deze goed zijn opgeslagen

Appendix D. Mind Map V2



Appendix E. MoSCoW Cards



Appendix F. Python code for heatmap

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline

# Read the excel sheet
mat = pd.read_excel('1.xlsx', sheetname='Average')

# Fill missing values (nan) with 0
mat = mat.fillna(0.0)

# Make matrix symmetrical
mat_symm = mat + mat.T

# Set back the diagonal that became 2 during addition
mat_symm = mat_symm.replace(2.0, 1.0)

# Plot the heat map
cmap = sns.cubehelix_palette(as_cmap=True, rot=-.3, light=1)
cg = sns.clustermap(mat_symm, metric="correlation", cmap=cmap)
plt.setp(cg.ax_heatmap.yaxis.get_majorticklabels(), rotation=0)
plt.setp(cg.ax_heatmap.xaxis.get_majorticklabels(), rotation=90)
plt.show()
```

Appendix G. Ontology: content-labels throughout the system

Label	Description
Aanleiding	Uitleg over het doel/de reden/klant-vraag waarom de aangemaakte user story of taak van belang is
Acceptatie criteria	Criteria waaraan het systeem moet voldoen om een user story/taak succesvol af te ronden
Behuizing maatvoering	Afmetingen van de behuizing (omhulsel van de elektronica)
Behuizing randvoorwaarden	Technische eisen voor gebruik, Printed Circuit Board afmetingen, waterdicht, inpotten
Brainstormsessie	Ideeën, notulen, tekeningen, etc. die uit de brainstormsessie zijn gekomen
Bug reports	Informatie rondom de bug; waarneming, in welke condities komt het voor, waar in de code, welke taken hebben hier waarschijnlijk mee te maken, hoe te reproduceren, impact, frequentie, versie
Doelgroep	Informatie rondom de groep mensen waarop het product zich richt
Git flow ID	Taaknummer/versienummer om een taak en Git te koppelen (MOT 1 etc)
Informatie van derden	Informatie rondom met wie er binnen Elitac over gesproken is, is er met de leverancier gesproken, contact gehad met nieuwe partij?
Kleurgebruik	Gebruikte kleuren binnen het grafisch ontwerp van bv applicatie/website (CMYK/RGB/HEX)
Logbestanden	(Automatisch gegenereerde) data die voortkomt uit bv testen
Logo’s, iconen en plaatjes	Bestaande logo’s, iconen en plaatjes
Onderzoeksmethode	Procedure van testen
Onderzoeksresultaten	Beschrijving van de resultaten die uit het onderzoek zijn gekomen, zijn er nieuwe materialen getest, etc.
Opdrachtgever	Informatie over het bedrijf/persoon die de opdracht geeft tot het ontwikkelen van een product
PCB afmetingen	Hoogte, breedte, dikte, hoeveelheid lagen, hoogste component op een printed circuit board. Enkelzijdig of beide kanten bestuukt? Kritieke componenten die invloed hebben op anderen (antennes etc.)

PCB functie	Beschrijving van de functionaliteit van een printed circuit board (technische specificaties)
PCB randvoorwaarden	Voorwaarden waaraan een printed circuit board moet voldoen, bv behuizing afmetingen
Personas	Beschrijvingen van fictieve personen die tot de doelgroep behoren. Deze kunnen inzicht geven in doelen, attitudes, demografische karakteristieken, wensen, informatie rondom de interactie met het product, etc.
Promo teksten	Inhoud die geplaatst moet worden op bv de website, folders, etc.
Schetsen	Ideeën die visueel zijn weergegeven
Software randvoorwaarden	Voorwaarden waaraan de software moet voldoen
Stretch afmetingen	Afmetingen van de stretch
Stretch randvoorwaarden	Voorwaarden zoals afmetingen, manier waarop de koper door de PU loopt
Technische documentatie	Bestanden/handleiding/referenties/websites met technische achtergrond informatie
Textiel maatvoering	Maatverdeling en maatvoering van de textieldelen
Textiel patronen	Uitslag voor textiel (knip sjabloon)
Textiel randvoorwaarden	Voorwaarden waaraan de textiel moet voldoen
User journeys	Weergave van de interactie van de gebruiker met het product
Voorbeeld code	Bron code die als voorbeeld dient voor de oplossing van het probleem

Appendix H. Checklist Requirements Jira

User Story / Task level

No.	Description	Acceptance Criteria	Priority	Status
1	Add a user story/task	• The title of the story/task can be inserted • A minimum of one text field is available to add information regarding the story/task	M	OK
5	Insert an indication of time to solve/execute the user story/task	• Professionals can insert an indication of time for each story/task • The time is described in points/minutes/hours	M	OK
7	Add dependencies to user stories/tasks	• A user stories or task that is dependent of another story/task can be linked	M	OK
8	See dependent user stories/tasks within the story/task	• For each story/task, dependent stories and/or tasks are visible for all professionals	M	OK
12	See which professional works on the user story/task	• Stories and tasks can be assigned to one or more professionals	M	OK
2	Add extra fields for information	• The professionals can add a minimum of 4 extra fields to insert information	S	OK
