POLDER VALLEY

UNIVERSITY OF TWENTE.

PLANNING AT POLDER VALLEY

Redesigning the planning process through qualitative research

Abstract

This document contains a Bachelor thesis for Industrial Engineering and Management. Through quantitative research the current situation is compared to an ideal situation. Then, through interventions and the design of a supportive tool the planning process is improved.

Eric P. Kamphuis

e.p.kamphuis@student.utwente.nl

Author:

Eric Pieter Kamphuis

Student number:

s1497510

Date:

04-09-2018

Studies:

Bachelor Industrial Engineering and Management

University of Twente:

First supervisor: *Dr. A.I. (Adina) Aldea, MSC*

Second supervisor: *Dr. M.E. (Maria) Iacob*

Polder Valley:

First supervisor: *N. (Nico) Kienhuis*

Second supervisor *P. (Peter) Klijndijk*

Contents

1 Introduction

In this chapter the company where the research takes place will be introduced first. Second, the problem is described. Then subsequently, the research objective, questions and approach, methodology, and an overview are presented.

1.1 Polder Valley

In this paragraph some information about Polder Valley is given as well as the motivation for the research.

1.1.1 Introduction

Polder Valley (PV) is a young company that focuses on the development and sales of software products that add value to IT-automation. The company has been founded as a UT spin-off from The Backbone (also an old UT spin-off). The Backbone is a part of the Invinitiv group together with IT2IT, ExplainiT and now Polder Valley. The Backbone as well as ExplainiT are still very involved when it comes to business development of Polder Valley. And they will ensure sales of developed products. At this moment Polder Valley focuses on a sole product, the Productivity Performer. This is their first product.

1.1.2 Research motivation

Not long-ago Polder Valley has started implementing Agile software development. Products they develop are due to a demand that has been discovered by The Backbone and ExplainiT. The first research question proposed by the company was “How can a product owner improve the product development process within Polder Valley?”. The product owner is a distinct role within Agile software development. However, it quickly became clear a thorough problem identification process was necessary. This question provided a starting point for the research as it pointed to an uncertainty regarding Agile software development at Polder Valley.

So, the initial problem describes a knowledge gap regarding adding a product owner to a development team. But, after further research on occurring problems it became clear the main problem was that the throughput time regarding software development was too low. This is very undesirable in the field of software development because it’s a field where the market can catch up to you very fast.

The problem owners are Nico Kienhuis (CEO Invinitiv) and Peter Klijndijk (Director the Backbone). They have the ambition to sell innovative software applications, wherefore a low throughput time in the software development process is required. People effected by possible changes are all the employees of Polder Valley, as it’s possible changes will be made regarding the planning structure, working or communicating within the company. The researcher’s role was between the software development team and the people responsible for the business development of Polder Valley. With an outside view the problem owners Nico Kienhuis and Peter Klijndijk hope to gain useful recommendations for improving Polder valley. Various resources were provided which can be found in appendix 9.1. Furthermore, a stakeholder analysis and a reflection regarding certain moral issues in appendices 9.2 and 9.3. Appendix 9.1-9.3 have been written from the perspective of the researcher.

1.2 Problem description

After an orientation phase where much information was collected about the company, there was still much uncertainty regarding the exact problems PV was experiencing. To gain an overview from different perspectives of the software development process within PV interview were held with (in this order) Gert Kienhuis, Diederik Bakker (part time developer and student), Peter Klijndijk and Nico Kienhuis. Similar questions were asked (slightly different regarding the position) to these participants regarding the way PV works, daily routines, the product they are working on, their view on software development and possible problems or challenges. After summarizing various problems/challenges mentioned, the identification of core problems was possible. This also led to rephrasing the initial action problem towards planned items. The main conclusions from the interviews can be found in appendix 9.4. In order to structure the search for a core problem to focus on a problem cluster was created. As described by Hans Heerkens, a problem cluster shows the different problems and their relations (Heerkans & van Winden, 2012, p. 46). The problem cluster was created by relating problems discovered during interviews. With a problem cluster one looks for causal relationships starting from the action problem as stated in consultation with my company supervisors. Once there doesn't seem to be a clear cause for a problem this problem can be identified as a core problem. In the problem cluster, the choice of a solvable core problem is also shown. The problem cluster is shown in Figure 1

Various versions were created, and this last version was shown to all the stakeholders asking for remarks regarding any problems or relations. All stakeholders agreed with my identification. The problem cluster led to the core problem, the absence of a documented planning/scheduling strategy. At the start of the research, all the planning and scheduling was being done via the expertise of employees. There was no clear support for choices made regarding the planning process. A concrete documented planning and scheduling strategy would enable PV to use historic data and review their own planning, leading to more knowledge and better planning in the future.

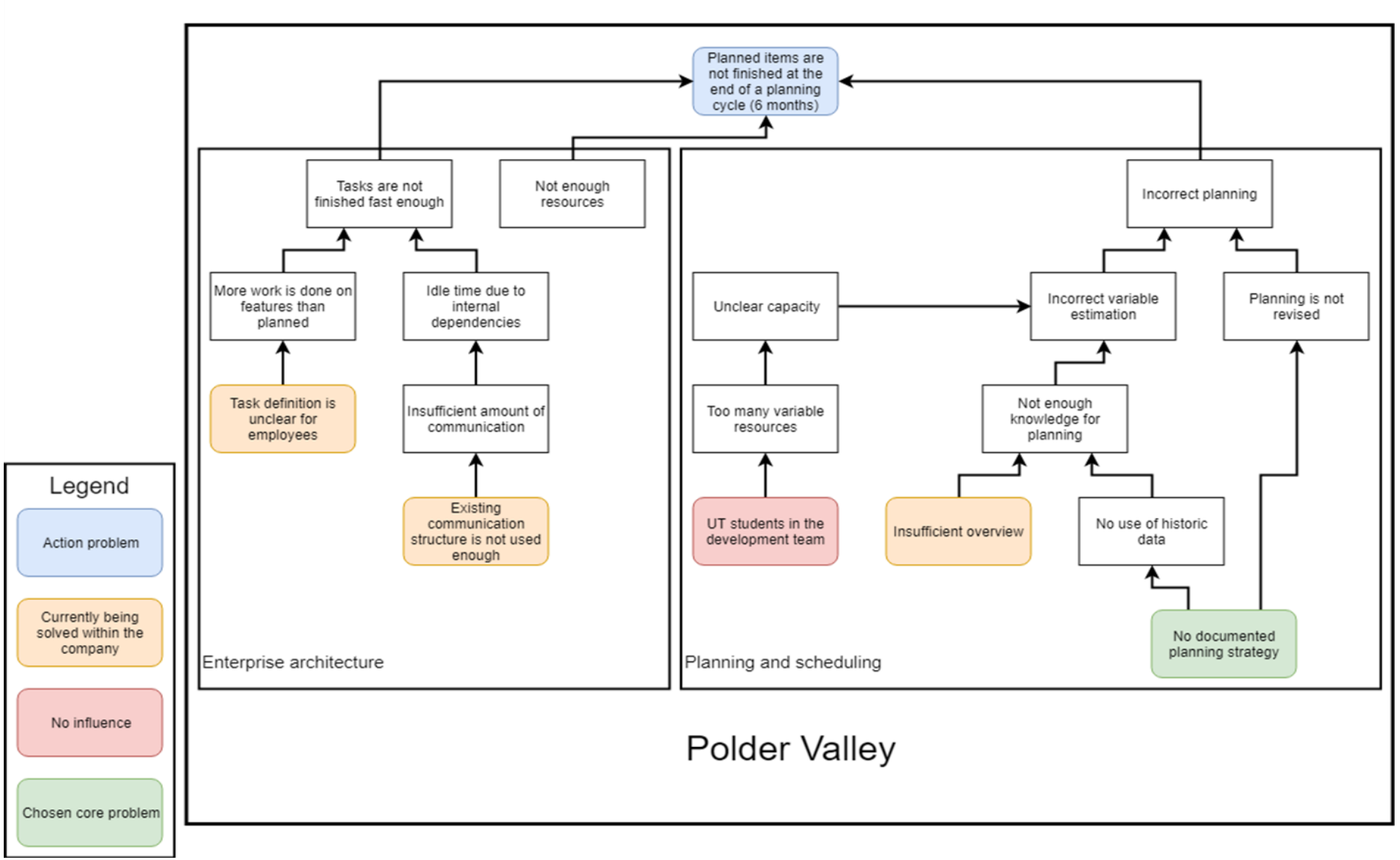


Figure 1 Problem cluster Polder Valley

1.3 Research objective

The research objective is to provide PV with insights in their current planning and scheduling process in order to suggest improvements. In the end there should be an improved and documented planning and scheduling process.

A concrete documented planning and scheduling strategy would enable PV to use historic data and review their own planning, leading to more knowledge and better planning in the future. Therefore, the focus is on this problem. There should be a clear system regarding the planning of a software development cycle consisting of sprints. In order to assess the starting point further, the key variables in Table 1 have been identified.

Table 1 Key variables overview

Key variable	Timeframe measured	Measurement
# items finished / # items planned	22-01 t/m 06-05	18/35
# extra items done during a sprint	22-01 t/m 06-05	0
# indicators used when planning and scheduling	06-05	4 (Priority, Effort, Capacity, Remaining work)
# items finished / # hours worked	22-01 t/m 06-05	18/732,75(students)
# predicted working hours / # hours worked (students)	20-03 t/m 06-05	254,14/354,75

In Table 1 there can be seen that a main problem is that items aren't finished during sprints. This is the case even though students work more hours than they predict. Students are considered in particular, as their working hours are variable.

Aiding in the research several deliverables have been created. Among the deliverables, three models have been created; an As-Is, Ideal and To-Be model. These models all show the planning and scheduling process in a different state. In paragraph 2.1 the theory behind the models is explained. Furthermore, a planning and scheduling tool has been designed which contains several dashboards which provide supporting information for the planning and scheduling process.

1.4 Research questions and approach

In this paragraph research questions and the approach to answer them are explained. The questions focus on creating the deliverables presented in the previous paragraph. In order to create these deliverables various research questions need to be answered. When sub-questions are defined, answering these questions leads to an answer of the main question. The collection of data is also assessed for each question.

1.4.1 As-Is model

This part will be addressed in chapter 3

1. What is the current planning/scheduling flow at Polder Valley?

To answer question 1, a **qualitative research** has been conducted among the **PV team**. This is because there was no existing data available. Data had to be collected in order to be able to answer this question. A **combination** of **interviews** with involved actors, **observation** of planning moments as well as the use of Visual Studio Team Services made it possible to map the current situation regarding the different **steps taken** and the **defined roles**. Besides this, the used **planning/scheduling variables** have been taken into account. All this information has been analyzed after creating a model using **BPMN**.

For the creation of the As-Is model data has been collected by observing various planning moments, interviewing the Product Owner and observing the process in Visual Studio Team Services. Created models have been verified with the Product Owner, because he has the most knowledge about the process.

1.4.2 Ideal model and gap analysis

This part will be addressed in paragraph 2.2 and chapter 4

2. What is the ideal planning/scheduling flow in an Agile software development environment?
    a. Which Agile software development methodologies can be used?
    b. How can Agile software development be used according to literature?
    c. How does Polder Valley want to perform planning/scheduling?
3. How does the current situation differ from the ideal situation?
    a. How does the As-Is model differ from the Ideal model?
    b. Which differences are unique for Polder Valley?

Regarding the ideal situation, the questions 2a and 2b have been answered using **qualitative research**. There is no 'best practice' Agile format available. Therefore, literature has been used in order to create an ideal overview that is applicable to PV. The subjects of this research have been a **combination** of **literary sources**, as well as **the management of PV** in order to answer questions 2c. The purpose of this has been to, once again, map the different **steps taken**, **defined roles** and **planning/scheduling variables**. To, also once again, create a model using **BPMN**. This led to two comparable situations. Then, a **gap-analysis** is performed in order to answer the questions 3a and 3b.

For the creation of the ideal model data has been collected through reviewing literary sources and an interview with the management of PV. The gap analysis doesn't require any data collection.

1.4.3 To-Be model

This part will be addressed in chapter 5

4. How can the planning process of Polder Valley be improved?
    a. How is the planning/scheduling done in comparable situations/companies?
    b. How can differences/sources of waste can be resolved using interventions?

c. Which variables should be used for the planning and scheduling at Polder Valley?

All questions above have been addressed in a **problem-solving research**. Here solutions are found to bridge the gap between the current and ideal situation. To serve as an inspiration for interventions, data is collected via **interviews** with external companies. **The** main research subjects are **the two previously created models** which have been subjected to **gap analysis**. The aim of the research is to minimize the **amount of waste**.

To do so, it has to be clear what is considered to be **waste**. This has been done in accordance with **the management of PV**. At the starting point of the research the main problem was a lack of structure. The focus could be on costs, amount of planning activities, throughput time or perhaps something else that becomes clear from the created models.

The problem-solving research' goal is to design suitable interventions. This knowledge has been combined with **literary sources**.

When it was clear which interventions are applicable and which variables can be made measurable. The models from the first two sections were combined to create a **To-Be model**. This is good for PV to have, especially when certain interventions or changes will take a long time to take effect.

For the design of the To-Be model the first two sections have used and data from TimeWriter has been analyzed. TimeWriter contains all the booked hours from the development team. Furthermore, interviews with external companies who develop software have been conducted to serve as an inspiration for interventions.

1.4.4 Planning/scheduling tool

This part will be addressed in chapter 6

5. How can the collected knowledge be used in a planning/scheduling tool?

The last part is an **applied research**, in order to leave PV with a supporting tool for their planning and scheduling process. It focuses on providing absent information. **Literary sources** have been studied in order to expand the range of possibilities for planning. **The systematic collection** in combination with key insights at this point in the research have led to a beneficial opportunity to implement a planning tool. The end result is a design of planning tool.

For the creation of the tool data has been collected via literary sources.

1.5 Research methodology

As a research methodology Design Science Research Methodology according to Peffers et al. (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007) has been applied. This methodology is based on a situation where an artifact is created as result of a research. It is also from the field of Management Information Systems in which software development environments are also situated. The Design Science Research Methodology (DSRM) Process Model can be seen in Figure 2.

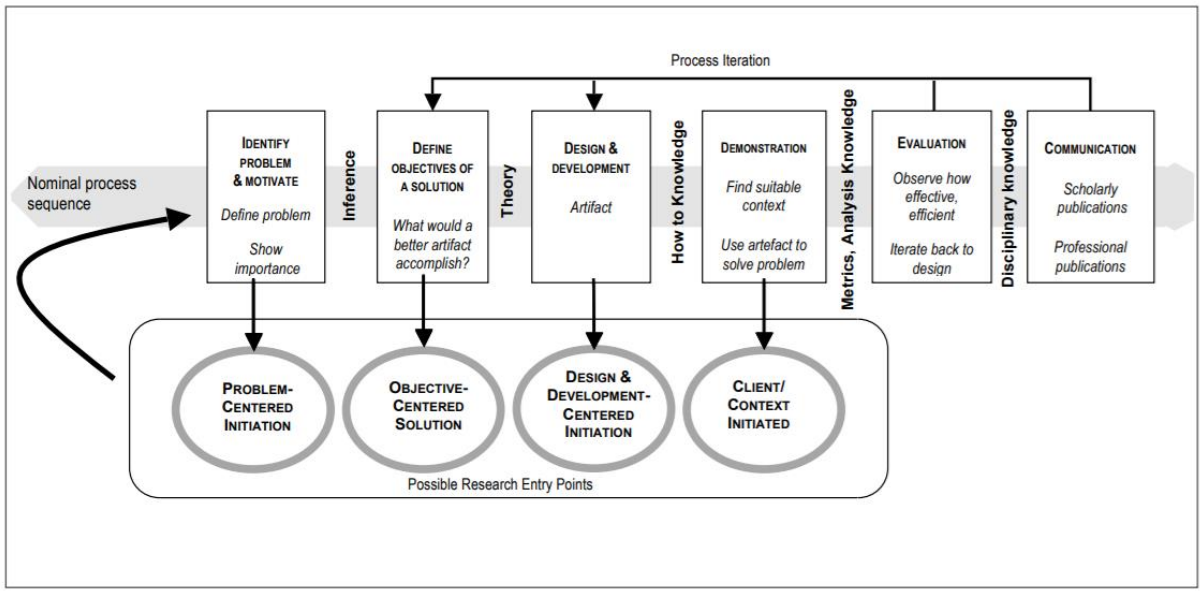


Figure 2 Design Science Research Methodology according to Peffers et al.

In the research all the steps have been followed, and a re-entry point has been used. The relationship of the model to this thesis is shown in Figure 3.

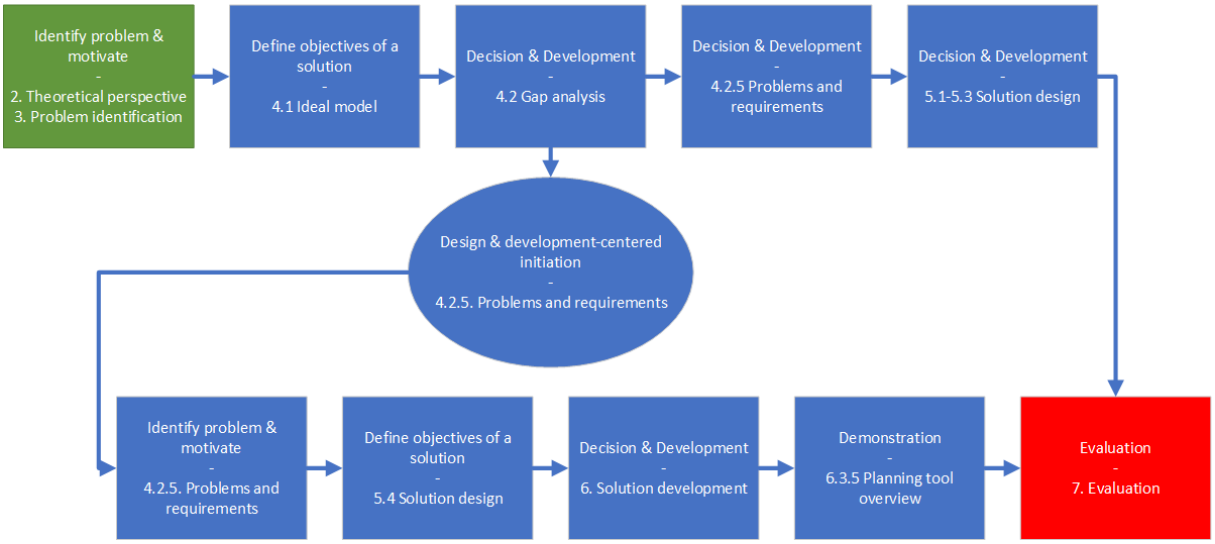


Figure 3 Use of DSRM for research at Polder Valley

Starting at the green block with the problem identification the model is followed until the red block at the end. In each block the phase from the DSRM is shown with the corresponding chapter or paragraph from this research. After the gap analysis, a re-entry point is used due to the uncovered necessity for a planning tool. This created a split where on the one hand direct interventions have been designed, and on the other hand a planning tool was designed.

1.6 Research overview

In this paragraph first of all, key constructs and variables of importance are explained. Then, an overview model of the research is presented.

1.6.1 Key constructs and variables

Key constructs and variables that are of importance during the research have been identified. Some definitions are based on literature. However, some constructs or variables are seldomly defined in literature, but rather just used. In this case a generic definition is provided which, just as all definitions, explains the way the construct or variable is viewed in relationship to the research.

Planning

For the research, planning is considered to be the choice of tasks to be done in a certain amount of time. There is no focus on the way of working on items or tasks because that is outside the scope of this research.

Scheduling

For the research, scheduling is considered to be the method of division of planned items and the estimation of capacity of employees.

Business Process Modeling

"The business process modelling space is organized using conceptual models. Business processes consist of activities whose coordinated execution realizes some business goal" (Weske, 2012, p. 73). This is how an overview of the processes can be created to suggest improvements.

As-is model

"Redesign projects for business processes usually start with analyzing and mapping an actual situation within an organization. This step is called "developing an AS-IS business process model" (Arkilic, Reijers, & Goverde, 2012, p. 1). This is a model made to get an overview of the current situation.

To-be model

From the as-is model a to-be model can be made. This model contains the desired state of the subject that has been modelled.

Ideal model

This model would be the ideal state. This could be the state of a comparable or a non-existent company. When a model is made of a non-existent company it could perhaps be made in according to the vision of management.

Gap analysis

"Gap analyzing is employed in order to identify the differences between baseline and target architecture based on architectural views" (Rouhani, Mahrin, Nikpay, Ahmad, & Nikfard, 2015, p. 7).

Waste

Waste is commonly known as unnecessary materials. However, it could also be lost time. In this research waste is considered lost time in relation to added value towards overall performance.

Planning/scheduling variable

This is a variable that has an influence on the planning or scheduling process and should therefore be considered in the research.

Conceptual model

A conceptual model is a model that shows the concepts and their relations of an application that could be designed. The goal is to enable the desired task-flow (Johnson & Henderson, 2012, p. 1). This should be made when an application is to be designed to serve as a planning tool.

Planning/scheduling tool

A planning tool is a support for planners to hold on to while planning for a certain amount of time. This could be an application or a framework that should help improve the process.

1.7 Model of research approach

Knowing which problem is being approach, what can be measured, and the intended deliverables A model has been created to clearly show an overview of the research. As the focus is on improvement, which can be done through waste reduction, a lean methodology is in place.