3	Customize existing information fields	• The properties of existing fields to insert information can be changed • Existing fields to insert information can be hidden	S	OK
4	Make fields for inserting information mandatory when adding a user story or task	• A minimum number of characters at mandatory fields can be added • The story/task can not be added without completing the mandatory fields	S	OK
9	See the subtasks within a story/task	• All added subtasks within a story or task are visible	S	OK
10	Ask someone to review a certain story/task	• A person can ask a specific person to review a specific story/task • The person who is asked to give feedback is notified	S	OK
11	Give feedback on a certain story/task	• The professional can give feedback to a story/task • The person who asked for feedback is notified when the task is reviewed	S	OK
6	Upload files to the user story/task	• More than one document/picture/etc. can be uploaded	C	OK
13	Organize comments within a user story/task	• Comments can be organized in a minimum of two categories	C	-

Subtask level

No.	Description	Acceptance Criteria	Priority	Status
17	See what professional works on the subtask	• Subtasks can be assigned to one or more professionals	M	OK

14	Add a subtask to a story / task	• The title of the subtask can be inserted • A minimum of one text field is available to add information regarding the subtask	S	OK
15	Insert an indication of time to execute the subtask	• Professionals can insert an indication of time for each subtask in the sprint • The time is described in points/minutes/hours	S	OK
18	Ask someone to review a certain subtask	• A person can ask a specific person to review a specific subtask • The person who is asked to give feedback is notified	S	OK
19	Give feedback on a certain subtask	• The professional can give feedback to a subtask • The person who asked for feedback is notified when the task is reviewed	S	OK
20	Add dependencies between subtasks	• A subtask that is dependent of another subtask can be linked	S	OK
16	Upload files to the subtask	• A documents/pictures can be uploaded	C	OK
21	See dependencies between subtasks	• For each subtask, one can see what other subtasks are dependent	C	OK
22	Organize comments within a subtask	• Comments can be organized in a minimum of two categories	C	-

Backlog

No.	Description	Acceptance Criteria	Priority	Status
23	See an indication of time to solve/execute a story or task in the backlog	• For each story/task in the backlog, the assigned time is shown	M	OK
24	See a hierarchy of importance of user stories/tasks in the backlog	• Professionals can prioritize user stories/tasks in the backlog • The most important stories/tasks are shown at the top of the backlog	M	OK
25	Have an overview of stories/tasks that address the same objective in the backlog	• It should be possible to cluster user stories/tasks in the backlog • Professionals can see all the clustered stories together	S	OK
26	See dependencies between stories/tasks in the backlog	• In the backlog, one can see what story/task is dependent of others	S	OK
27	See the backlog visually instead of a list	• The professional is able to choose between different ways in which the backlog can be presented	S	OK
28	Have a ticket system to check newly added items for in the backlog	• The newly added item is send for approval to the person who manages the ticket system • Person who manages the ticket system is notified that a new item is added and up for approval • The item can be accepted or declined	S	OK
29	Insert reasoning for declined items	• A minimum of one text field is available to add information about the decision	S	OK
30	Have an overview of the items	• Declined tasks are stored in a second backlog	S	-

	that were declined after being added to the ticket system			

Sprint level

No.	Description	Acceptance Criteria	Priority	Status
31	Add a workflow status to a story/task	• Not more than one status can be chosen for a story/task/subtask	M	OK
33	See the workflow for user stories/tasks within a sprint overview	• All professionals can see the assigned statuses of user stories/ tasks/subtasks within the sprint overview	M	OK
36	See what professionals are working on the subtasks within a sprint overview	• Professionals that are assigned to a subtasks are visible in the sprint overview	M	OK
37	See the progress of the sprint	• Professionals are able to see the progress of the overall sprint-status	M	OK
32	Add a workflow status to a subtask	• Not more than one status can be chosen for a subtask	S	OK
34	See the workflow for subtasks within a sprint overview	• All professionals can see the assigned statuses of subtasks within the sprint overview	S	OK
39	See the consequences for the next sprint when the planning changes	• Professionals can see if a sprint embodies a task that is related to a not yet completed task	S	OK