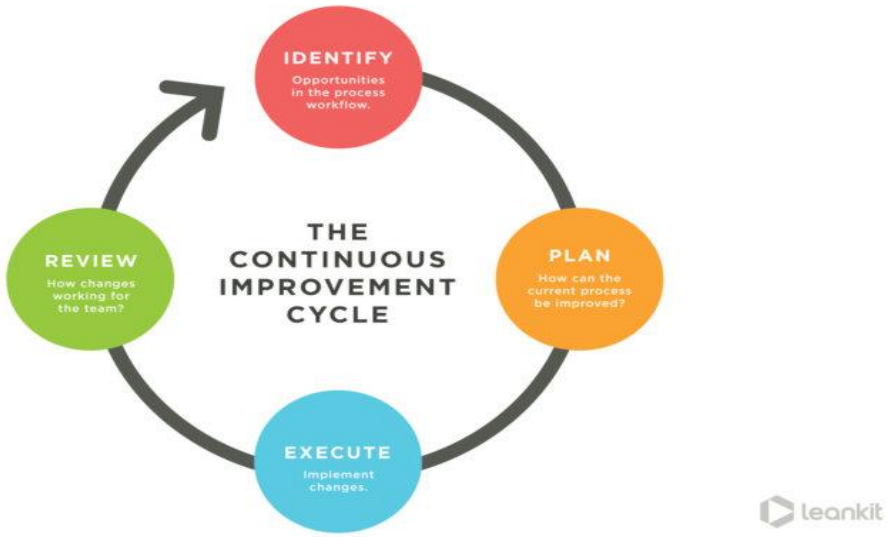


Figure 4 The continuous improvement cycle according to Leankit - https://leankit.com/learn/kanban/continuous-improvement/

As can be seen in Figure 4, opportunities are identified. In the research this has been done through the creation of flowcharts. Then, improvements are planned and executed. Of course, everything should be reviewed to check for success and learn from possible mistakes. Finally, the process starts over. The theoretical model below is representable for one cycle of continuous improvement. As the To-Be model becomes the As-Is model for the next cycle and new improvements can be made. The ideal model should remain, as this is the ideal (and perhaps unreachable) state.

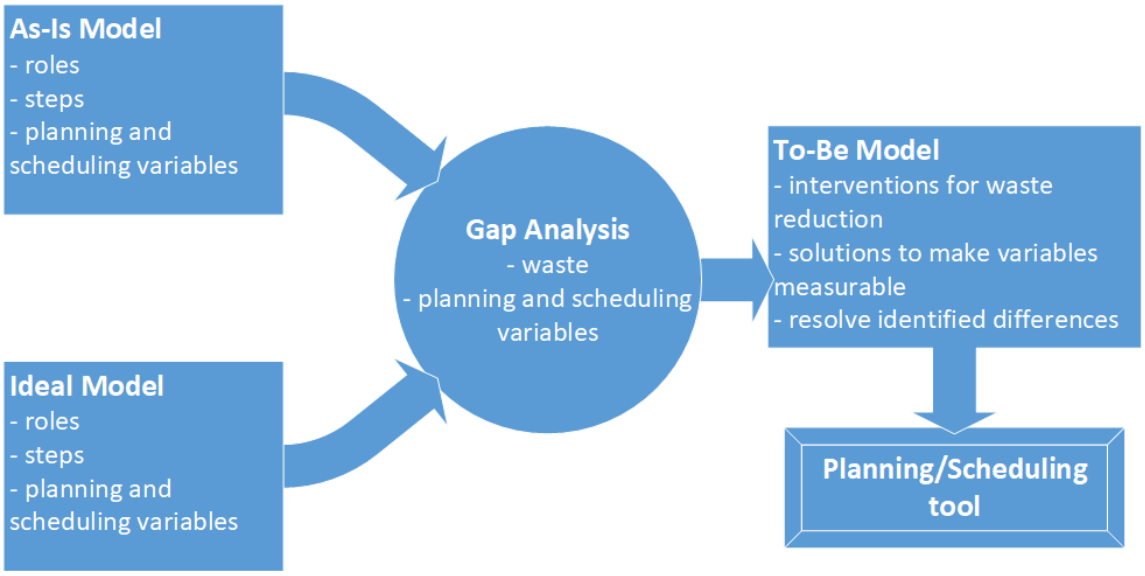


Figure 5 Research approach for research at Polder Valley

As can be seen in Figure 5, the research starts with mapping the current planning process (As-Us) at PV and the ideal situation is modelled (based on literature and the vision of management). While doing so, the focus was on mapping the roles, steps taken and the communication flow while planning. Second, differences have been analyzed while paying attention to waste and the handling of planning/scheduling variables. This way we could discover what the best ideal model for PV is (To-Be). The To-Be model has been made through suggesting interventions for waste reduction, solutions that lead to making certain variables measurable and resolve other identified differences which perhaps are relatively easy to solve. After this, planning literature has been combined with all the gained knowledge to create a planning tool to aid in the planning process.

2 Theoretical perspective

In this chapter theories and frameworks are explained and literature on Agile software development is presented.

2.1 Combining Business Process Model and Notation (BPMN), RACI and Value Stream Mapping (VSM)

In this paragraph we explain how different models have been combined in order to map the planning process at Polder Valley. First, the main model choice is explained. Then, in two preceding paragraphs, additions to this main model are motivated. These combined modeling techniques are applied in models shown in paragraphs 3.3.2 and 4.1.3. As explained in the resources in appendix 9.1 BiZZdesign has been provided as a tool to create an elaborate model containing all these elements.

2.1.1 BPMN as a Business Process Model

Models have been created showing the situations As-Is, Ideal and To-Be in chapters 3, 4 and 5. Business Process Model should be created according to a certain standard. There are various examples of models that can be used to map a process (Weske, 2012):

- Control Flow Patterns
- Petri Nets
- Event-driven Process Chains
- Workflow Nets
- Yet Another Workflow Language
- Graph-Based Workflow
- Business Process Model and Notation

Business Process Model and Notation is the newest Business Process Model Type and appears to be the standard for mapping business processes. BPMN was designed by the Business Process Management Initiative (BPMI).

> "The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation" (Object Management Group, 2009, p. 1).

The most recent version of BPMN is defined as BPMN2.0 which contains the newest constructs as designed by BPMI. As described in the report, BPMN2.0 consists of the following elements (Object Management Group, 2011):

1. Flow objects: The main graphical elements to define the behavior of a business process.
2. Data
3. Connecting objects: To connect flow objects.
4. Swimlanes: To group the primary modeling elements.
5. Artifacts

BPMN is a very useful tool as it is widely seen as the standard for mapping business processes. It is also possible to enrich BPMN with extra information as is explained in the following paragraphs.

2.1.2 RACI in the models

To get an overview of different roles people have within a process a RACI analysis has been applied. According to Kahn and Quraishi (Khan & Quraishi, 2014, p. 2) the letters in RACI stand for the following:

> **R**esponsible
>
> The person is assigned to get the work done. May delegate work or may be supported by others. Only one person is responsible, think of the lead or manager.
>
> **A**ccountable
>
> The person who will signoff on workpackages/ deliverables. Ultimately only one person, but often includes others (e.g. a sign-off document requiring signatures of multiple approvers)
>
> **C**onsult
>
> Those people who contribute to the work by providing information (consultancy), either by providing information or directly working at the direction of the person responsible.
>
> **I**nformed
>
> Those people who need to be Informed, but not contributing (i.e. *do not have active role*) .

Mapping these roles in a matrix per task and people involved creates an insightful overview. One can then apply horizontal or vertical analysis. An example of an insight when conducting vertical analysis could be: “A lot of R’s: Is it possible for the individual(s) to stay on top of so much? Can the activity be broken into smaller, more manageable chunks?” (Morgan, 2008, p. 2).

It isn’t uncommon to combine BPMN and RACI. This can be seen in literature. “Integrating a RASCI matrix into a BPMN model means enriching the process model with RASCI information, i.e., making the model RASCI-aware” (Cabanilas, Resinas, & Ruiz-Cortés, 2011, p. 2). In this case there is referred to a RASCI matrix which adds the S for Supported. In this research ‘supported’ roles aren’t considered in the analysis as no cases where identified where this role was present. Also, within the small organization that Polder Valley is, people are always consulted instead of merely supportive. Looking for an explanation of the difference between consulted and supported the following definition is found (Management Mania, 2016):

- **S** - **Support** - who provides support during the implementation of the activity / process / service?
- **C** - **Consulted** - who can provide valuable advice or consultation for the task?

Support appears to be necessary but not adding value whereas a consultation does hold value. At Polder Valley we have observed that every involvement in an activity is of added value. When this would not be the case I think it should be considered if an activity should be cancelled. If one thing would be supportive in the planning process it would be Visual Studio Team Services, which is an application and not considered to have a role.

2.1.3 VSM in the models

The flow of value throughout the process has also been taking into account using VSM. As explained before, we focus on minimizing the waste. VSM is an element of the Lean methodology which focuses on the reduction of waste. Taghizadegan describes it as following (Taghizadegan, 2006, p. 66):

> Value stream mapping will identify staff, information, and materials. It will also distinguish between value and nonvalue-added actions to improve value-added activities and reduce nonvalue-added actions. These are activities that external customers are willing to pay for. Value stream mapping is a visual flowchart that tracks materials, activities, and information required for the project. It is used to chart the existing and future process with a focus on value-added and nonvalue-added time.

This is what I will apply when conducting my research. It is important to make a clear distinction between activities that add value and nonvalue-added activities. When uncovering waste, I will take the following steps (Taghizadegan, 2006, p. 67):

1. Draw and complete a process flowchart for the project.
2. Distinguish all the job functions that add value to customer requirements, such as lower pricing, less defects, on-time delivery, and faster shipping.
3. Identify the nonvalue-added activities that do not add any value to the product-that is, inspection, counting, moving, reworking, or manual assembly in place of automation.
4. Select the activities that are important to be continued and actions that should be excluded or discontinued.

Although this is meant to be for manufacturing environments. One can say a software development environment is very similar. The main difference is the machines are developers. And therefore, they are slightly more difficult to control and predict. These steps relate to the research design. Step 1 is performed when creating an As-Is model and an Ideal model. Step 2 and 3 are performed during the gap analysis. And step 4 leads to a To-Be model. Therefore, VSM has been applied in this research.

It is important to choose correct value measurements. Value can, for example, take into account various times or costs. For this research, the focus has been on processing time as a cost, and an operationalized amount of value added to overall performance as a benefit for activities.

Combining BPMN with RACI and VSM has proven to be important because at the starting point it was unclear where exact problems could be pinpointed. The combination provided us with a broad perspective and overview regarding the planning and scheduling process. Which made it possible to suggest interventions related to different aspects of the process.

2.2 Agile software development

In this paragraph there is focus on available literature regarding Agile software development. First, an introduction into Agile software development is presented. Then, a systematic literature review is conducted regarding Agile methodologies. Second, Different planning levels shown. Third, an ideal execution of the Scrum process is portrayed. Last, literature is presented regarding planning and scheduling variables.

2.2.1 Introduction

At PV Agile Software Development is applied. This methodology has been of importance throughout the entire research. The past years, many software development teams have switched from the traditional method to this method. This is because this enables companies to gain feedback on their product often and ensure they are making something the end-user wants. In the following paragraphs, elements that are important in Agile software development are presented. The difference between the traditional method, or waterfall method, and Agile software development is shown in Figure 6.

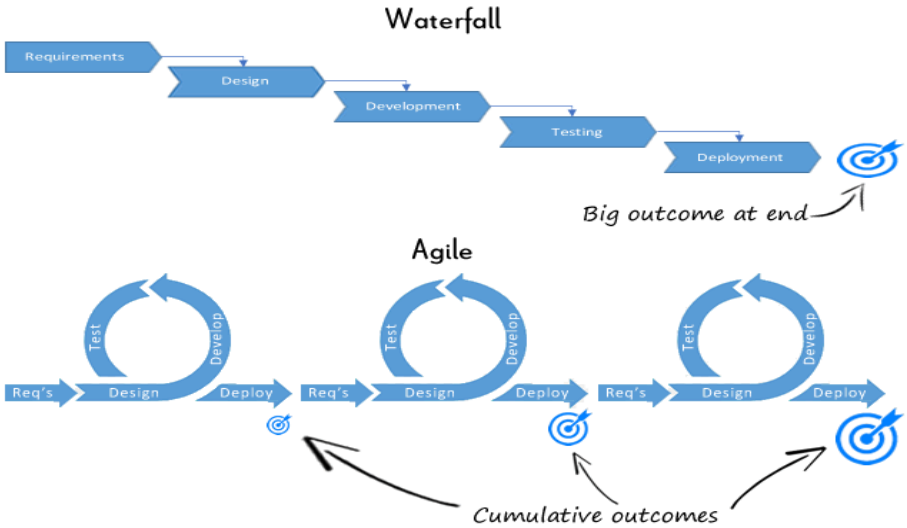


Figure 6 Waterfall vs. Agile according to Schaeffer- http://www.crmsearch.com/agile-versus-waterfall-crm.php

Working Agile means that instead of working a long time to one release date, the team works in sprints of 1-4 weeks (3 weeks at PV). After every sprint there should be new functionalities which can be shown to stakeholders to receive feedback. This way of working has a significant effect on the team. "Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto" (Agile Alliance, 2018):

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.

This manifesto has led to the twelve principles of agile which can be found in appendix 9.5. Further information on Agile software development follows in this chapter and in paragraph 3.2.

2.2.2 Which planning strategies or methods are applicable in the Agile software development process? – A systematic literature review

In order to create a model of the ideal situation I need to gain insights in existing planning practices in Agile software development environments. The reason it's important to research this topic is because so far, I haven't encountered this topic much. And when discovering what an ideal situation would be I should have an overview of all the possibilities. Also, when conducting interviews with experts this is important. The conceptual matrix, the protocol and the sources can be found in appendix 9.6.

The literature review has yielded a very good overview of planning strategies. However, after a closer review not all found concepts are a real strategy or methodology. There are several concepts that have been named more often. However, these are always named in (almost) the same manner. It is good to restructure the found concepts to create an overview. These concepts could all be of use for the design of a planning strategy. In Table 2 Summary Agile software development methodologies SLR they are structured with the amount of times they we're mentioned.

Table 2 Summary Agile software development methodologies SLR

Planning methodologies:	Count	Planning elements:	Count	Planning indicators:	Count
Communication focused planning	**1**	Business planning	**1**	Customer engagement	**1**
Extreme programming (XP)	**2**	Configuration management	**1**	Project planning responsibility	**1**
Intermediate approach	**1**	Planning games	**5**		
Iterative planning/adaptive planning/continuous planning	**11**	Forecasting based planning	**1**		
Model-based release planning	**2**	Increment planning	**1**		
MoSCoW prioritization	**1**	Multi-tiered planning/organizational planning	**2**		
Organic planning	**1**	Strategic planning	**1**		
Release iteration planning method	**1**	Replanning	**1**		
Scaled Agile Framework	**2**	Roadmapping	**1**		
Scrum model	**1**	Upfront planning	**3**		
Traditional planning	**8**				

As can be seen, the review mostly provides an overview rather than topics for discussion. Many concepts are only named in one paper. There could be concluded from this that there is no best practice of Agile.

Now we can discuss interesting findings encountered during the literature review. The first that can be concluded from the literature review is that almost every source mentions traditional planning in relation to iterative methodologies. Such as in S1 where is explained that "the Agile project management methodology has been widely used in recent years as a means to counter the dangers of traditional, front-end planning methods that often lead to downstream development pathologies" (p. 1). An example where this is agreed upon is where S2 states "Agile methods were developed to overcome several disadvantages related to traditional software development methodologies" (p. 9). So, it becomes clear that the traditional methods had disadvantages. Which meant it was time to move towards a new methodology.

Second, Upfront planning is also necessary according to S1 Agile shouldn't abandon upfront planning. "Certain factors, such as the size of the project, safety requirements and known future requirements, call for upfront planning even in Agile projects, whereas turbulent, high-change environments call for less upfront planning and a greater use of Agile methods" (p. 3). This is supported by S3 saying:

> Since the coordination is programmed through upfront planning with little communication, one person or a very small set of people, needs to have a deep insight into the full technical details of the entire software system in order to specify all details necessary for individual work packages and correct integration (p. 4).

So, it can be concluded Agile planning requires upfront planning of an iteration, followed by communication within a team. This means that even though upfront planning can be seen as a key part of traditional planning. Upfront planning is still of importance in Agile planning methodologies.

Third, planning games are mentioned more often. The most common form is planning poker, "a simplified form of the Wideband Delphi method, is popularly used to gain consensus on estimates on the relative sizes of requirements (i.e., user stories)" (S3, p. 13). However, planning poker must be used in the right context. As planning poker and comparable planning games "mention that customers will establish ''priorities'', without proposing a concrete technique to do so. Particularly, Planning poker only considers effort and not business value" (S11, p. 5).

Last, many common Agile elements appear to come from Extreme Programming. "Regarding planning XP works inwards doing a release planning, iteration planning and a daily stand-up" (S3, p. 18). Another aspect, planning poker, is "proposed by eXtreme Programming (p. 5)" according to S11. This leaves the question if this is perhaps a relatively older Agile methodology which is not explained often. A further research into this topic could be useful.

2.2.3 Planning levels

There are various levels of planning. As commonly known people usually have a global planning and a detailed planning. In Agile/Scrum this is no different. It can be stated that planning doesn't start at the planning of a release, but there should be vision at the very start that points towards the direction an organization wants to develop. This is supported by Cohn's "planning onion" (Cohn, 2006, p. 28):

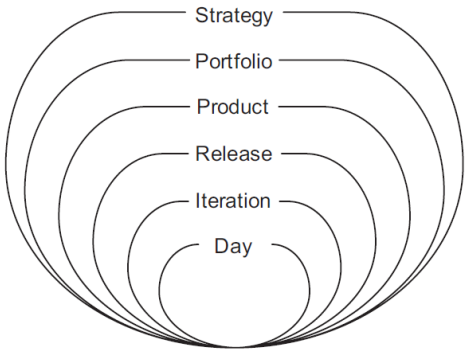


Figure 7 The planning onion according to Cohn

According to Cohn “most agile teams are only concerned with the three innermost levels of the planning onion” (Cohn, 2006, p. 28). This should be fine, as long as strategy, portfolio and product planning are considered in a good way by higher management. Daily planning is self-explanatory, iteration planning is the planning of a single sprint. Release planning is explained by Cohn in the following way:

> The goal of release planning is to determine an appropriate answer to the questions of scope, schedule, and resources for a project. Release planning occurs at the start of a project but is not an isolated effort. A good release plan is updated throughout the project (usually at the start of each iteration) so that it always reflects the current expectations about what will be included in the release.

Then, the path of the development of a single product should be planned. In the case of multiple products, a portfolio of products should also be managed and planned. Lastly, there should be a strategic planning for a company as a whole. Further on in my research I will mainly focus on the inner three levels. But, I will also assess the presence of planning in other levels.

It's commonly known that it is important to set goals for certain activities. As explained in the ‘Planning onion’ there are also higher levels of planning that are of importance. A good way to bring structure to this is through ‘The Golden Circle’ by Simon Sinek as shown in Figure 8.

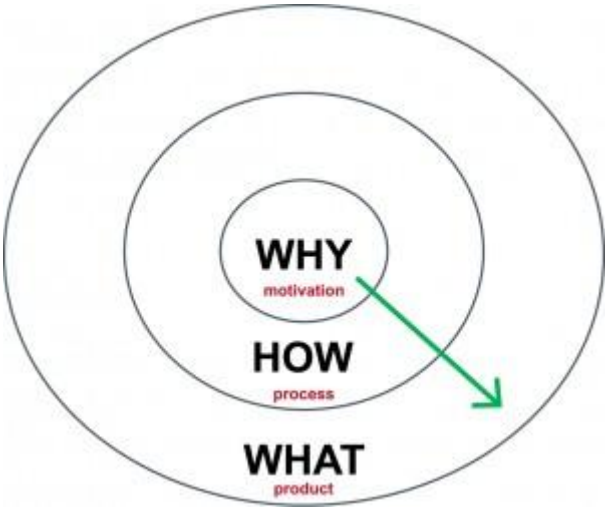


Figure 8 The Golden Circle by Simon Sinek as shown by BoscoAnthony - http://boscoanthony.com/the-golden-circle/

Here ‘Why’ is an explanation to what is being done. This can be seen as the vision statement. “Your vision statement gives the company direction. It is the future of the business, which then provides the purpose” (Skrabanek, 2017). Then, reviewing the ‘How’ part, this how you plan on reaching this vision. This also referred to as the mission statement. “Your mission statement drives the company. It is what you do/the core of the business, and from it come the objectives and finally, what it takes to reach those objectives” (Skrabanek, 2017). ‘What’ explains which product(s) are sold in order to achieve the previous. Then, each product can have its own vision. And throughout the planning process goals can be set for release and even sprint that adhere to this vision.

This provides us with a good framework to structure planning activities in created models. Every planning should start with strategy planning and flow down towards daily planning activities. This can be seen in created models in the following chapters.

2.2.4 Scrum process

Scrum is the most widely used Agile software development methodology. Polder Valley also applies this framework. Therefore, we take a closer look at how this methodology should be implemented. The best way to explain how Scrum work is by showing it in one image:

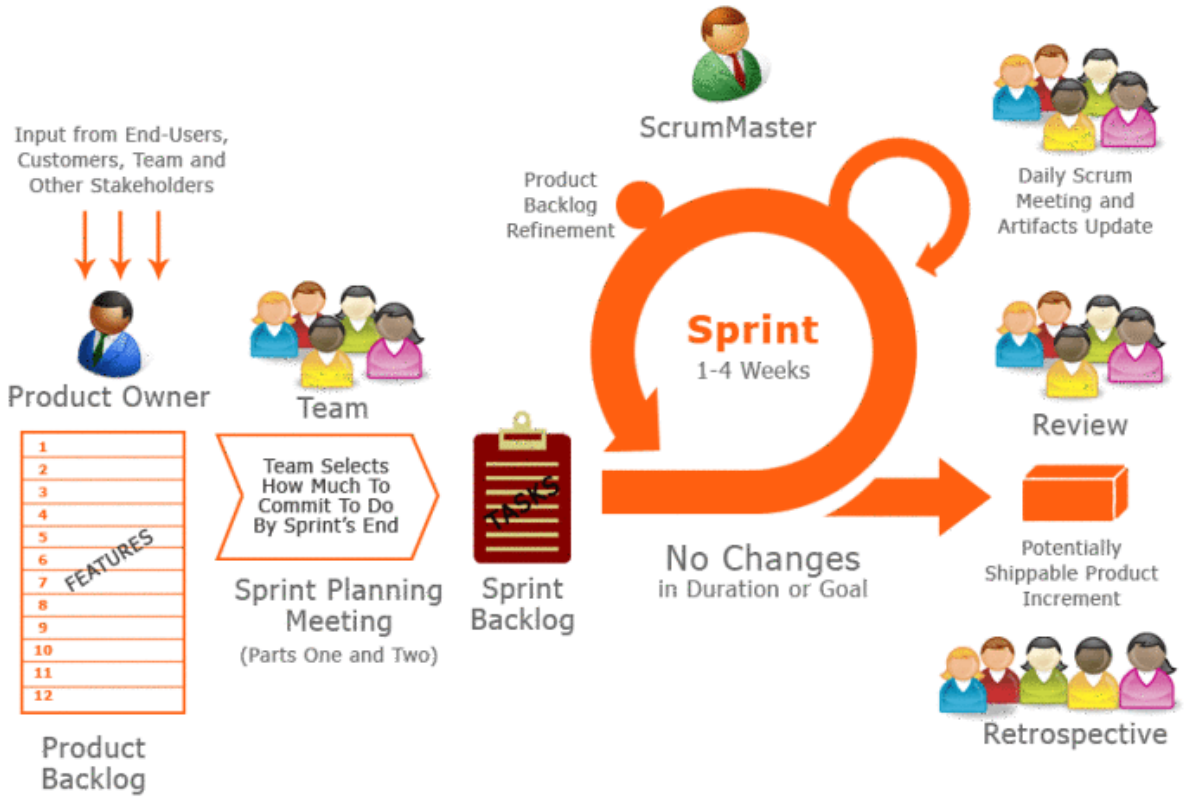


Figure 9 Scrum process according to Essential Solutions - http://www.essentialsln.com/agile-software-development/

Detailed definitions of various elements are shown in appendix 9.7. This overview provides a framework for the creation of the Ideal model in paragraph 4.1. We can see how the Product Owner manages the product backlog which is eventually passed on to the team as the sprint backlog. During the sprint various meetings occur which enable the team to continuously develop a product.