40	Have an indication of time regarding tasks within a sprint overview	• The assigned time is shown for each item in the sprint overview • The time is described in points/minutes/hours	S	OK
35	Customize the workflow stages	• Workflow stages can be added • A workflow stage name can be changed • Workflow stages can be removed	C	OK
38	See the consequences within a sprint overview when the planning changes	• In the sprint overview, one can see the feasibility of the sprint when the planning changes	C	OK

Long term planning

No.	Description	Acceptance Criteria	Priority	Status
41	See the progress in the long term planning (developments)	• The active sprint is displayed in the long term planning • Completed sprints are displayed in the planning • Professionals can see the sprints related to the near future	M	OK**
42	Distinguish different projects in the long term planning	• Different projects can be displayed • The starting date is shown for each project • The ending date is shown for each project	M	OK**
44	See dependencies in the long term planning	• Professionals can see what sprints are related to not yet finished sprints	M	-
45	Add an important date to a project into the long term planning	• Professionals are able to add an important date like a deadline to a project	M	OK**

43	Use phases within the long term planning	• The long term planning can be divided into different phases • Multiple sprints can be assigned to a phase	S	-
46	Add multiple important dates to projects into the long term planning	• A minimum of three important dates can be added to a project planning	S	OK**
47	Add information to important dates	• A minimum of one text field is available to add information regarding the date	S	OK**
49	See the workload of a professional in the long term planning	• For each professional the workload is shown during the different sprints in the planning	S	-
50	Add absences of professionals to the long term planning	• A name for the absence can be inserted (vacation, etc.) • Absences can be added for professionals separately	S	OK**
52	See a visual overview of the long term planning	• It can be chosen to see the long term planning to be presented visually	S	OK**
48	See in the long term planning what professionals are on a project	• Professionals can see who of the professionals is working in a sprint	C	-
51	See the availability of a professional in the long term planning	• Absences are displayed in the long term planning	C	OK**

Others

No.	Description	Acceptance Criteria	Priority	Status
55	Work with multiple professionals in the system at the same time	• All professionals can work in the system at the same time • All changes made in the system are carried through so the system is up to date	M	OK
57	Track changes made in the system	• Recently made changes are traceable • All professionals can see into recently made changes	M	OK
53	Have a dashboard with information	• When starting up the system, a dashboard is shown • Each professional has his/her own dashboard	S	OK
54	Customize the dashboard for each professional separately	• Every professional is able to add items on his/her own dashboard • Every professional can change items on his/her own dashboard • Every professional can remove items on his/her own dashboard	S	OK
56	Be up to date on relevant changes made while working in the system	• Professionals working in the system see when a change is made concerning the sprint one is working on • One can also see what change is made	S	OK
58	See progress of system versions	• A professional can see the deadline • It is visible how many user stories/tasks are done till completion • One can see when not yet started tasks are	S	OK

		scheduled		
59	Manage system versions	• Software versions can be managed within the system	C	-***
60	Document information in the system	• Information can be documented in the system • The information is accessible outside of the story/task it is added to	C	OK**
61	Register work hours	• Work hours can be registered	C	OK
62	Have a phone application of the system	• A phone application of the system is available for every common mobile platform	C	OK

* Plugin

** Other program compatibility: Confluence + Calendars

*** Cannot be managed within the system, but there is a plugin to link the two systems

Appendix I. Checklist Requirements Taiga

User Story / Task level

No.	Description	Acceptance Criteria	Priority	Status
1	Add a user story/task	• The title of the story/task can be inserted • A minimum of one text field is available to add information regarding the story/task	M	OK
5	Insert an indication of time to solve/execute the user story/task	• Professionals can insert an indication of time for each story/task • The time is described in points/minutes/hours	M	OK