2.2.5 Planning and scheduling variables

In Agile software development there are many things that can influence the success of the team and the amount of work that can be done during a sprint. It would be good to have a clear insight in some of these variables. Therefore, several metrics have been identified which could be of good use. The aim of identifying these metrics is in order to be able to improve the process in the future in a more efficient way and aid in the planning process. "In Agile mindset, estimating is applied as a way to predict how much the team can get done to guide sprint planning—not as a target that should be achieved as closely as possible" (Kupiainen, Mäntylä, & Itkonen, 2015, p. 144). Kupiainen et al. identified the following metrics which they rated according to occurrences and importance (Kupiainen, Mäntylä, & Itkonen, 2015, p. 155):

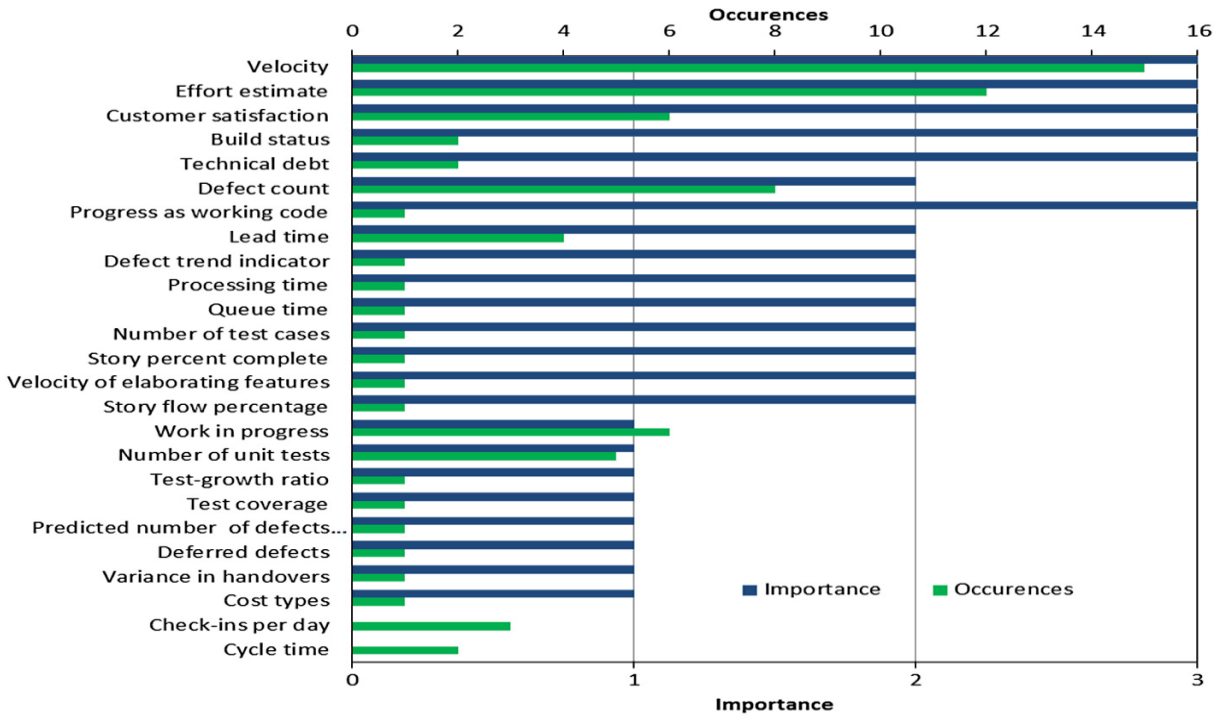


Figure 10 High influence metrics based on number of occurrences and perceived importance factor according to Kupiainen et al.

The measure of importance is of course somewhat subjective and was based on the following (Kupiainen, Mäntylä, & Itkonen, 2015, p. 156):

> Metrics were considered important if the author of the primary study or case employees praised the metric. Also, metrics were considered important if there were signs of continuous use of the metric. Furthermore, if metrics had positive correlation to important output measures such as project success, they were considered important.

Metrics can be divided in the categories; sprint and project planning, sprint and project progress tracking, understanding and improving quality, fixing software process problems, and motivating people. The metrics that are related to sprint and project planning are the following (Kupiainen, Mäntylä, & Itkonen, 2015, p. 152):

- Velocity
- Effort estimate
- Value to customer
- Lead time
- Task done/undone
- Task's expected done date
- Predicted number of defects
- Skills needed

Velocity refers to the following; the amount of "feature points developed per iteration" (Kupiainen, Mäntylä, & Itkonen, 2015, p. 162). These points are usually the result of the effort estimation. Several examples of visualizations are shown in appendix 9.8. These metrics serve as a basis for the creation of a planning tool in chapter 6.

3 Problem identification

In this chapter first of all, the current state of the company is assessed using an existing model. Second, Agile software development at Polder Valley has been reviewed. Third, a model is created regarding the current situation based on paragraph 2.1. Then, this model is analyzed.

3.1 Capability Maturity Model

To understand where PV stands as an organization it's beneficial to use an existing model as a framework. The Capability Maturity Model for Software (Paulk, Curtis, Chrissis, & Weber, 1993) was developed to make a distinction between immature and mature software development organizations. There are five levels which in indicate the maturity of a company ranging from initial to optimizing. The first level is the initial level. This level doesn't have any characteristics and is therefore not present in the image. For this survey I only considered the development team, the Product Owner and the Scrum Master (no Business owners or developers). Because they have the most accurate view of the presence of elements as they are the ones involved with them. After conducting a small survey among the development team, which can be found in appendix 9.9 as well as the initial results, Figure 11 can be created.

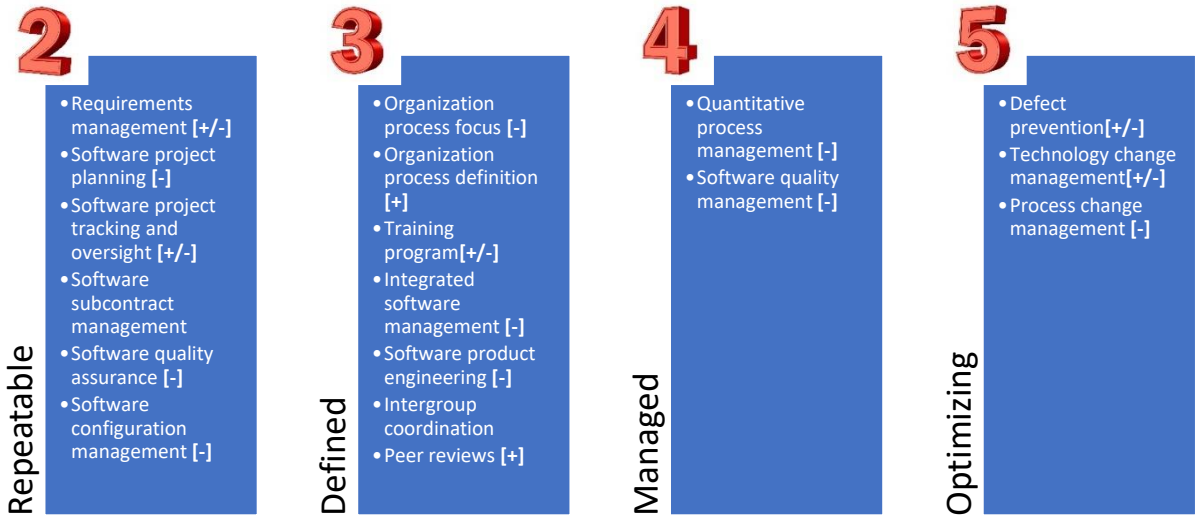


Figure 11 CMM survey results summary

A [+] means an aspect is present in the organization as an [-] means the aspect is absent. When looking at the characteristics of the levels, PV is becoming a level 2 organization: "Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications". A level 3 organization has the following characteristics: "The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for developing and maintaining software (Paulk, Curtis, Chrissis, & Weber, 1993, p. 9)". It's important that findings during the research help PV obtain all level 2 characteristics instead of focusing on level 3, 4 or 5 characteristics. It can be seen that Polder Valley is currently assessed as a company striving to reach the second level. This doesn't mean that it is bad that there is some focus on higher level characteristics. This also supports the research objective to improve software project planning as well as software project tracking and oversight. As these are both level 2 characteristics.

3.2 Agile software development at Polder Valley

In this paragraph various aspects of Agile software development that are present at Polder Valley are explained. Subsequently, Scrum, Kanban, VSTS and planning cycles are explained. Then, the roles at Polder Valley are explained and the desired Agile methodology is assessed. This paragraph is of importance to understand the current situation and the As-Is model.

3.2.1 Scrum

In Scrum, on each day of a sprint, the team holds a daily Scrum meeting called the 'daily Scrum'. Meetings are typically held in the same location and at the same time each day. Ideally, a daily Scrum meeting is held in the morning, as it helps set the context for the coming day's work. These Scrum meetings are strictly time-boxed to 15 minutes. This keeps the discussion brisk but relevant (Mountain Goat Software, 2018). At PV these meetings are held about two times a week, due to the lack of present workers at the office. Other elements are as shown in paragraph 2.2.4. Which elements are present at Polder Valley becomes clear in paragraph 3.3.

3.2.2 Kanban

A Kanban board is a work and workflow visualization tool that enables you to optimize the flow of your work. Physical Kanban boards, like the one pictured below, typically use sticky notes on a whiteboard to communicate status, progress, and issues. Online Kanban boards draw upon the whiteboard metaphor in a software setting (Planview, 2018).

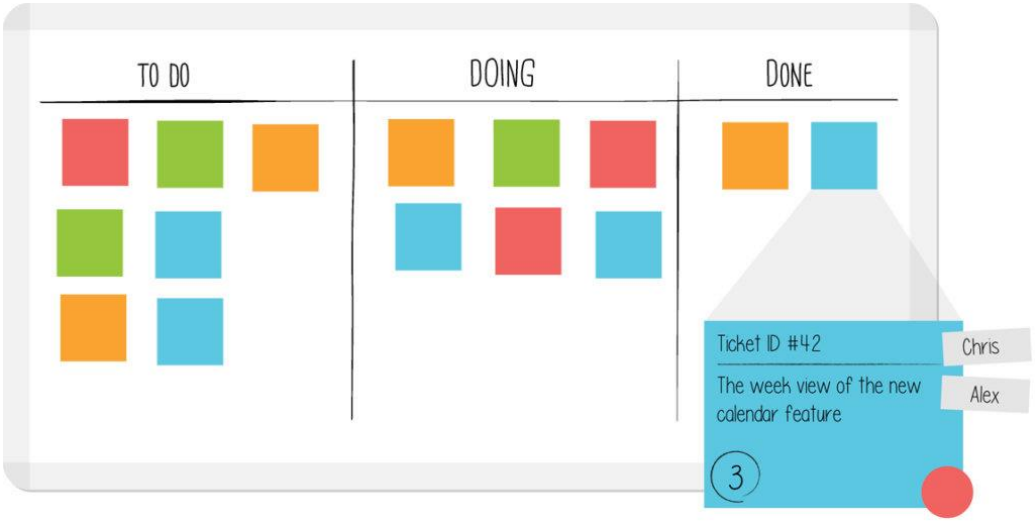


Figure 12 Kanban board example - https://leankit.com/learn/kanban/kanban-board/

3.2.3 Microsoft Visual Studio Team Services (VSTS)

VSTS is the coding, planning, overview environment used at PV. This environment contains Kanban boards containing tasks and backlog items. This is also the place where code branches can be merged, and code can be tested (Microsoft, 2018). VSTS has a very present role in the planning process and can be used a source of data for my research. An example and some more information about VSTS can be found in appendix 9.1.

3.2.4 Planning cycle (MMP)

The current planning cycle, and the one I will be present for at PV is one of 6 months that started in January. At the end of the current planning cycle the stakeholders want to have a Minimum Marketable Product (MMP). "The MMP describes the product with the smallest possible feature set that addresses the needs of the initial users (innovators and early adopters) and can hence be marketed and/or sold" (Pichler, 2013).

3.2.5 Roles

There are different roles within the team that are used throughout this thesis. It's possible somebody has multiple roles. At Polder Valley this is the case. The relation of different roles for Polder Valley has been considered as following for this thesis. All these actors are considered stakeholders of the product being developed.

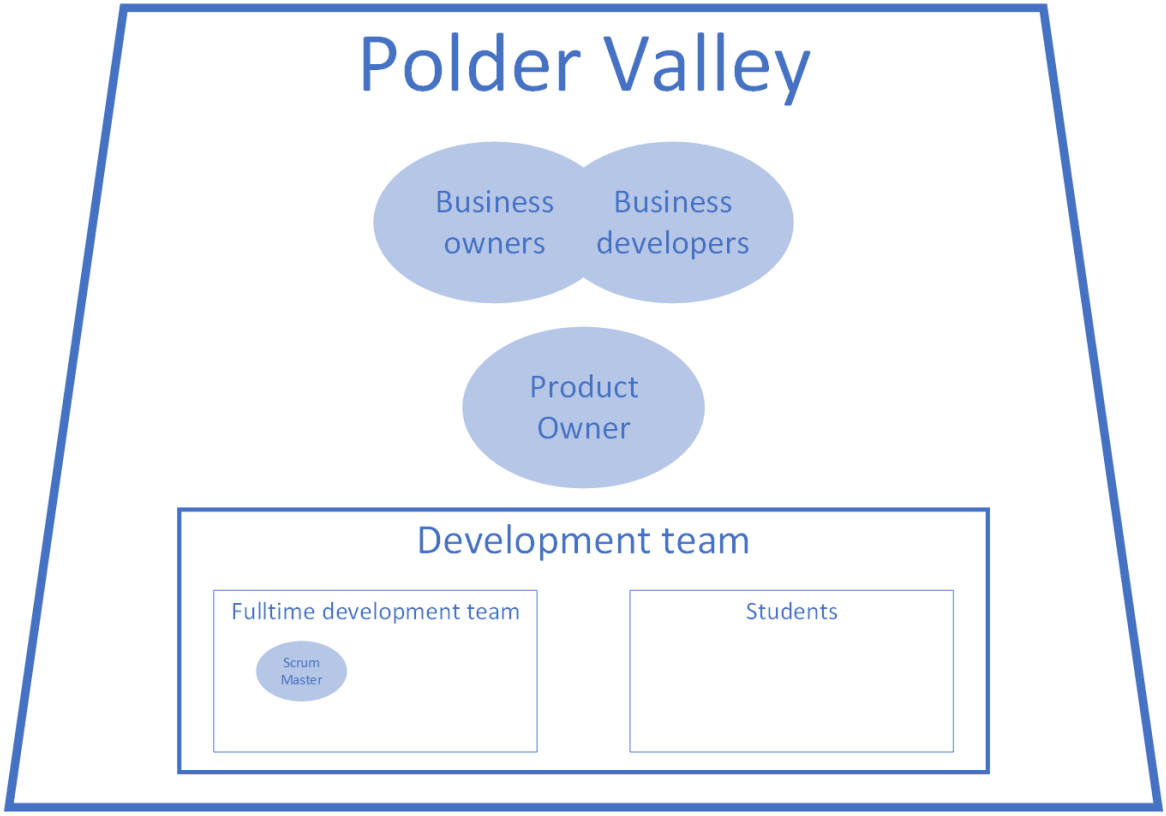


Figure 13 Polder Valley team structure overview

Business owners are the ones accountable in the end for Polder Valley.

Business developers are focused on operationalizing ideas and developing Polder Valley as an organization.

"The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals" (Schwaber & Sutherland, 2018).

"The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values" (Schwaber & Sutherland, 2018).

The case of Scrum Master is interesting. This person has the role of Scrum Master but is also considered a fulltime developer. Besides all the terms explained so far, there are more Agile software development terms of importance throughout this thesis such as an Epic or a Burndown chart, which is shown in Figure 14. These terms are explained according to literature in appendix 9.10.

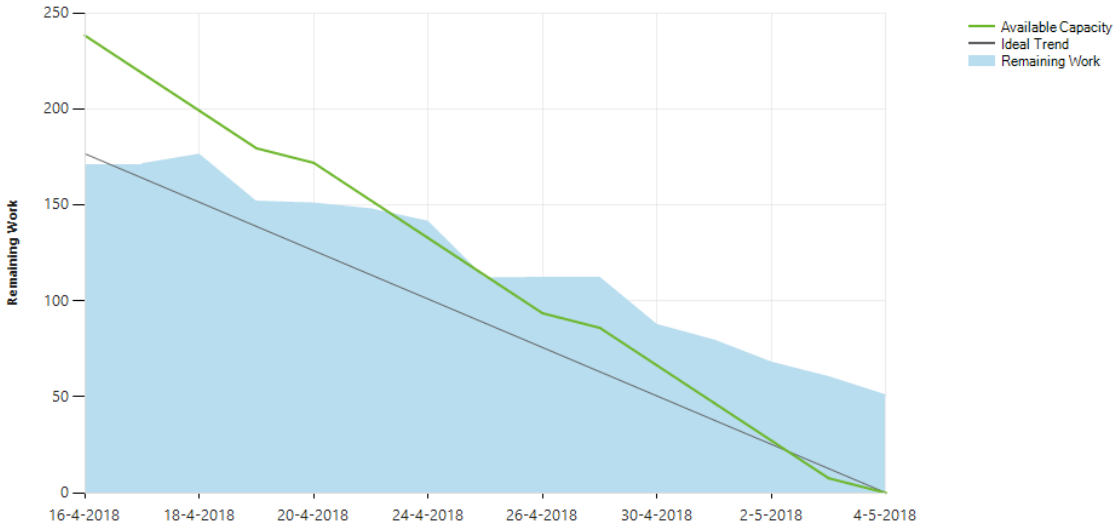


Figure 14 Burndown chart example

3.2.6 Desired Agile methodology

Scrum is a widely adopted Agile framework, as well as eXtreme Programming. We have been able to see this in paragraph 2.2.2. Kanban is also used at Polder Valley. At least through the use of a Kanban board. Therefore, Kanban is also taken into account. Further research and talking to people at Polder Valley also introduces Lean Agile Software Development and DevOps (combination of development and operations).

Polder Valley has the ambition to perform Scrum as a methodology of Agile. However, there are also other methodologies. As explained before, the most used methodologies are Scrum, Kanban, Lean eXtreme Programming and DevOps. Each methodology has their own aspects which have been mapped using the definitions from Objectstyle (Krush, 2017) and VersionOne (VersionOne, 2018). In Table 3, if an element is present in a framework this is marked with an ‘x’. In the last column there is an ’x’ if this element is present within Polder Valley. At the bottom row, there’s shown how much of a framework has been adopted by Polder Valley.

Table 3 Agile elements methodology comparison

Element	Scrum	Kanban	Lean	eXtreme Programming	DevOps	Polder Valley
User story	x					x
Task	x					x
Backlog	x					x
Sprint backlog	x					x
Product increment	x	x		x		
Extensions/reports	x					x
Planning/replenishment meeting or Iteration plan	x	x		x		x
Daily stand-up	x			x		
Review	x					x
Retrospective	x					x
Scrum Master	x					x
Product Owner	x					x
Pull system		x				x
Kanban Board/Visualization		x				x
Ideation		x				
Acceptance (definition of done or pre-defined test)		x		x		x
Flow management/minimizing WIP		x				
Process mapping			x			x
Set-Based Design			x			
Minimum Viable Product			x			x
Continuous development			x	x	x	
As fast as possible (direct value added)			x		x	
Release plan (months)				x		x
Pair negotiation				x		
Unit test				x		x
Pair programming				x		
Planning game				x		x
Coding standards				x		
Automation					x	x
Presence of framework in PV	**83%**	**57%**	**40%**	**45%**	**33%**	**-**

It can be seen Polder Valley mostly adopts the Scrum methodology, as attended. But there is also a significant presence of elements from other methodologies. The best choice is to fully implement a methodology rather than to only pick particular elements. When a methodology is fully implemented it can be reviewed to perhaps stop or remove certain activities or elements. When elements of other methodologies are present this is not necessarily a problem. Especially when they don't cost a significant amount of time to maintain or clearly add value to overall performance.

3.3 As-Is model

In this paragraph, first of all, there is a summary of the collected data. Then, a BPMN model is presented.

3.3.1 Data collection

In this paragraph, various planning moments have been observed. Subsequently, the planning meeting, retrospective meeting, refinement meeting and the sprint update meeting. Then, an interview was held with the product owner.

Besides this data needed to be collected on how much time the team spends on the activities and how much they value the elements these activities add value to. Therefore, the team filled in a survey. The contents of the survey can be found in appendix 9.11. Furthermore, most of the roles became clear from the observations and interview. However, for filling in gaps and validation the roles were presented to the product owner often. The results of the survey are shown in Figure 15.

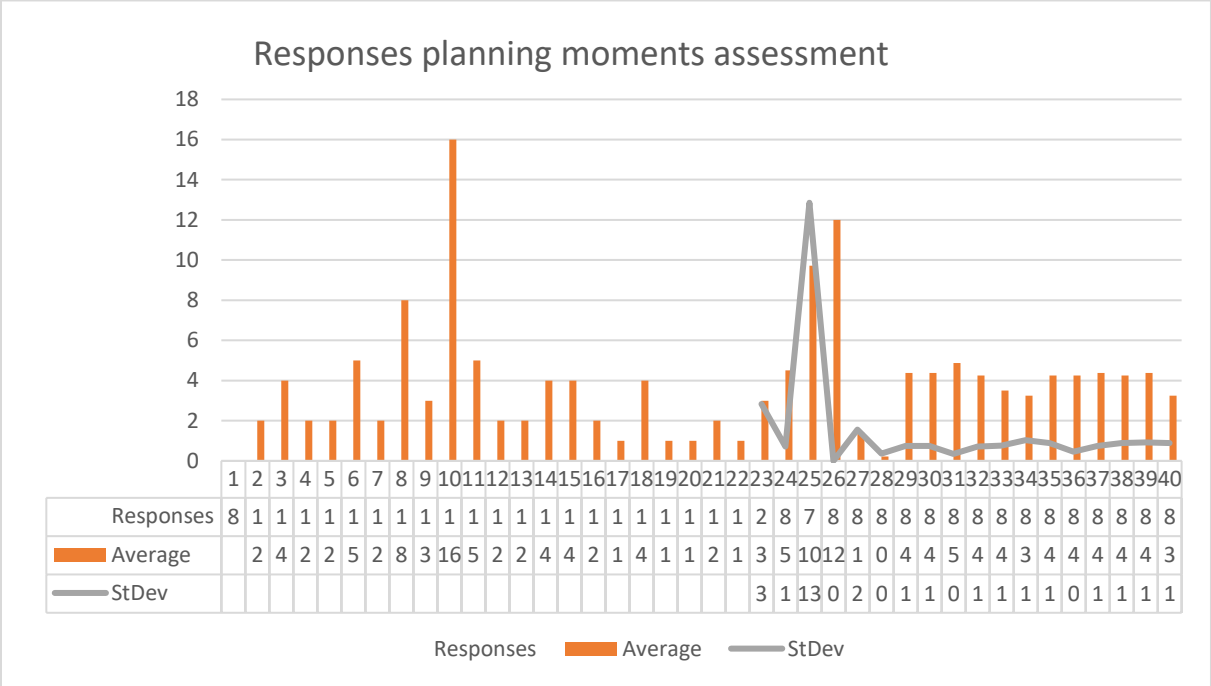


Figure 15 Results planning moments assessment

The answers shown are in the same order as the questions were asked. Answers have been operationalized where Strongly disagree = 1, Strongly agree = 5. I don't know wasn't chosen. For the peak in the standard deviation noisy data was not considered for further analysis. The cause of the noise seemed to be a misinterpretation of the question. Then, the data is used in the BPMN model in paragraph 3.3.2.

For the collection of data time spent by Business owners and developers hasn't been considered. They aren't involved in the planning process on a daily basis but merely on a macro level. As problems and difficulties are related to the team, they are only observed and questioned them for the analysis of the context. The outcome of the survey will be combined with the time costs for the purpose of analysis in paragraph 3.4. The observations were done during the planning meeting, retrospective meeting, refinement meeting and sprint update meeting. The purpose was to only observe and log activities for the creation of the As-Is model. An interview with the Product Owner was done to fill the gaps and get information on unobservable activities. The findings of the observations and the interview can be found in appendix 9.12.

3.3.2 BPMN model

Figure 16 and Figure 17 contain an overview of the entire model containing 5 connected pools. Close-up views of the individual pools can be seen in appendix 9.13. Strategic and release planning happens per planning cycle. The current planning cycle focuses on releasing an MMP. The next cycle will focus on releasing a Beta version. A release planning contains various sprints (3 weeks at Polder Valley). Each sprint has an individual planning. The current planning contains 8 sprints, but this can differ. All activities in this pool occur every 3 weeks. Within a sprint there are activities which occur on a weekly basis. These activities are shown in the pool 'Weekly planning'. Activities un 'Daily planning' occur on a daily basis. The model is viewed more closely through analysis in paragraph 3.4.

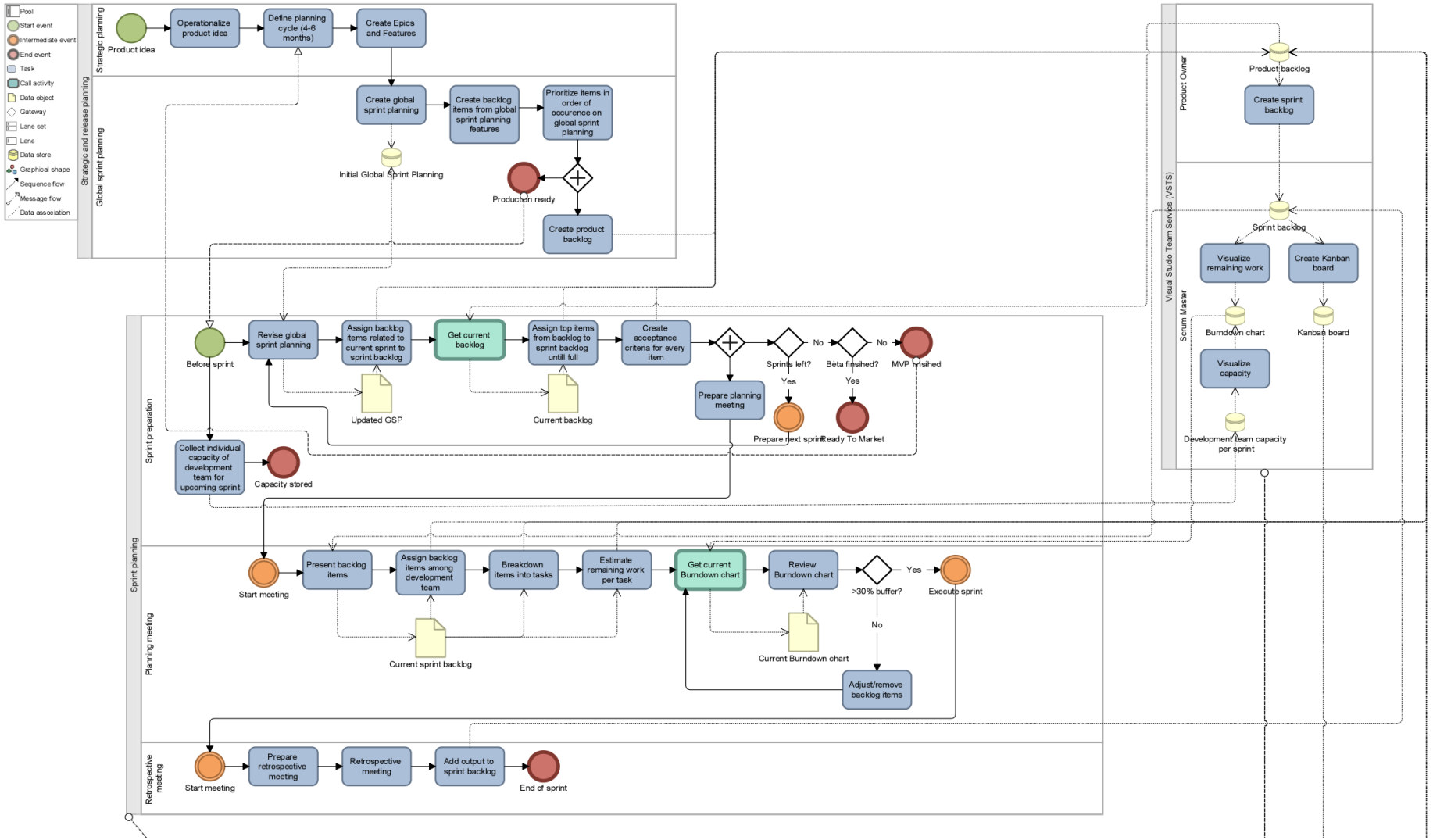


Figure 16 Planning process overview model 1/2

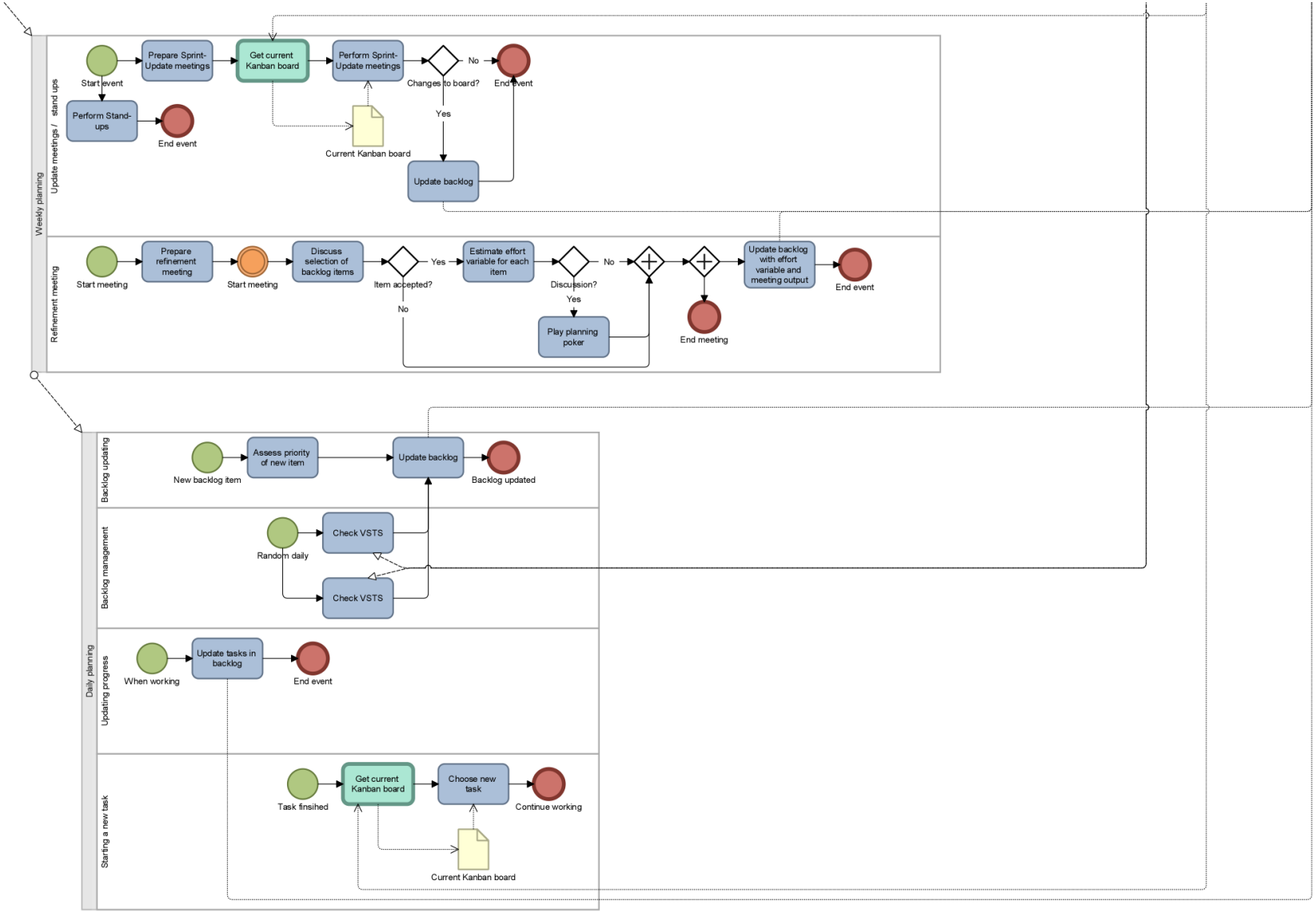


Figure 17 Planning process overview model 2/2

3.4 Analysis of the current situation

In this chapter the findings of paragraph 3.3 are analyzed. First, an overview is presented. Then subsequently, there is a focus on steps, processing time and value added, RACI roles and, planning and scheduling variables.

3.4.1 Overview

Taking into consideration the BPMN model, one can see it's quite difficult. After creating different versions for validation by my supervisor and the Product Owner the model from paragraph 3.3.2 could be created. But, because the model is quite complicated an overview model has been created.

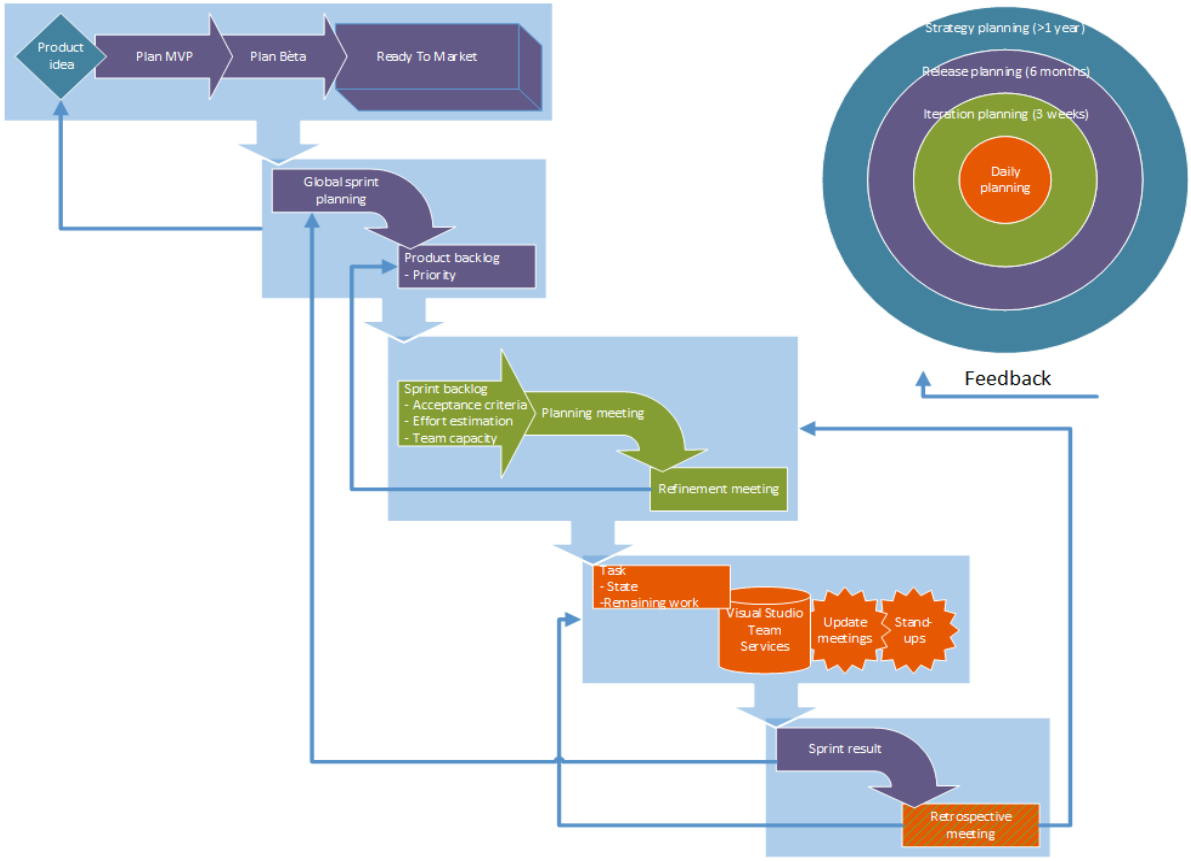


Figure 18 Current planning process overview model

The process starts with the idea for a product and a plan for an MMP. In order to achieve this, a product backlog is created which is refined during iterations. Each iteration has its own backlog which is called the sprint backlog. This sprint backlog is assessed during the planning meeting where the team commits to the work and breaks it down into tasks. Tasks are done on a daily basis with the support of update meetings and stand-ups. Visual Studio Team Services is used to visualize the entire process. At the end of a sprint all the planned work has either been done or not. In the latter case, this can influence the global sprint planning. Lastly, a retrospective session is done in order to improve the entire process and tailor it to the development team's needs.

3.4.2 Processing time and value added

As each process contains a mean processing time it's possible to calculate the number of hours spent on these tasks for each individual function. Also, the available hours can be taken into account to obtain an overview of the capacity. For the fulltime employees, the available hours are known. For the student's the bookable hours from 1-1-2018 to 31-3-2018 have been analyzed. This yields the results shown in Table 4. The hours are shown per sprint (3 weeks).

Table 4 Hours per sprint

Function	Hours available	Hours spent on planning	Hours left for development	Percentage overhead
Product Owner	48	55	-	115%
Scrum Master	108	24	84	22%
Fulltime developer	96	15	81	16%
Students	145,6154	63,75	81,86538	44%
Total	**397,6154**	**157,75**	**246,8654**	**40%**

Two main problems are clear. First of all, the Product Owner has more hours of work then he has available. Second, the students spend almost half of their time on planning related activities. This is problematic. Also taken into account that the following activities haven't been taken into account:

- The Product Owner spends 4 hours a week assisting the team in another way
- The Scrum Master spends 3 hours a week assisting the team in another way
- The development team spends 1 hour per sprint on estimating their own capacity

For each activity an assessment was done to state to which element or activity (from the questionnaire as explained in paragraph 3.3.1) it adds value. This enables us to calculate the amount of work done to be able to provide for one of the elements or activities. For example, preparing the planning meeting is necessary to have the planning meeting. Or estimating the remaining work per task is necessary for the visualization of the remaining work per task. Time costs for strategic and release planning we're not measurable and are therefore not taken into account. Analysis has been done regarding the value-added activities. However, there were no clear improvement possible as a result of these activities. The analysis can be found in appendix 9.14.

3.4.3 RACI roles

As explained before, the model contains all the different roles present during an activity. Each activity has the roles defined. The overview of all activities and roles can be found in appendix 9.14. In Figure 19 a summary is shown with the count of roles per function. This is shown for Business owners (BO), Business developers (BD), Product Owner (PO), Scrum Master (SM), Fulltime developers (FDT), the Development team (DT).

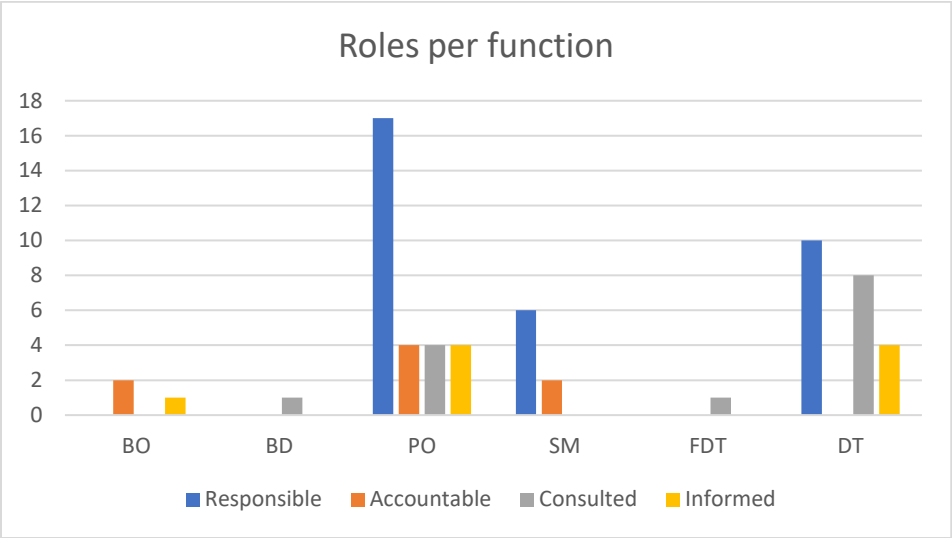


Figure 19 Number of different roles per function

A few things stick out in this overview. First of all, the Product Owner appears to have a lot of responsibility compared to any other group within the team. Second, within different roles people aren't often informed during activities. Last, there is no responsibility for Fulltime Developers in relation to the development team in general.

3.4.4 Planning and scheduling variables

From the model we see four metrics being used. There is an estimation of the capacity, remaining work per task, effort of a backlog item and the priority. The first two are actively used to assess the amount of work to be done in a sprint.

Looking at the model and questioning certain things I've uncovered several variables that could be influencing the current situation:

- The correctness of the metrics being used (compared to actual hours of work)
- The number of acceptance criteria an item has
- When a lot of work is done
- The amount of new backlog items added during a sprint
- The number of bugs during a sprint

It could also be useful to analyze tasks to assess correctness of effort estimation, see what kind of tasks are finished, look at the number of tasks per person and measure the amount of test that are done.

4 Solution objectives

In this chapter paragraphs 2.2.3 and 2.2.4 are combined with paragraph 2.1 to create an ideal model. Then, using chapter 3 and the preceding contents of this chapter a gap analysis is performed.

4.1 Ideal model

In this paragraph an ideal model is created. First, an overview model is presented. Then, the view of the management of Polder Valley is taken into account. Finally, the BPMN model are shown.

4.1.1 Overview model

Using literature, as explained in the introduction of this chapter, an ideal model could be created. This started with the creation of an overview model. This model is considered to be an unreachable state. And furthermore, it considers only fulltime employees. Through combining knowledge about the planning levels and the application of Scrum the overview in Figure 20 was created.

The different colors relate to the planning levels as explained in paragraph 2.2.3. The levels are inspired by an ideal Scrum process as described in paragraph 2.2.4. Furthermore, the different categories of metrics are explained which can provide useful insights. These are the categories as identified in paragraph 2.2.5 (Kupiainen, Mäntylä, & Itkonen, 2015, p. 152). This model is ending. However, we can still consider that there are various releases (containing sprints, etc.). After the ending of the sprint the next one begins, until a release is finished (when the next release begins).