7	Add dependencies to user stories/tasks	• A user stories or task that is dependent of another story/task can be linked	M	-
8	See dependent user stories/tasks within the story/task	• For each story/task, dependent stories and/or tasks are visible for all professionals	M	-
12	See which professional works on the user story/task	• Stories and tasks can be assigned to one or more professionals	M	OK
2	Add extra fields for information	• The professionals can add a minimum of 4 extra fields to insert information	S	OK
3	Customize existing information fields	• The properties of existing fields to insert information can be changed • Existing fields to insert information can be hidden	S	OK
4	Make fields for inserting information mandatory when adding a user story or task	• A minimum number of characters at mandatory fields can be added • The story/task can not be added without completing the mandatory fields	S	-
9	See the subtasks within a story/task	• All added subtasks within a story or task are visible	S	OK
10	Ask someone to review a certain story/task	• A person can ask a specific person to review a specific story/task • The person who is asked to give feedback is notified	S	OK
11	Give feedback on a certain story/task	• The professional can give feedback to a story/task • The person who asked for feedback is notified when the task is reviewed	S	OK
6	Upload files to the user story/task	• More than one document/picture/etc. can be uploaded	C	OK
13	Organize comments within a user story/task	• Comments can be organized in a minimum of two categories	C	-

Subtask level

No.	Description	Acceptance Criteria	Priority	Status
17	See what professional works on the subtask	• Subtasks can be assigned to one or more professionals	M	OK

14	Add a subtask to a story / task	• The title of the subtask can be inserted • A minimum of one text field is available to add information regarding the subtask	S	OK
15	Insert an indication of time to execute the subtask	• Professionals can insert an indication of time for each subtask in the sprint • The time is described in points/minutes/hours	S	-
18	Ask someone to review a certain subtask	• A person can ask a specific person to review a specific subtask • The person who is asked to give feedback is notified	S	OK
19	Give feedback on a certain subtask	• The professional can give feedback to a subtask • The person who asked for feedback is notified when the task is reviewed	S	OK
20	Add dependencies between subtasks	• A subtask that is dependent of another subtask can be linked	S	-
16	Upload files to the subtask	• A documents/pictures can be uploaded	C	OK
21	See dependencies between subtasks	• For each subtask, one can see what other subtasks are dependent	C	-
22	Organize comments within a subtask	• Comments can be organized in a minimum of two categories	C	-

Backlog

No.	Description	Acceptance Criteria	Priority	Status
23	See an indication of time to solve/execute a story or task in the backlog	• For each story/task in the backlog, the assigned time is shown	M	OK
24	See a hierarchy of importance of user stories/tasks in the backlog	• Professionals can prioritize user stories/tasks in the backlog • The most important stories/tasks are shown at the top of the backlog	M	OK
25	Have an overview of stories/tasks that address the same objective in the backlog	• It should be possible to cluster user stories/tasks in the backlog • Professionals can see all the clustered stories together	S	OK
26	See dependencies between stories/tasks in the backlog	• In the backlog, one can see what story/task is dependent of others	S	-
27	See the backlog visually instead of a list	• The professional is able to choose between different ways in which the backlog can be presented	S	OK
28	Have a ticket system to check newly added items for in the backlog	• The newly added item is send for approval to the person who manages the ticket system • Person who manages the ticket system is notified that a new item is added and up for approval • The item can be accepted or declined	S	OK
29	Insert reasoning for declined items	• A minimum of one text field is available to add information about the decision	S	OK
30	Have an overview of the items	• Declined tasks are stored in a second backlog	S	-

	that were declined after being added to the ticket system			

Sprint level

No.	Description	Acceptance Criteria	Priority	Status
31	Add a workflow status to a story/task	• Not more than one status can be chosen for a story/task/subtask	M	OK
33	See the workflow for user stories/tasks within a sprint overview	• All professionals can see the assigned statuses of user stories/ tasks/subtasks within the sprint overview	M	-
36	See what professionals are working on the subtasks within a sprint overview	• Professionals that are assigned to a subtasks are visible in the sprint overview	M	OK