This model does contain a few elements that shouldn't be necessary in the ideal situation. These are the blocks with the grey color. However, assuming the ideal situation can never be reached, we assume that the product as well as the process should be improved. Therefore, we take into account the result of a sprint which yields feedback. Through this feedback the product could perhaps be improved on any level (from the goal to a single feature). The review meeting, which occurs at the end of sprint, are there to provide the development team with a complete view of the product. As they are also a stakeholder regarding the product this could also lead to improvements on any level regarding the product being made. Last, a retrospective meeting is held in order to improve the process. Again, on any level.

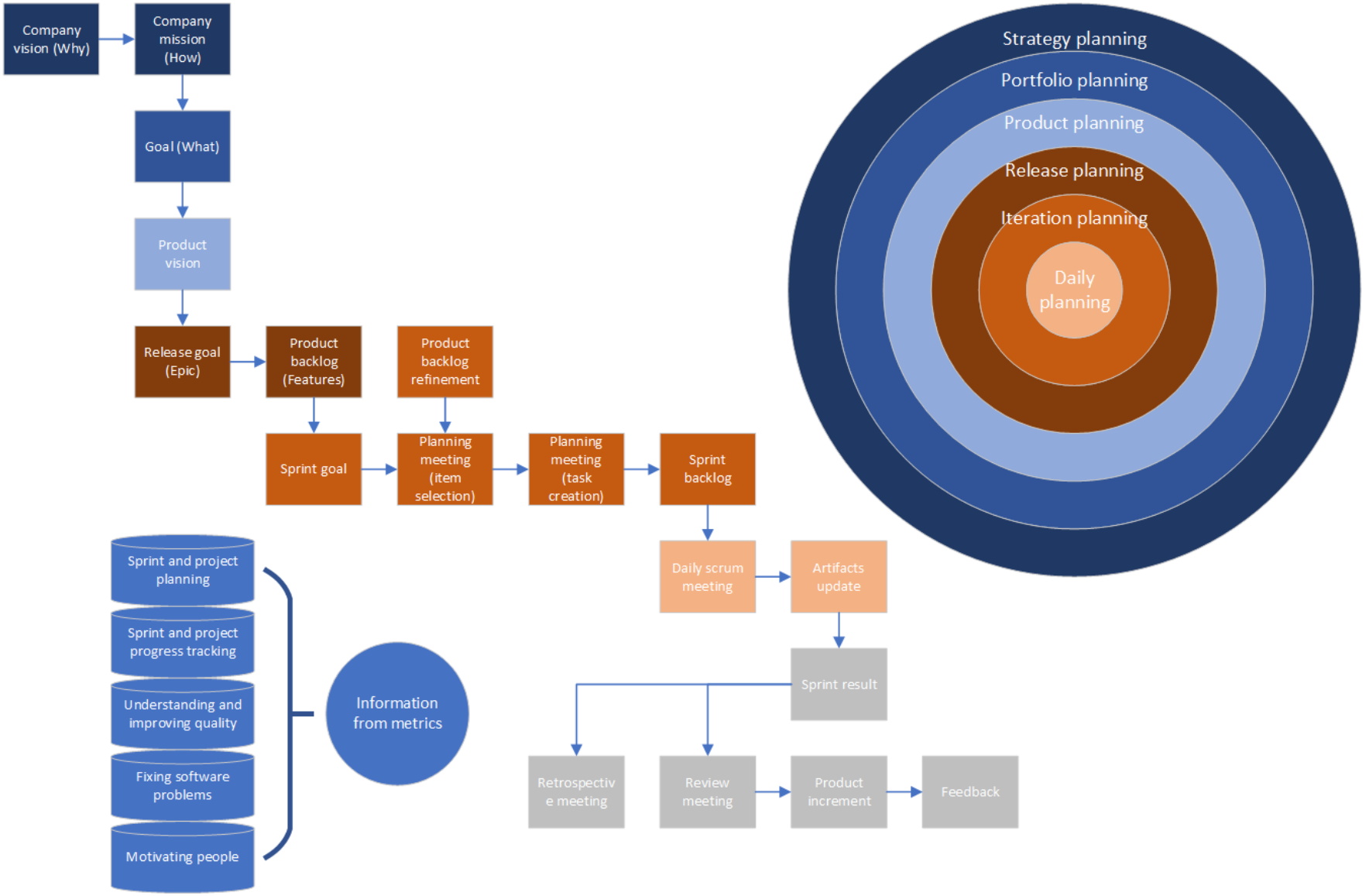


Figure 20 Ideal planning process overview model

4.1.2 Management Polder Valley (RACI roles)

The management of Polder Valley has some requirements to the way of working considered in this research. As mentioned before, time is to be considered rather than costs. Scrum is chosen as a methodology and the ideal model focuses on fulltime employees (even though now there are mostly students working).

As the current situation has all the roles required to be operational in the ideal situation, management can be consulted regarding the division of roles they would ideally desire. Looking at the overview model. Management desires the following division of roles according to the RACI method as used in paragraph 3.4.3. After consulting the current Product Owner the division of roles is created as in Figure 21. The entire division of roles is shown in appendix 9.16.

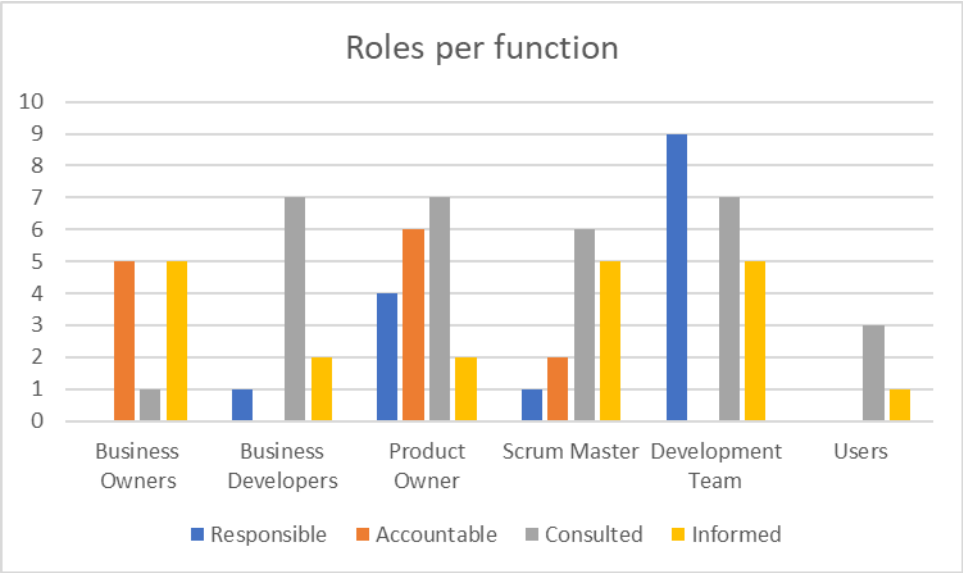


Figure 21 Roles per function ideal model

Users have been involved, as in the ideal situation continuous development will be part of the process. Then, users become a part of the process. It stands out that Business owners are either accountable or informed regarding a part of the process. They are merely informed in order to make the correct decisions. Furthermore, most of the aspects are the Development teams' responsibility, but the development team is never accountable. And, different functions are often consulted, which means they are actively involved in processes, but not responsible.

4.1.3 BPMN model regarding VSM

Knowing processing times from the current situation a BPMN model has been created showing the ideal situation. The meeting times are fixed, and processing times from the As-Is model are assumed to be representable for the Ideal model. An overview of the model can be seen in Figure 22 and Figure 23.

Whereas the roles are as explained in paragraph 4.1.2. The time spent by various functions would be as shown in Table 5. The hours are per sprint (3 weeks). For the development team these calculations are based on three fulltime developers. Looking at the number of hours this is comparable to the current situation. Similar to paragraph 3.3, only processes that cost time per iteration are taken into account. The focus regarding this aspect is on the members of the development team. So just as in the As-Is model time costs for management aren't considered.

Table 5 Hours per function in ideal situation

Function	Hours available	Hours spent on planning	Hours left for development	Percentage overhead
Product Owner	120	16	-	13%
Scrum Master	120	10,5	-	9%
Development team	360	46,5	313,5	13%
Total	600	73	423	12%

As can be seen there are no problems in the ideal situation. Every function has a lot of hours left to spend on tasks at hand. For the Product Owner this focus could be on checking work done or collecting input from users. Furthermore, the Scrum Master is ideally separated from the development team. As this allows the Scrum Master to remain impartial when resolving possible conflicts.

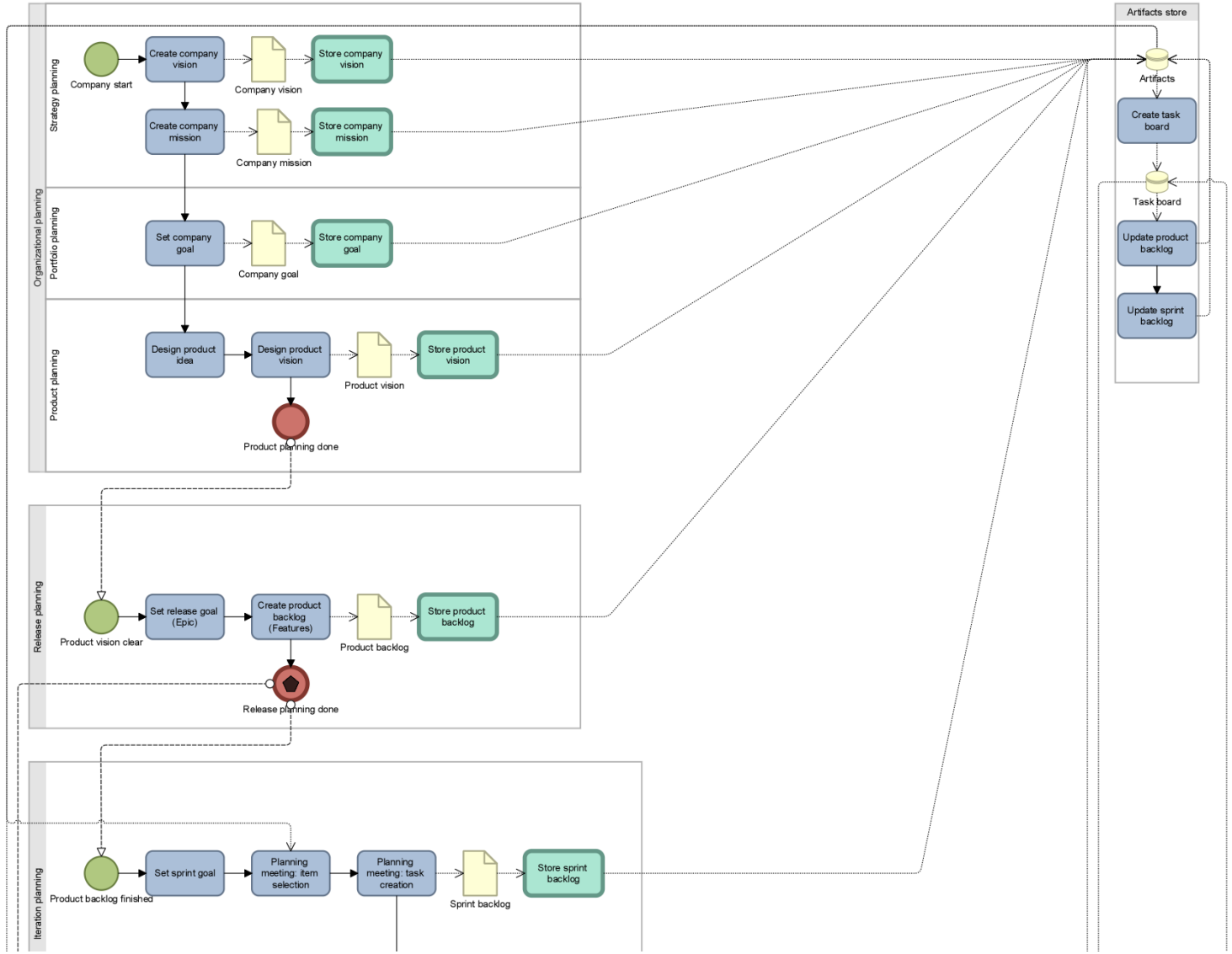


Figure 22 Ideal model overview 1/2

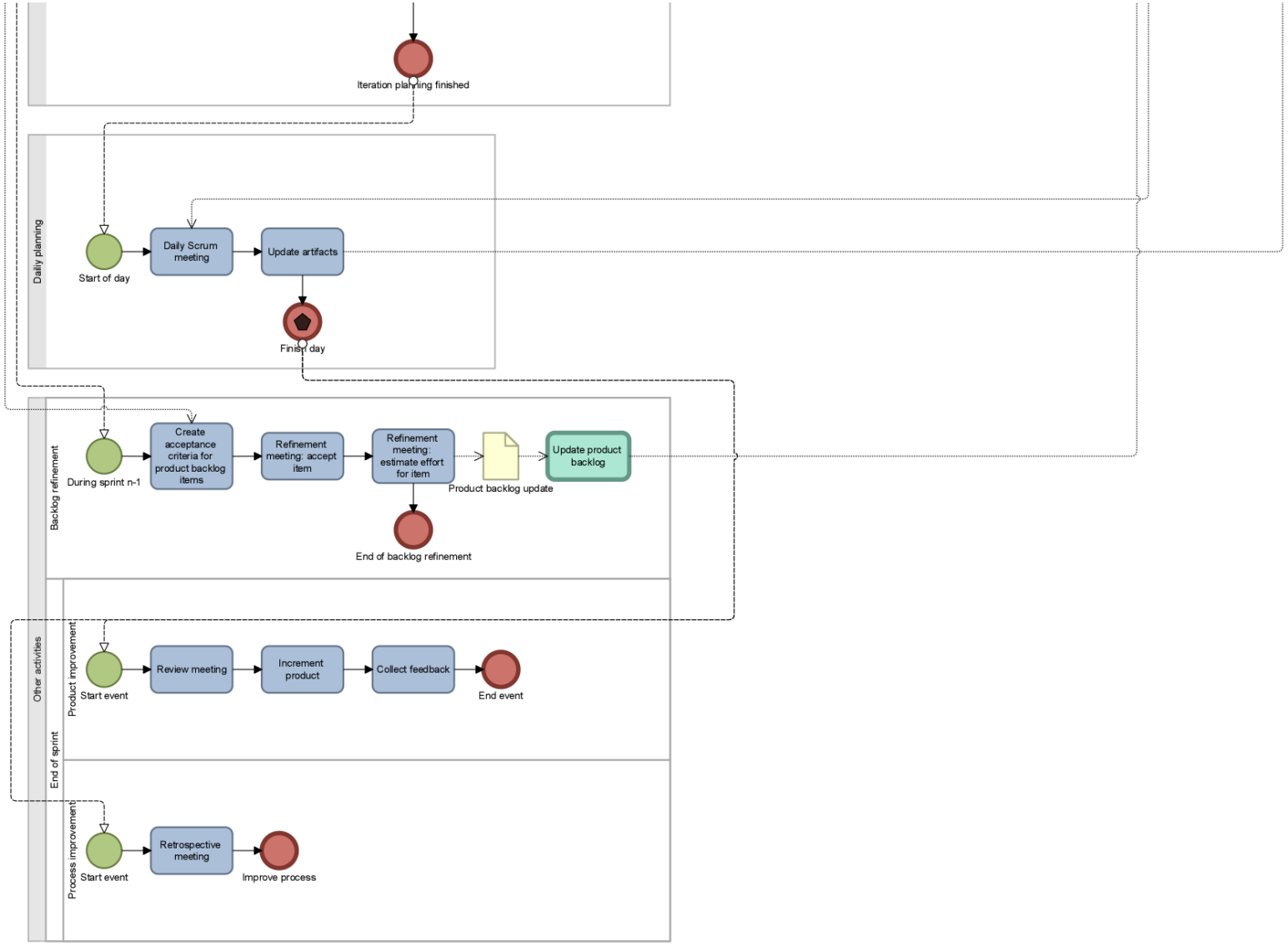


Figure 23 Ideal model overview 2/2

4.2 Gap analysis

In this paragraph a gap analysis between paragraphs 3.3 and 4.1 is performed. First of all, the steps that are taken are compared where different types are identified. Second, the processing time of all the activities is compared. These parts adhere to the VSM theory. Third, the RACI roles are analyzed. Fourth, planning and scheduling variables are assessed. Finally, problems and requirements are identified.

4.2.1 Steps taken

When comparing the steps taken (or the activities) of the models in paragraphs 3.3.2 (As-Is) and 4.1.3 (Ideal). We can clearly see that in the As-Is model there are more defined activities. Some activities are split up or phrased differently. These activities are shown in Table 6.

Table 6 Extra steps taken (As-Is) compared to the Ideal situation

Level	Activity	Done by	Type
Release planning	**Prioritize items in order of occurrence on global sprint planning**	PO	Sprint design
Release planning	**Store items in Visual Studio Team Services**	PO	Updating artifacts
Sprint preparation	**Revise global sprint planning**	PO	Sprint design
Sprint preparation	**Assign backlog items related to current sprint**	PO	Sprint design
Sprint preparation	**Get current backlog**	-	Automated process
Sprint preparation	**Assign top items from backlog to current sprint until full**	PO	Sprint design
Sprint preparation	**Collect individual capacity of DT for upcoming sprint**	SM	Capacity collection
Sprint preparation	**Prepare planning meeting**	PO	Meeting preparation
Retrospective meeting	**Prepare retrospective meeting**	SM/PO	Meeting preparation
Meetings	**Prepare Sprint-Update meetings**	SM	Meeting preparation
Meetings	**Get current Kanban board**	-	Automated process
Weekly planning	**Prepare refinement meeting**	PO	Meeting preparation
Weekly planning	**Update backlog with effort variable and meeting output**	PO	Updating artifacts
Backlog management	**Assess priority of new item**	PO	Updating artifacts
Backlog management	**Update backlog**	PO	Updating artifacts
Backlog management	**Check VSTS**	SM	Checking work
Backlog management	**Check VSTS**	PO	Checking work
Daily planning	**Update tasks in backlog**	DT	Updating artifacts
Daily planning	**Get current Kanban board**	-	Automated process
Daily planning	**Choose new task**	DT	Updating artifacts

Several types of activities have been identified. The first type is ‘Sprint design’. All these activities are done by the Product Owner and can be seen as a part of the PO’s work. This is also the case with “Updating artifacts”. This is done by the PO and the Development Team (DT). This is a part of the way of working which isn’t inherent to software development. Then, the ‘Capacity collection’ costs time as well. This is because it is necessary to do this for the students. Making this activity obsolete would be an improvement. Also, meetings are actively prepared. Perhaps, if a system were in place to make these activities unnecessary it would be an improvement too. Last, there are some automated processes present. These processes don’t cost any time and aid the team. Therefore, the presence of these activities is fine.

There are also various activities identified in the Ideal model that aren't present in the As-Is model. These activities are shown in Table 7.

Table 7 Missing steps taken (Ideal) compared to the current situation

Level	**Activity**	**Done by**
Strategy planning	**Create company vision**	BO, BD, PO
Strategy planning	**Create company mission**	BO, BD, PO
Portfolio planning	**Set company goal**	BO, BD, PO
Product planning	**Design product vision**	BO, BD, PO DT
Iteration planning	**Set sprint goal**	PO, DT
Product improvement	**Review meeting**	PO, DT, US, BO, BD
Product improvement	**Increment product**	PO
Product improvement	**Collect feedback**	PO, US

Here a distinction can be made between three different types of activities. First of all, the top four activities are the first four activities that should be done in planning. These activities are all of importance for the highest level of planning. It is important that it's clear throughout the organization what different visions, missions or goals are. Since, people then know exactly what they are making. Second, no goal is defined for a separate iteration. Now, the team merely commits to an amount of backlog items. However, if a clear goal were present this can lead to more clarity towards the items being worked on, and the relationships between them. Last, at this moment, there is no continuous development being done. This means that there is no product that is being improved. The team is still working on the first MMP. Still, there are internal stakeholders. Therefore, review meetings should be done, product incrementation should be clearly done and feedback should be collected. This is an important aspect of Agile software development as this is where the involvement of customers leads to a better product.

4.2.2 Processing time

As we could see in Table 4 and Table 5 there is much more time spent on planning in the As-Is model than in the Ideal model. Where in the current situation 40% of all available hours go towards planning every sprint this is only 12% in the Ideal model. This could be caused by the extra steps shown in Table 8. All the time costs in this paragraph are per sprint (3 weeks).

Table 8 Time costs of extra steps taken Table 6

Activity	Time costs (mins)
Prioritize items in order of occurrence on global sprint planning	0
Store items in Visual Studio Team Services	0
Revise global sprint planning	120
Assign backlog items related to current sprint	120
Get current backlog	0
Assign top items from backlog to current sprint until full	120
Collect individual capacity of DT for upcoming sprint	120
Prepare planning meeting	210
Prepare retrospective meeting	60
Prepare Sprint-Update meetings	180
Get current Kanban board	0
Prepare refinement meeting	120
Update backlog with effort variable and meeting output	360
Assess priority of new item	180
Update backlog	180
Check VSTS	900
Update tasks in backlog	1 575
Get current Kanban board	0
Choose new task	0
Total	**4 245**

4 245 minutes (70,75 hours) per sprint are spent extra. This is 70,75 hours which is about 18% of the total time available. When we combine Table 8 with Table 6 with respect to the types of activities we see create Table 9.

Table 9 Time costs of extra steps taken per activity type

Activity type	Performed by	Time costs (mins)
Sprint design	PO	360
Updating artifacts	PO	720
Updating artifacts	DT	1575
Capacity collection	SM	120
Meeting preparation	PO/SM	570
Checking work in VSTS	SM	180
Checking work in VSTS	PO	720

This provides useful insights with respect to paragraph 4.2.1 and the problems identified there. This shows us that only the 690 minutes spent on capacity collection and meeting preparation are unnecessary overhead time. All other activities are functional tasks.

Then, we compare the overlapping activities of the Ideal and As-Is model with respect to time costs in Table 10.