37	See the progress of the sprint	• Professionals are able to see the progress of the overall sprint-status	M	OK
32	Add a workflow status to a subtask	• Not more than one status can be chosen for a subtask	S	OK
34	See the workflow for subtasks within a sprint overview	• All professionals can see the assigned statuses of subtasks within the sprint overview	S	OK
39	See the consequences for the next sprint when the planning changes	• Professionals can see if a sprint embodies a task that is related to a not yet completed task	S	-
40	Have an indication of time regarding tasks within a sprint overview	• The assigned time is shown for each item in the sprint overview • The time is described in points/minutes/hours	S	OK
35	Customize the workflow stages	• Workflow stages can be added • A workflow stage name can be changed • Workflow stages can be removed	C	OK
38	See the consequences within a sprint overview when the planning changes	• In the sprint overview, one can see the feasibility of the sprint when the planning changes	C	OK

Long term planning

No.	Description	Acceptance Criteria	Priority	Status
41	See the progress in the long term planning (developments)	• The active sprint is displayed in the long term planning • Completed sprints are displayed in the planning • Professionals can see the sprints related to the near future	M	-
42	Distinguish different projects in the long term planning	• Different projects can be displayed • The starting date is shown for each project • The ending date is shown for each project	M	-
44	See dependencies in the long term planning	• Professionals can see what sprints are related to not yet finished sprints	M	-
45	Add an important date to a project into the long term planning	• Professionals are able to add an important date like a deadline to a project	M	-

43	Use phases within the long term planning	• The long term planning can be divided into different phases • Multiple sprints can be assigned to a phase	S	-
46	Add multiple important dates to projects into the long term planning	• A minimum of three important dates can be added to a project planning	S	-
47	Add information to important dates	• A minimum of one text field is available to add information regarding the date	S	-
49	See the workload of a professional in the long term planning	• For each professional the workload is shown during the different sprints in the planning	S	-
50	Add absences of professionals to the long term planning	• A name for the absence can be inserted (vacation, etc.) • Absences can be added for professionals separately	S	-
52	See a visual overview of the long term planning	• It can be chosen to see the long term planning to be presented visually	S	-
48	See in the long term planning what professionals are on a project	• Professionals can see who of the professionals is working in a sprint	C	-
51	See the availability of a professional in the long term planning	• Absences are displayed in the long term planning	C	-

Others

No.	Description	Acceptance Criteria	Priority	Status
55	Work with multiple professionals in the system at the same time	• All professionals can work in the system at the same time • All changes made in the system are carried through so the system is up to date	M	OK
57	Track changes made in the system	• Recently made changes are traceable • All professionals can see into recently made changes	M	OK
53	Have a dashboard with information	• When starting up the system, a dashboard is shown • Each professional has his/her own dashboard	S	OK
54	Customize the dashboard for each professional separately	• Every professional is able to add items on his/her own dashboard • Every professional can change items on his/her own dashboard • Every professional can remove items on his/her own dashboard	S	-
56	Be up to date on relevant changes made while working in the system	• Professionals working in the system see when a change is made concerning the sprint one is working on • One can also see what change is made	S	OK
58	See progress of system versions	• A professional can see the deadline • It is visible how many user stories/tasks are done till completion • One can see when not yet started tasks are	S	-

		scheduled		
59	Manage system versions	• Software versions can be managed within the system	C	-**
60	Document information in the system	• Information can be documented in the system • The information is accessible outside of the story/task it is added to	C	OK
61	Register work hours	• Work hours can be registered	C	OK*
62	Have a phone application of the system	• A phone application of the system is available for every common mobile platform	C	OK

* Plugin

** Cannot be managed within the system, but there is a plugin to link the two systems