Table 10 Time costs overlapping activities As-Is and Ideal model

Level ideal model	Activity Ideal model	Time costs	Activity As-Is model	Time costs
Product planning	Design product idea	0	Operationalize product idea	0
Release planning	Set release goal (Epic)	0	Define planning cycle (4-6 months)	0
			Create Epics and Features	0
	Create product backlog (Features)	0	Create global sprint planning	0
			Create backlog items from global sprint planning features	0
Iteration planning	Planning meeting: item selection	225	Present backlog items	105
			Assign backlog items among DT	105
			Breakdown items into tasks	105
			Estimate remaining work per task	105
	Planning meeting: task creation	225	Get current Burndown chart	0
			Review Burndown chart	105
			Adjust/remove backlog items	105
Daily planning	Daily Scrum meeting	900	Perform Sprint-Update meetings	630
			Update backlog	1575
	Update artifacts	900	Perform Stand-ups	270
Backlog refinement	Create acceptance criteria for product backlog items	480	Create acceptance criteria for every item	480
	Refinement meeting: accept item	540	Discuss selection of items for upcoming iteration	630
	Refinement meeting: estimate effort for item	540	Assess effort variable for each item	630
			Play planning poker	630
Process improvement	Retrospective meeting	450	Retrospective meeting	630
Total		**4260**		**5790**

In the table it is clearly shown that activities that overlap between the As-Is and Ideal model costs more time in the current situation. This is mostly caused by the presence of students which negatively up the overhead planning time costs due to the presence of more people in the team. Also, due to the combined Scrum Master/Developer role more time costs are inclined. Also, there is more time spent regarding the Daily Scrum meeting. Even though the meeting is done only twice a week to reduce overhead. Here seems to be room for improvement. Daily Scrum meetings are never done on Friday (As-Is) due to the absence of fulltime developers.

4.2.3 RACI Roles

Looking at the difference of the defined roles regarding responsibility we start by looking at the overlapping activities in Table 11. In this table, yellow labeled roles in the Ideal model are missing in the As-Is model. Yellow labeled roles in the As-Is model are not present in the related level of the Ideal model. Red labeled roles in the As-Is model have a different responsibility compared to the Ideal model. Whereas, green items are a match compared to the Ideal model.

Reviewing the table a few main things. First of all, the Scrum Master is sometimes missing in the current situation. This is most likely due to the Scrum Master role being combined with the role of a developer (in the DT). It appears that this is sometimes done by the Product Owner (in Iteration planning and Process improvement). Also, Business Developers aren't involved actively in processes related to Product, and Release, planning. Third, users and stakeholders aren't involved where they should be (Backlog refinement). When we consider stakeholders this ranges from everyone involved in the process. For example, the development team is a stakeholder as well as the Business Owners. However, this is due to the current state of the organization and is likely to occur in the future. Last, for every 'wrongly assigned' responsibility, the cause could be different. This, together with the extra and missing steps as defined in paragraph 4.2.1, is taken into account in chapter 5.

Table 11 RACI roles in overlapping steps As-Is and Ideal model

Level ideal model	Activity Ideal model	R	A	C	I	Activity As-Is model	R	A	C	I
Product planning	Design product idea	BD	BO	PO	SM, DT	Operationalize product idea	PO	BO	BD	DT
Release planning	Set release goal (Epic)	PO	BO	BD, DT		Define planning cycle (4-6 months)	PO	BO	DT	
						Create Epics and Features	PO			
	Create product backlog (Features)		PO	BD, DT	BO	Create global sprint planning	PO		DT	BO
						Create backlog items from global sprint planning features	PO			
Iteration planning	Planning meeting: item selection	PO, DT			SM	Present backlog items	PO		DT	
						Assign backlog items among DT	DT			PO
						Breakdown items into tasks	DT		PO	
						Estimate remaining work per task	DT			PO
						Get current Burndown chart				
	Planning meeting: task creation		DT	SM		Review Burndown chart	DT		PO	
						Adjust/remove backlog items	DT		PO	
Daily planning	Daily Scrum meeting	DT		SM		Perform Sprint-Update meetings	SM		DT	
						Update backlog	SM		DT	
	Update artifacts	DT		SM		Perform Stand-ups			FDT	
Backlog refinement	Create acceptance criteria for product backlog items		PO	DT, US, BO, BD		Create acceptance criteria for every item		PO		DT
	Refinement meeting: accept item		PO	DT, US, BD	BO	Discuss selection of items for upcoming iteration	PO		DT	
	Refinement meeting: estimate effort for item	DT	PO	SM		Assess effort variable for each item	DT			PO
						Play planning poker	DT	SM		PO
Process improvement	Retrospective meeting		SM	PO, DT		Retrospective meeting	DT		PO	

4.2.4 Planning and scheduling variables

As defined in paragraph 3.4.4 there is an estimation of the capacity, remaining work per task and effort of a backlog item in the current situation. Related to the types of metrics defined in paragraph 2.2.5 these are all metrics supporting 'Sprint and project planning'. There are no metrics of other types present. In an overview is shown of each metric type with the two most important metrics according to Kupiainen et al. (Kupiainen, Mäntylä, & Itkonen, 2015).

Table 12 Most important metrics per category according to Kupiainen et al.

Metric type	Popular metrics
Sprint and project planning	Velocity Effort estimate
Sprint and project progress tracking	Technical debt Progress as working code
Understanding an improving quality	Customer satisfaction Build status
Fixing software process problems	Lead time Story flow percentage
Motivating people	Technical debt Defect trend indicator

This combined with visualizations presented in appendix 9.8 can serve as a basis for providing more knowledge through metrics.

4.2.5 Problems and requirements

Through the preceding contents of this chapter several problems have been identified which require a certain solution. First of all, the missing, and extra, steps are discussed. With the capacity estimation and meeting preparation in special. Second, the time costs of daily stand-ups are discussed. Third, roles are discussed. Last, planning and scheduling variables are discussed. All discussion, hypotheticals and questions raised form a basis for solutions designed in chapter 5.

Missing/extra steps

As identified in paragraph 4.2.1 the current situation has missing, as well as extra, steps with respect to the ideal situation. Two of these (groups of) steps could be redesigned. Respectively, the steps related to capacity estimation and meeting preparation. Other steps appeared to be tasks related to a function in general rather than of incredible importance for planning. However, when creating a To-Be model with the changed steps these steps could still be present.

The requirements to solving this problem is to create a feasible To-Be model which includes the minimum number of extra steps and the maximum number of missing steps.

Capacity estimation

At this moment the capacity estimation is done in a single step as shown in Figure 24. In order to replace this step capacity must be estimated in a constant way. This can be done through analyzing past capacity estimations and worked hours.

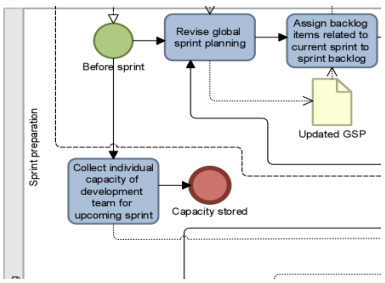


Figure 24 Capacity estimation As-Is model

Preparing for meetings

Preparing for meetings is done for the following meetings:

- Planning meetings
- Retrospective meetings
- Sprint-Update meetings
- Refinement meetings

For all these meetings the general content and structure can be reviewed to discover if the meetings are possible to hold without preparation. Therefore, the requirements for solving these problems are the creation of a fixed structure which doesn't require preparation.

No daily Scrum meetings

As identified in paragraph 4.2.2 the abnormal structure surrounding the daily Scrum meeting leads to more time costs than the regular way of working would have in an ideal situation. The current structure is shown in Figure 25 Scrum meetings in As-Is model. A different structure could decrease these costs.

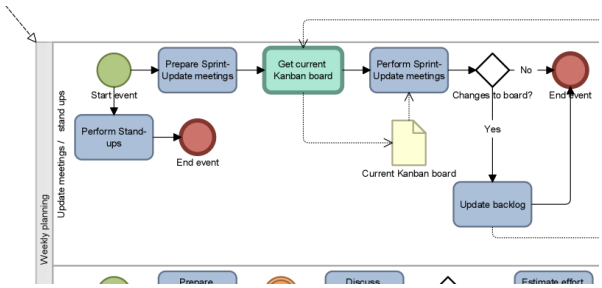


Figure 25 Scrum meetings in As-Is model

The requirements for solving the problem is the redesign of the current structure with a focus on minimizing time costs. This is done based on the current division of employees. Roles can be switched around.

Scrum Master/Developer

As explained in paragraph 4.2.3 the combination of the Scrum Master and developer role results in a different distribution of roles with the respect to the Ideal situation. A clear distinction regarding which role is taken in which situation would provide more clarity.

Incorrect roles (RACI) in activities

A shown in paragraph 4.2.3 the division of roles regarding responsibility differs when comparing the current and ideal situation. This gap can be shortened as part of creation of a To-Be model. The To-Be model is created based on paragraphs 3.3 and 4.1. Then, roles can be decided on. Within this model the amount of difference with the Ideal roles should be minimized.

Not enough use of metrics

As shown in paragraph 4.2.4 there is little use of metrics. Metrics can be purposeful in, for example, making decisions and tracking progress. When there is a way to make more information available this would be beneficial towards this. The requirement for this problem is that there is a design of an adjustable dashboard which can provide metrics through real-time data.

5 Solution design

In this chapter various solutions to problems explained in paragraph 4.2.5 are presented. First, Agile software development practices at other companies are viewed. These practices serve as an inspirational source for solutions. Second, interventions are presented. Then, a To-Be model is created using BPMN. Finally, a decision is made regarding planning and scheduling variables should be measurable.

5.1 Agile software development practices at other companies

In order to get a good overview of good Agile software development several interviews have been conducted with people working in software development. A summary of each interview can be found in appendix 9.17. In order to respect their privacy only their functions are mentioned. From all the interviews a concept matrix has been created. These interviews serve as a useful benchmark regarding Agile software development and serve as an inspiration for interventions.

The complete matrix can be found in appendix 9.18. In Table 13 the findings are shown.

Table 13 Concept matrix summary interview companies

Concept	Findings
Agile software development	Agile/Scrum can be implemented in different ways. Often teams can organize themselves. It's important to watch out for too much routine.
Team size	Ranges from 1-22. 8-12 appears to be an average size.
Number of teams	-
Product Owner	In companies with more products there is always a clear PO. Companies with one product don't necessarily have a PO. It can be done by the customer, but this doesn't seem ideal.
Scrum Master	Not always present, seems to be good when a team is not yet completely familiar with Agile.
Students	Students are used in a very 'free' way. They don't often work on core tasks.
Sprint Length	1-4 weeks.
Tools	Jira is used a lot.
Planning meetings	Usually teams have a weekly meeting, more dedicated Scrum teams do have this.
Retrospective meetings	More often done per quartile.
Review meetings (internal)	Done every 6-8 weeks.
Product demo's (external)	Sometimes other ways are used to collect feedback from customers. But there is always a structure in place to do so.
Refinement meetings	Only done by dedicated Scrum teams.
Daily meetings	Smaller teams have this less than larger/more complicated teams.
Backlog management	With project-based development the customer decides. Otherwise, the team has a fitting structure.
Roadmap/themes	A clear goal seems more important. A roadmap is always good to have.
Scrum/Kanban	Kanban are used very often. More experienced teams need less structure. Either teams are dedicated or cherry pick.
Continuous development	Somebody needs to be assigned to maintenance. Continuous development is a standard.
Acceptance criteria	Created by the PO. Sometimes a standardized format.
Achieving sprints	Quite often not all items are finished at the end of a sprint. This should be handled with care, so no work gets left behind.
Positive influence on planning	Clear stories, capacity (enough), good PO (assess BV and fast decisions), prevent bugs from occurring, working together, variable scope, plan micro not macro (too much).
Negative influence on planning	Overhead, too much freedom (individualism), bad communication, expectations, fear of commitment and downside of positive points.
Capacity estimation	Velocity is used for an indication, but hard to control.

It became most clear that Agile is used very differently. Usually teams adapt Agile to their specific needs. Mostly this provides a nice overview which reflects the way successful companies apply software development. It also portrays how experimental the field of Agile software development still is.

5.2 Interventions

In this paragraph interventions are proposed. First, interventions are proposed related to Value Stream Mapping. Second, interventions are proposed related to the division of roles regarding responsibility. Third, planning and scheduling variables that are to be made measurable are identified. Finally, the To-Be model is presented.

5.2.1 VSM

In this paragraph interventions are subsequently proposed related to capacity management, meeting preparation, daily Scrum meetings, and steps and tasks.

Capacity estimation

In the current situation, capacity is individually collected for every member of the development team. This is done taking into account days off, vacation days and, most important, the general capacity of students. To collect, the latter in particular costs time and leaves room for error. The capacity of students is heavily influenced by study pressure. So, if there is an exam week the capacity is low. If there's a holiday week the capacity is high. Designing an intervention requires some analysis and the design of several possible interventions. The complete analysis is shown in appendix 9.19. The results are shown in Figure 26 and Table 14.

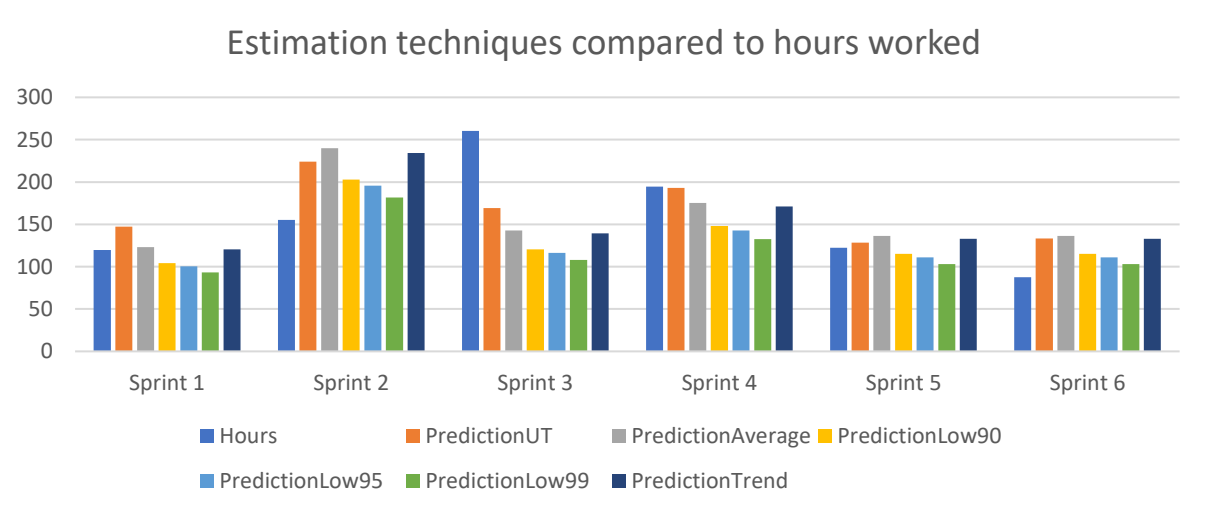


Figure 26 Estimation techniques compared to hours worked

None of the estimations seem very accurate. Therefore, we compare the error scores in Table 14.

Table 14 Estimation techniques compared to hours worked (scores)

Sprint	Hours	PredictionUT	PredictionAverage	PredictionLow90	PredictionLow95	PredictionLow99	PredictionTrend
Mean Abs error	-	26,62%	29,86%	26,41%	26,66%	27,13%	28,38%
Mean error	-	-1,67%	-4,46%	-19,24%	-22,07%	-27,61%	-6,69%

None of the estimation techniques have an absolute error below 26%. The best prediction is the prediction based on the University of Twente schedule where the mean error is only -1,67%.

From all this information it can be concluded that the capacity of the students is very difficult to predict. As an intervention the following is possible:

> Keep on collecting data regarding the various estimations. The advice is to use the prediction related to the University of Twente, and to fill this in in Visual Studio Team Services where the capacity is collected. Besides this, if somebody has a significantly divergent capacity (e.g. week of no time or week fulltime), this can be stored instead. Furthermore, the capacity can be without the 'overhead time costs' which are allocated to planning activities to obtain a better view.

The Prediction related to the UT schedule seems most promising. This is considered as a solution in chapter 6.

Meeting preparation

As explained in paragraph 4.2.2 preparing meetings costs time. This doesn't have to be necessary when meetings are always clearly defined, which is the case according to Scrum theory.

First of all, the planning meeting is defined. As explained in paragraph 4.1.1 the planning meeting consists of two major parts. First, the selection of the items. Then, the creation of tasks. Adding the setting of a sprint goal as explained in paragraph 2.2.3 and combining this with the current way of working we can create the following meetings structure:

1. Set sprint goal
2. Select related items from approved items
3. Commit to items (total effort)
4. Assign responsible person per item
5. Create tasks per item
6. Create remaining work estimation per task

Following this structure doesn't require any preparation when approved items are available due to product backlog refinement.

Second, there is the retrospective meeting. As explained in one company interview (Table 13) there shouldn't be too much routine when facilitating communication. This means a retrospective should vary. Therefore, a retrospective should be prepared.

Third, sprint update meetings or the daily Scrum meeting. In the daily Scrum meeting the following three questions are always answered (Mountain Goat Software, 2018):

1. What did you do yesterday?
2. What will you do today?
3. Are there any impediments in your way?

The result of this meeting is the Scrum Masters responsibility as a task. It is no task related to planning. This structure doesn't require any preparation. During the meeting, artifacts are updated

Finally, there's the refinement meeting. During the refinement meeting a product backlog item is first accepted. An item has to be accepted by stakeholders and the development team. During the meeting, only the development team is present. Other stakeholders should give their acceptance elsewhere. If an item isn't accepted it has to be refined, either during the meeting or outside. Once an item is accepted the development team estimates its effort. As long as the product backlog is prioritized, the item with the highest priority can be refined. Then, no preparation is required.

Daily Scrum meetings (time costs)

As explained in paragraph 4.2.5 there is an unusual structure regarding daily Scrum meetings. Ideally, these are done daily with the development team and the Scrum Master. Having less meetings isn't a possibility due to the necessity according to the Scrum methodology as explained in paragraph 2.2.4. However, the artifacts are now changed by anyone, during anytime of any day. According to Scrum this is a part of the Scrum meeting and should be done within the 15 minutes. At this point the team doesn't do this during the Scrum meeting. We can assume that if the meetings were 30 minutes this should be possible.

Knowing this the following structure for Scrum meetings can be proposed:

- Scrum meetings will be held daily using the option to call in from elsewhere at a fixed time.
- Students only join if they have done work the previous day and/or are planning on doing work the current day.
- Otherwise students can view changes on the task board on a dashboard (chapter 6).
- This is the only moment where the Scrum Master makes changes to artifacts.

The reason the Scrum meeting is done every two days now is because students don't work every day. Therefore, the assumption is made that students still won't join every day, but still every other day. In the current situation, as shown in paragraph 4.2.2, the Scrum meetings cost 2 205 minutes per sprint. This would cost (with 2 fulltime employees and 5 students) 1 620 minutes per sprint. The main reason for this is that the artifact updating is centered around the Scrum update meeting.

Redesigning steps and defining tasks

From the paragraphs 4.2.1 and 4.2.2 it has become clear the steps can be redesigned. We add to this the previous interventions and come to the following steps, starting with strategy planning.

In the As-Is model not much attention is paid to strategy planning. The first step is the design of a product idea which is actually product planning.

Table 15 Steps To-Be model (strategy planning)

Planning level	**Activities (in order)**
Strategy planning	Review company vision
Strategy planning	Review company mission

In observation it has been seen that there is attention being paid to portfolio planning with the release of a second product being planned. However, it also isn't a part of the planning process in the As-Is model. The same counts for product planning.

Table 16 Steps To-Be model (portfolio and product planning)

Portfolio planning	Review company goal
Product planning	Review product vision

For the release planning in the As-Is model, a global sprint planning was created. This is not often used in Agile software developing as seen in paragraph 5.1. Therefore, we use the structure of the Ideal model.

Table 17 Steps To-Be model (release planning)

Release planning	Set release goal
Release planning	Create product backlog

As we have defined that the global sprint planning isn't necessary to use, and the items to work on are chosen by the team during the planning meeting as defined in paragraph 4.1.1. The retrospective remains the same but is addressed in the 'End of sprint' level. Furthermore, due to the design of an estimation technique in paragraph 5.2.1, the capacity is no longer actively collected.

Table 18 Steps To-Be model (iteration planning)

Iteration planning	Set sprint goal
Iteration planning	Select related items from approved items
Iteration planning	Commit to items (total effort)
Iteration planning	Assign responsible person per item
Iteration planning	Create tasks per item
Iteration planning	Create remaining work estimation per task

In the As-Is model, there were daily and weekly planning activities. It has been defined in paragraph 4.2.1 that many activities here were extra and related to functions (also due to involvement of only one person). Also, many activities have been moved to the level 'backlog refinement'. And, Scrum meetings will be done daily according to paragraph 2.2.4.

Table 19 Steps To-Be model (daily planning)

Daily planning	Daily Scrum meeting
Daily planning	Update artifacts

Then, there is the level 'backlog refinement'. Where, as explained in this paragraph many activities have been moved to and forged into one process flow.

Table 20 Steps To-Be model (backlog refinement)

Backlog refinement	Assess (new) product backlog items priority
Backlog refinement	Create acceptance criteria for product backlog items
Backlog refinement	Present items to Stakeholders
Backlog refinement	Present items to Development team
Backlog refinement	Adjust/remove item
Backlog refinement	Adjust/remove item
Backlog refinement	Estimate effort for item

Then, there is the end of the sprint which contains some new activities with respect to the As-Is model due to the implementation of continuous development.

Table 21 Steps To-Be model (end of sprint)

Product improvement	Have Review meeting
Product improvement	Increment product
Product improvement	Collect feedback from Users
Process improvement	Prepare Retrospective meeting
Process improvement	Have Retrospective meeting

Now, quite some steps that are present in the As-Is model are no longer present. This is because these activities are tasks that are inherent to a specific function. Steps that are left over from the As-Is model but can be viewed as functional tasks. Where the Product Owner should always prioritize the backlog in VSTS and prepare the retrospective, when is decided that he/she is responsible. Also, VSTS should be checked. The Scrum Master does this as well. The Product Owner and Scrum Master should have a clear understanding about this, so that two people don't check the same thing. Finally, the development team should always update their work in the sprint backlog (e.g. remaining hours).Any other activities aren't of vital importance to the planning process.

5.2.2 RACI

In this paragraph the new division of responsibilities is discussed. And a solution for the absence of a dedicated Scrum Master is presented.

Division of responsibilities new steps

The newly created steps have all been assigned RACI-roles. We allocate the roles according to the division of responsibility from paragraph 4.1.2. This yields the division of roles as shown in Table 22.

Table 22 Activities To-Be model with roles

Activities (in order)	**R**	**A**	**C**	**I**
Review company vision (why)		BO	BD PO	SM DT
Review company mission (how)		BO	BD PO	SM DT
Review company goal (what)	BD	BO	PO	SM DT
Review product vision	PO	BO	BD DT	
Set release goal	PO	BO	BD DT	
Create product backlog		PO	BD DT	BO
Set sprint goal	DT	PO		SM
Select related items from approved items	PO DT			
Commit to items (total effort)	PO DT		SM	
Assign responsible person per item		DT	SM	
Create tasks per item		DT	SM	
Create remaining work estimation per task		DT	SM	
Daily Scrum meeting	DT		SM	
Update artifacts	SM		DT	
Assess (new) product backlog items priority	PO			DT US
Create acceptance criteria for product backlog items	PO			
Present items to Stakeholders		PO	US	
Present items to Development team		PO	DT	
Adjust/remove item	PO			US
Adjust/remove item	PO			DT
Estimate effort for item	DT	PO	SM	
Have Review meeting	DT		PO US	
Increment product		PO		BO BD DT US
Collect feedback from Users		PO		BO BD DT US
Prepare Retrospective meeting	SM			
Have Retrospective meeting		SM	PO DT	

Some activities have a red shade. These will be discussed further in this paragraph.

Scrum Master/Developer

The red-shaded activities in Table 22 are activities where the Scrum Master role cannot be combined with a role in the development team. For some activities it is possible for the Product Owner to take over the role of Scrum Master. In other cases another solution is necessary. A redesign of the roles is shown in Table 23. The final division of roles is shown in appendix 9.20.

Table 23 Improved RACI roles To-Be model

Commit to items (total effort)	PO	DT		SM								
Assign responsible person per item			DT	PO								
Create tasks per item			DT	PO								
Create remaining work estimation per task			DT	PO								
Daily Scrum meeting	DT			SM								
Update artifacts	SM			DT								
Estimate effort for item	DT		PO	SM								
Prepare Retrospective meeting	SM											
Have Retrospective meeting			SM	PO	DT							

For the remaining red-shaded activities a developer is forced to take on the role of Scrum Master. This is no easy task, because then one person must wear to hats. A way to share this responsibility is by passing on this role within the team either per week or per sprint. Then, the load is carries on more shoulders. For the retrospective it could be good when the meeting could be leaded by somebody outside of the team. The (weekly or sprintly) Scrum Master can still be responsible for the occurrence of the meeting. This way the load of being the Scrum Master is shared with the entire team and a learning experience is provided. Ideally, the role would only be passed on among fulltime developers. However, there could be experimented with this.

5.3 To-Be model

With the contents of paragraph 5.2 a To-Be model has been created which is shown in Figures; Figure 28, Figure 29 and Figure 30. This model starts with the organizational levels named strategy, portfolio and product planning. Here a company vision, mission and goal are created. Furthermore, a product vision is formed. This is followed by release planning where a decision is made for an Epic and the product backlog is created consisting of the supporting features. Then, we move to iteration planning. This is done during the planning meeting. During the iteration there are daily scrum meetings where artifacts are updated. During any working period, there is also backlog refinement going on. A part of this backlog refinement is the refinement meeting. The backlog refinement of a single item end with this item being marked as ready (to be worked on). Finally, the product as well as the process can be improved via a review meeting, retrospective meeting and the collection of feedback from users.

To show the improvement of this model, the time costs are compared in Table 24. It is important to remind that there are no daily Scrum meetings on Friday due to the absence of fulltime developers. In this case, one of the fulltime developers would still remain Scrum Master (for comparison). Regarding the roles as defined in paragraph 5.2.2 a complete list is shown in appendix 9.20.

Table 24 Comparison As-Is and To-Be time costs

Function	Hours available	Hours spent on planning (As-Is)	Percentage overhead (As-Is)	Hours spent on planning (To-Be)	Percentage overhead (To-Be)
Product Owner	48	55	115%	25	52%
Scrum Master	108	24	22%	15	14%
Fulltime developer	96	15	16%	14	15%
Students	145,6154	63,75	44%	55	38%
Total	**397,6154**	**157,75**	**40%**	**109**	**27%**

Users are actively involved for 30 minutes per sprint. And, for the To-Be model activities where a function is either responsible, accountable or consulted are considered time costing. Being informed shouldn't cost time on a regular basis.

We can now also asses the difference when changing the sprint length to 2 or 4 weeks considering the To-Be model. Then, we get the time costs shown in Table 25.

Table 25 Comparison To-Be model 2 weeks, 3 weeks and 4 weeks, time costs

Function	Hours available (2 weeks)	Hours spent on planning (2 weeks)	Percentage overhead (2 weeks)	Hours available (3 weeks)	Hours spent on planning (3 weeks)	Percentage overhead (3 weeks)	Hours available (4 weeks)	Hours spent on planning (4 weeks)	Percentage overhead (4 weeks)
Product Owner	32	17,83	56%	48	25	52%	64	32,17	50%
Scrum Master	72	11,5	16%	108	15	14%	144	18,5	13%
Fulltime developer	64	10,5	16%	96	14	15%	128	17,5	14%
Students	97,0770	37,5	39%	145,6154	55	38%	194,1539	67,5	35%
Total	**265,0770**	**77,33**	**29%**	**397,6154**	**109**	**27%**	**530,1539**	**135,67**	**26%**

Here we can see there is a minor gain of time for development when comparing a 4-week sprint to a sprint of 3 weeks. If the sprint length were changes to 2 weeks, the overhead will only be more. As 3 weeks has been proven to be a good sprint length according to the team a change seems unnecessary.

The division of responsibility in the To-Be model is shown in Figure 27.

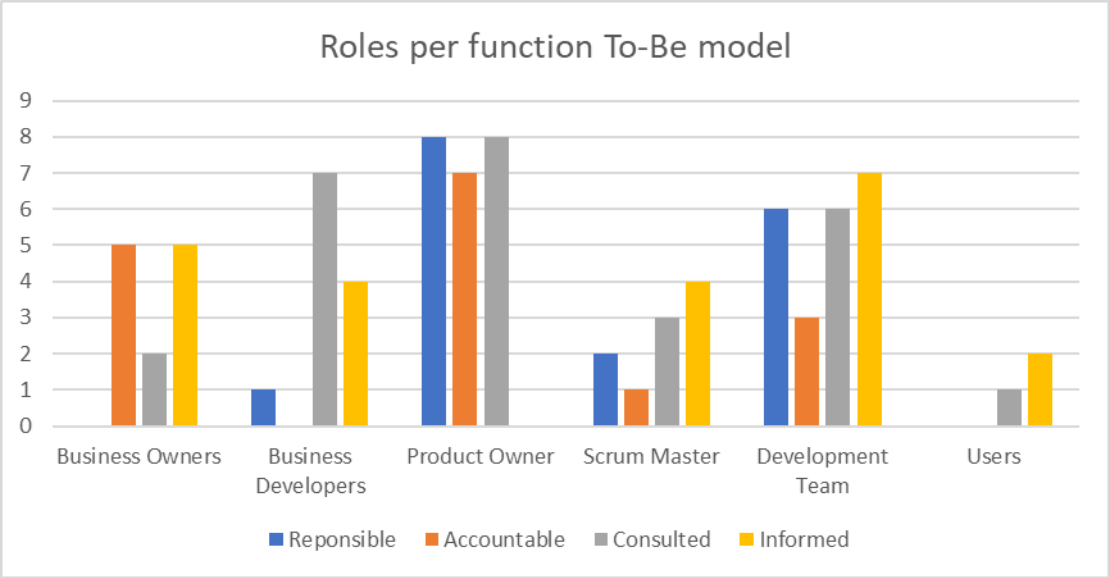


Figure 27 Roles per function To-Be model

It can be seen that different functions are often informed, compared to the As-Is model. Furthermore, the responsibility is more spread out between the Product Owner, and development team. It would have been more desirable to give more responsibilities to the Scrum Master. But as explained, this is not possible for Polder Valley in the current composition. Last, Business Developers are often consulted.

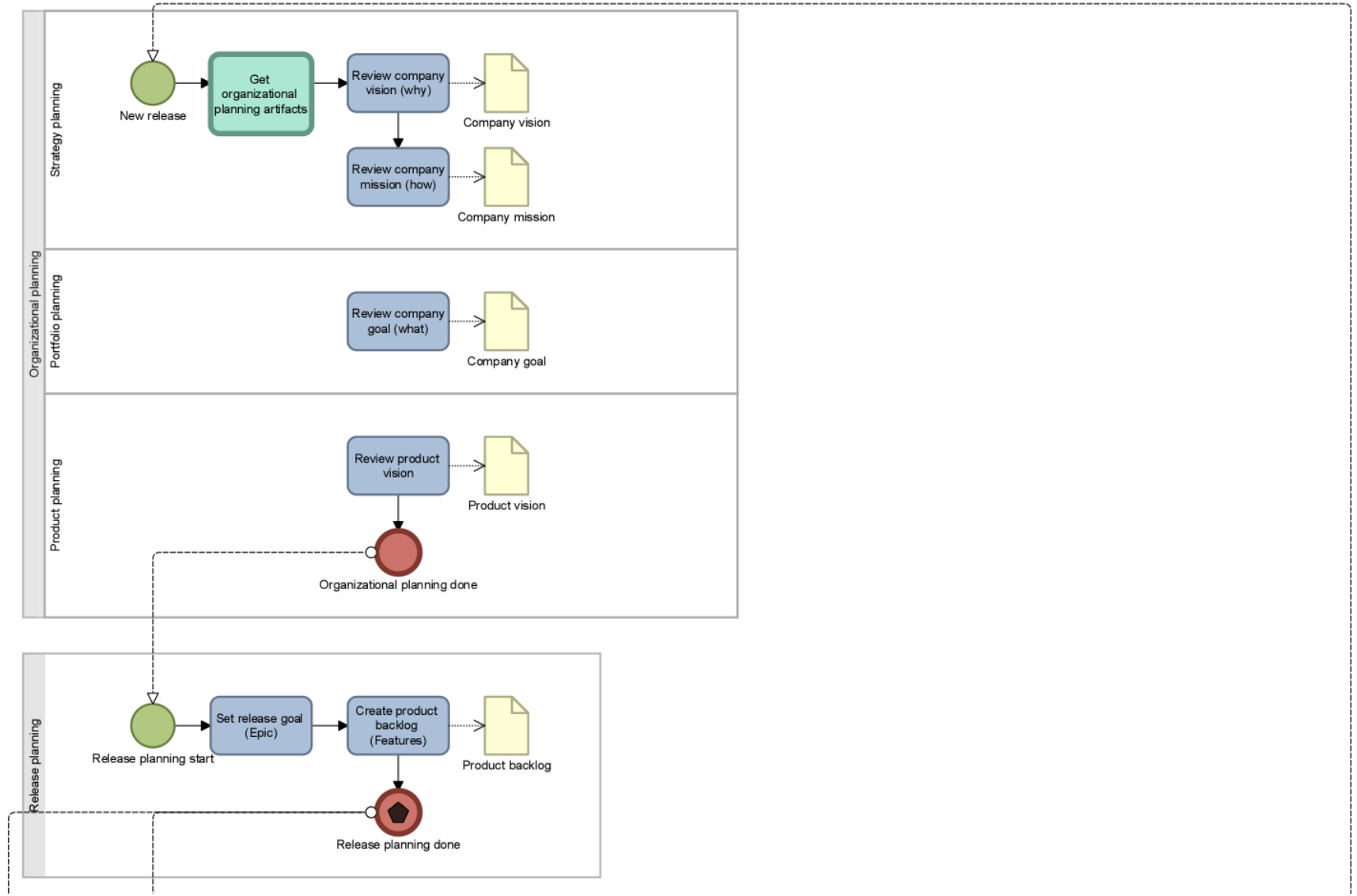


Figure 28 To-Be model 1/3

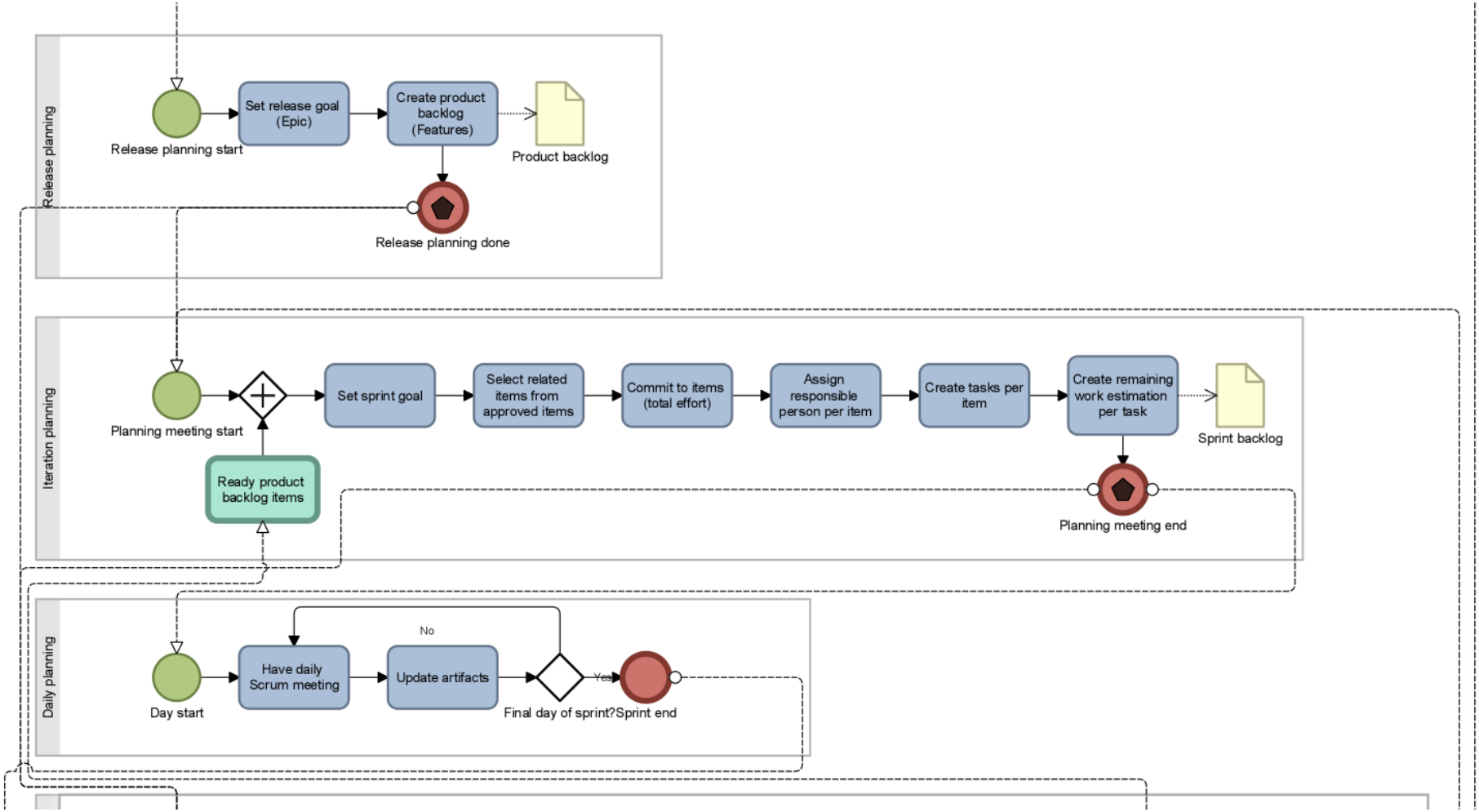


Figure 29 To-Be model 2/3

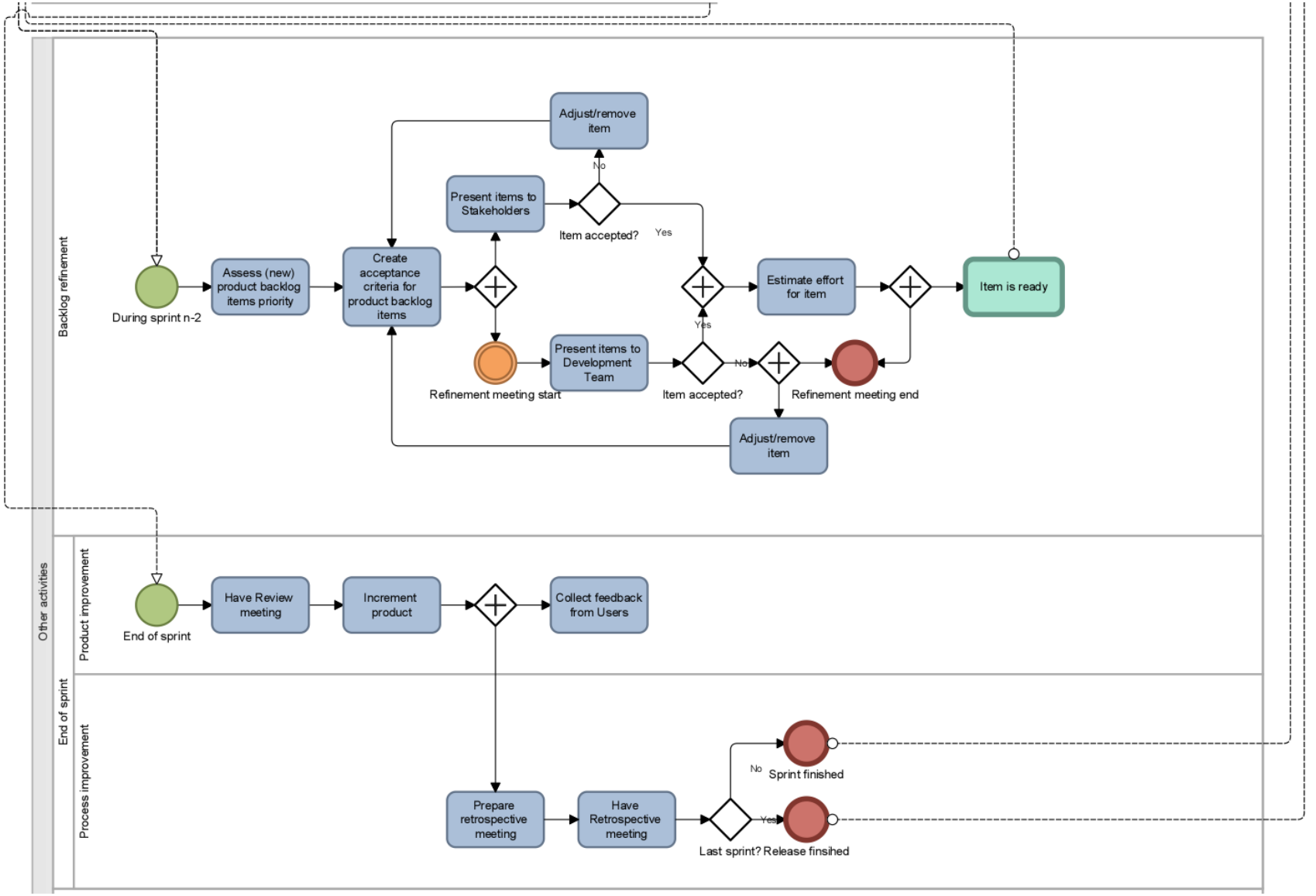


Figure 30 To-Be model 3/3

5.4 Planning and scheduling variables to make measurable

As shown in paragraph 4.2.4 not many variables are used in relation to the planning and scheduling process. In paragraph 2.2.5 many metrics have been identified which can be made measurable. However, the focus of this research is on the planning process. Therefore, we will focus more extensively on metrics that support this. For example, it's undesirable that it costs time to provide certain information that is relevant in the planning process. As it is more ideal to have information at hand instantly. So, we will focus on how to use capacity, remaining work and effort in a more efficient way. As in the current situation, time is spent to collect data regarding these variables. This part is continued in chapter 6.

6 Solution development

In this chapter a practical solution is developed building forward on the variables presented in paragraph 2.2.5. In this chapter an overview is shown of the planning tool. In appendix 9.20 a more detailed description is given where first, a design is made for a planning tool. Second, a plan is made for the use of data regarding the tool. Last, visualizations of metrics are defined and designed.

As at Polder Valley there is worked with Microsoft applications. MS PowerBI is an application which enables users to create dashboards using live data. PowerBI can be connected to VSTS and data can be filtered to do so. The application will be accessible via an URL when the entire team has a MS PowerBI pro license. An example of the environment is shown in Figure 31.

Figure 31 PowerBI web example

Then, the application has different tabs, showing different dashboards. The tabs can be easily accessed as shown in Figure 32.

Figure 32 PowerBI tabs example

Throughout the tables shown in Table 26 are used.

Table 26 Tables used in planning tool

Table	Created	Used in tab
Features backlog (ID and Title)	Automatically	Release progress
Items backlog	Automatically	Capacity estimation
Tasks backlog	Automatically	Sprint progress, Task changes
Capacity from VSTS	Automatically (filled in in VSTS)	Capacity estimation, Sprint progress
TimeWriter data	Manually/automatically (more research needed)	Capacity estimation
UT-weeks overview (e.g., quartile 1, week 5)	Manually (yearly)	Capacity estimation
Iteration overview	Automatically	Capacity estimation
Combined table iteration overview and UT-weeks	Automatically	Capacity estimation
Burndown chart table	Automatically	Sprint progress
Table of Scrum planning moments to be made into tasks	Automatically	Sprint progress
Epic burndown chart table	Automatically	Release progress

Using these tables several dashboard items are created per tab. The ‘Capacity estimation’ tab contains the following items:

- Graph showing average hours worked per UT-week, per student (one average)

- Indicator showing the expected weekly capacity per student for the upcoming sprint (one value)
- Graph showing accuracy of estimation per sprint
    - KPI showing the mean absolute error

This dashboard can theoretically replace the need for individual capacity estimation and provide key insights regarding the capacity of students.

The 'Sprint progress' tab shows the following items:

- Burndown chart with an ideal-line adjusted to capacity
- List of sprint ceremonies to be added as tasks in the VSTS backlog (with the remaining work)

This dashboard provides a more correct overview of the sprint progress compared to the Burndown chart shown in VSTS.

The 'Release progress' tab shows the following items:

- Velocity chart with estimation for upcoming sprints which could be shown in points or hours worked based on the last n sprints. It can include an average based on the trend and include a (100-α)% confidence interval. Also, a line for the maximum and minimum of the last n sprints could be added. In Figure 33 an example velocity chart is shown prediction sprint 18, 19 and 20 based on the last 17 sprints where the main prediction is 35 points for sprint 19. The lower and upper limit of the 5% confidence interval included show 29 and 40 points. Moving the cursor over sprint 18 or 20 would show the prediction for these sprints. This chart was created in PowerBI.

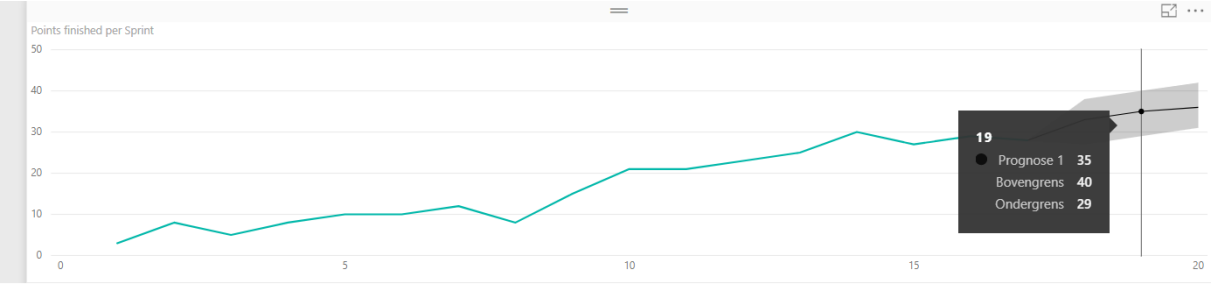


Figure 33 A Velocity chart example including a confidence interval

- Epic/Feature burndown, as seen in Figure 34, which also includes an overview of points that have been added during sprints and a prediction of how many more sprints are necessary to finish the Epic. A similar graph can be created for a single feature.

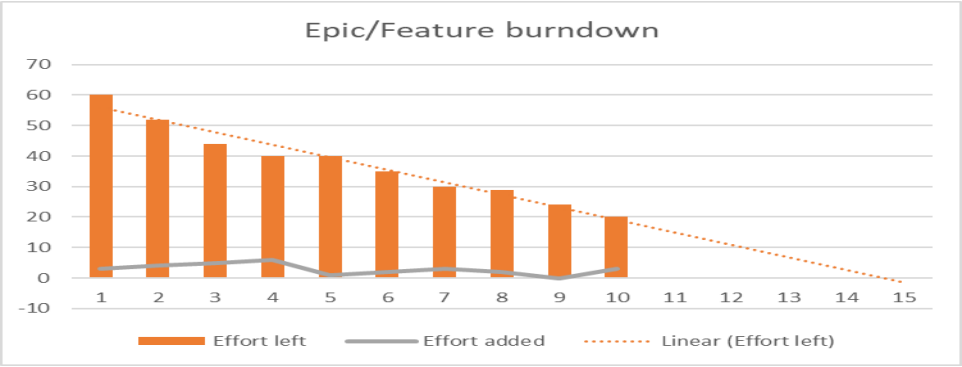


Figure 34 Epic/Feature burndown example

This tool aids in estimating the effort for an upcoming sprint. Theoretically the item selection for a sprint can be based in this information.

The ‘Task changes’ tab shows the following item:

- A table with an overview of recent task changes

This enables team members who can’t be present during a daily Scrum meeting to easily stay updated regarding progress.

The tool is fully customizable after creation, there is a possibility to add and change dashboard and their items.

In this chapter, there is shown what the tool could be like. However, in the current situation this serves as an inspiration for Polder Valley towards using metrics more. Somebody from the team could be made responsible for the creation and maintenance. Or another way could be found to make metrics work for the team in an effective manner.

7 Evaluation

In this chapter solutions from chapters 5 and 6 are evaluated. First, interventions are evaluated. Then, the planning tool is evaluated.

7.1 Interventions

Two main interventions proposed are the implementation of the To-Be model and the Planning tool. The To-Be model includes the daily Scrum meeting becoming daily, redefined steps and a proposed division of responsibility. The planning tool includes an optimization of the capacity estimation and makes it possible to have daily Scrum meetings every day. Besides these interventions, a clearly defined meeting structure has been proposed and it has been advised to pass on the Scrum Master role.

When management would decide to implement one or more interventions these should be evaluated. The To-Be model can be evaluated by re-assessing the time costs after implementation. The planning tool can be evaluated by calculating the usage and using a similar evaluation as shown in paragraph 7.2. The meeting structures and circulating Scrum Master role can be evaluated if necessary during retrospective meetings. Also, time costs can be considered for these two interventions.

7.2 Use of planning tool

The proposed planning tool has been evaluated according to the Unified Theory of Acceptance and Use of Technology (Venkatesh, Morris, Davis, & Davis, 2003). Venkatesh et al. compare several technology acceptance models to create the model in Figure 35.

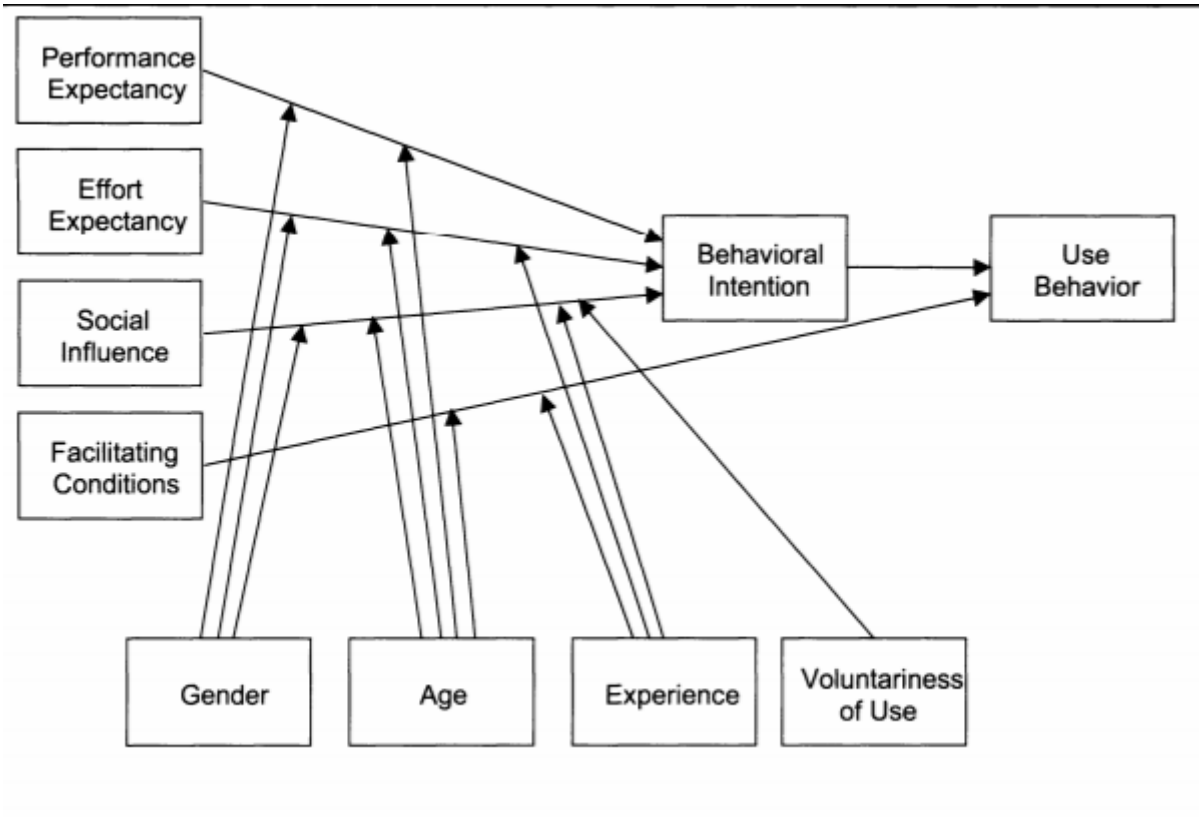


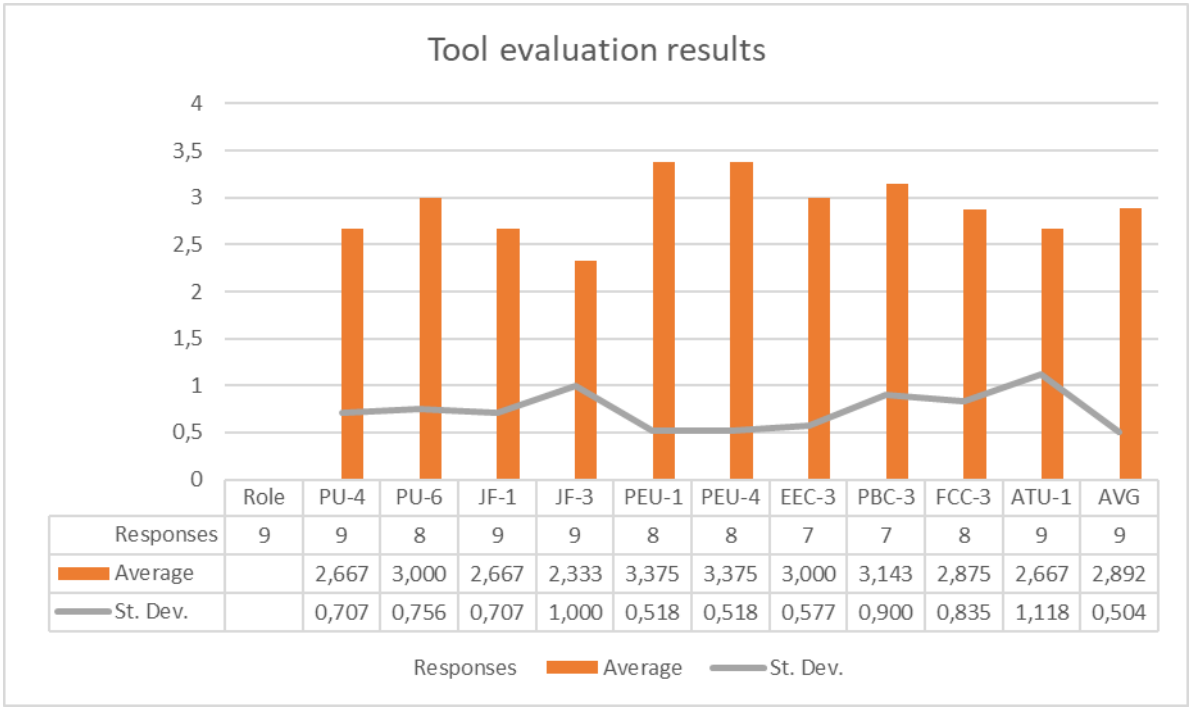
Figure 35 Research Model UTAUT according to Venkatesh et al.

Some questions have been taken identified by Venkatesh et al. that seemed fitting to assess the planning tool design. The questions shown in Table 27 have been used.

Table 27 UTAUT questions planning tool

Construct	Code	Questions
Performance Expectancy	PU-4	Using the system would enhance my effectiveness on the job.
Performance Expectancy	PU-6	I would find the system useful in my job.
Performance Expectancy	JF-1	Use of the system will have a positive effect on the performance of my job.
Performance Expectancy	JF-3	Use of the system can significantly increase the quality output of my job.
Effort Expectancy	PEU-1	Learning to operate the system would be easy for me.
Effort Expectancy	PEU-4	I would find the system to be flexible to interact with.
Effort Expectancy	EEC-3	Using the system involves little time doing mechanical operations (e.g., data input).
Facilitating Conditions	PBC-3	I have the knowledge necessary to use the system.
Facilitating Conditions	FCC-3	Using the system fits into my work style.
Attitude Towards Technology	ATU-1	The system makes work more interesting.

The choice of questions was based on sample questions from Venkatesh et al. which were applicable. Participants were shown Chapter 6 and asked these questions. The results of the questionnaire are shown in Figure 36. There were 9 respondents who answered all questions. Answers were operationalized by deciding I totally disagree = 1 to I totally agree = 4. When Not applicable was answered this is not taken as a response to the question.



	Role	PU-4	PU-6	JF-1	JF-3	PEU-1	PEU-4	EEC-3	PBC-3	FCC-3	ATU-1	AVG
Responses	9	9	8	9	9	8	8	7	7	8	9	9
Average		2,667	3,000	2,667	2,333	3,375	3,375	3,000	3,143	2,875	2,667	2,892
St. Dev.		0,707	0,756	0,707	1,000	0,518	0,518	0,577	0,900	0,835	1,118	0,504

Figure 36 Tool evaluation results

We can see from the results that, in general, Polder Valley would adopt the created tool. However, responses seem quite different and the standard deviation is relatively high. Many different answers have been given. Therefore, we review the individual results to assess differences in Figure 37. However, we do see that PEU-1 scores the highest. This means adopting to the tool would be easy for the team in general. JF-3 has the lowest score. So in general, the tool wouldn't increase the quality output of the team. This could be because most of the respondents aren't directly responsible for improvement of the planning process. For example, the tool wouldn't improve the quality of any code. Which is the main concern of many respondents.

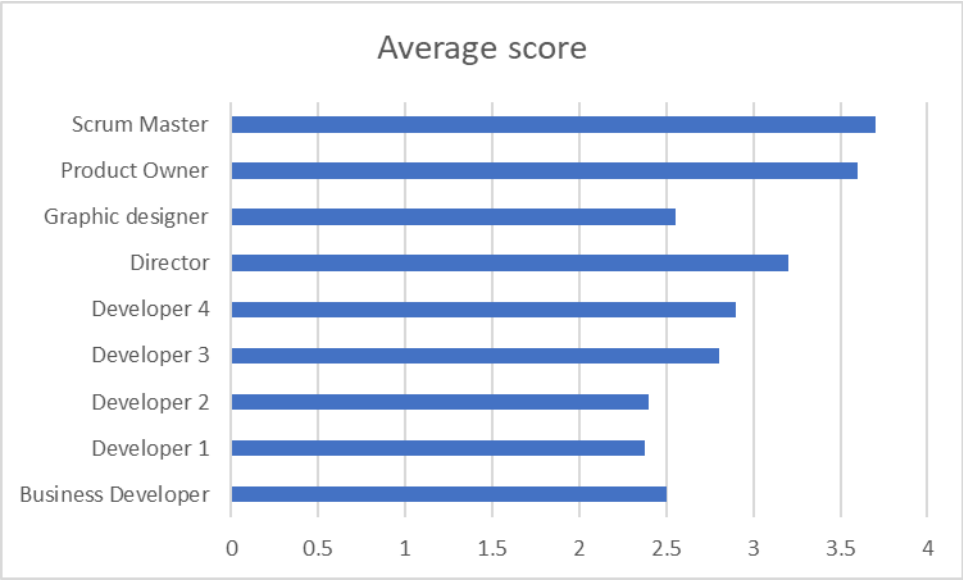
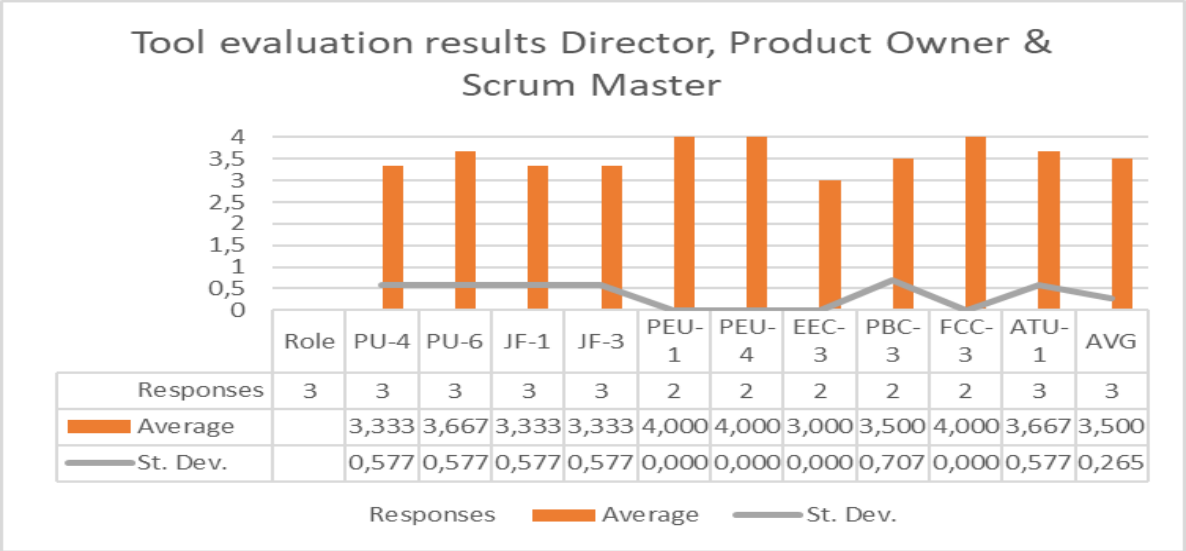


Figure 37 Individual results tool evaluation questionnaire

Here we can see that there are three potential users who would fully agree they would use the tool because it is beneficial. The Scrum Master, Product Owner and Director (or Business Owner). These are also the people that hold more responsibilities via planning activities. Therefore, it is logical that they are more prone towards adaption of the tool. They can also be seen as the main audience for the tool. Their results are shown in Figure 38.



	Role	PU-4	PU-6	JF-1	JF-3	PEU-1	PEU-4	EEC-3	PBC-3	FCC-3	ATU-1	AVG
Responses	3	3	3	3	3	2	2	2	2	2	3	3
Average		3,333	3,667	3,333	3,333	4,000	4,000	3,000	3,500	4,000	3,667	3,500
St. Dev.		0,577	0,577	0,577	0,577	0,000	0,000	0,000	0,707	0,000	0,577	0,265

Figure 38 Tool evaluation results Director, Product Owner & Scrum Master

These results show a very positive view regarding the possible acceptance of a planning tool. For every question there is agreeance regarding the statement. Also, the standard deviation is low due to the fact that only 3 and 4 ratings were given. This provides assurance regarding the added value of a planning tool for Polder Valley.

8 Conclusions and recommendations

In this chapter, first, conclusions are presented. For each research question there is presented what has been done to answer it and what the results has been. Second, recommendations are given. This also includes contributions. Third, as discussion regarding validity, reliability and limitations is held. When addressing limitations possible future studies are also identified. Finally, the contributions to Polder Valley are summarized and a reflection is done from the researcher's point of view.

8.1 Conclusions

In this paragraph each research question is addressed separately. Some research questions have been broken down into smaller questions which together answer the main research question. Each time a recap is shown regarding the work done and main conclusions are summarized.

8.1.1 What is the current planning/scheduling flow at Polder Valley?

To answer this question the As-Is model from chapter 3 has been created. The BPMN model, containing elements of RACI and VSM, focused on steps taken, processing time, role division (regarding responsibility) and planning/scheduling variables. The data which enabled us to create the model was collected by observation of planning moments and an interview with the Product Owner, who also provided verification.

Mapping the current situation, several interesting things were concluded. First of all, Polder Valley is currently assessed as a company striving to reach the second level of the Capability Maturity Model. The second level characteristics should be achieved before focusing on higher level characteristics. Second, as every other organization, Polder Valley has their own structure. Scrum, a Kanban-board, VSTS and planning cycles are used. Third, The Scrum Master who is also a developer provides an interesting combination. Last, the management of Polder Valley wishes to perform Scrum as an Agile methodology. However, there are also many aspects present of other methodologies present in the organization.

The entire team provided data for the creation of the model. There were little problems with noise, and the ones that did occur were dealt with properly. Business owners and developers weren't considered in the analysis as the research was focused on the development team (including the Product Owner and Scrum Master). The following levels of planning were clearly present: Strategic and release planning, Sprint planning, Weekly planning, and Daily planning.

The creation of an overview provided a clearer insight in the current situation. Further analysis shows that the Product Owner has more hours of work then he has available per sprint (55/48), and students spend almost half of their time on planning related activities (44%). The view of the team regarding the value added of by activities to overall performance was also analyzed. But no clear improvements were uncovered when analyzing the value addition of activities. Regarding the roles, the Product Owner has a lot of responsibility compared to any other group within the organization and people aren't often informed. They are either responsible or consulted. Furthermore, fulltime developers don't have more responsibility than students. Which is questionable due to their part regarding the total number of hours worked. Last, it has been seen that are estimations done for capacity, remaining work per task, priority, and effort per backlog item.

8.1.2 What is the ideal planning/scheduling flow in an Agile software development environment?

To answer this question the Ideal model from paragraph 4.1 has been created. It's a BPMN model similar to the model described in paragraph 8.1.1. Data collection has been done by answering the three following questions using literary sources and considering the view of the management of Polder Valley. The main conclusions are given after the answers to the three sub-questions.

Which Agile software development methodologies can be used?

To answer this question a systematic literature review has been performed in paragraph 2.2.2 to identify different Agile software development methodologies.

From the review there could be concluded that there is no best practice of Agile. Everybody is moving from traditional planning methods to Agile methods. However, upfront planning is still necessary. Also, planning games are often used for effort assessment. In conclusions it is shown that there are many different Agile methodologies and applications of them.

How can Agile software development be used according to literature?

To answer this question literary sources were collected regarding Agile software development. Information was collected in paragraphs 2.2.3-2.2.5.

First, Agile teams are mostly concerned with the three most inner levels of the planning onion. To aid in planning, goal setting can be done according to the Golden Circle by Simon Sinek. When considering how to work Agile it is seen that Scrum is the most widely used Agile software development methodology. Furthermore, there are many variables that influence success. It is good to have an insight in some of these variables using metrics. There are various categories of metrics which can be assessed separately.

How does Polder Valley want to perform planning/scheduling?

To answer this question, the management of Polder Valley has been consulted regarding the division of roles (regarding responsibility) in paragraph 4.1.2.

When doing this, stakeholders outside of the development team have also been involved in the process as the focus is on a situation where continuous development is present. Viewing the result, it stands out that Business Owners are either accountable or informed during the process, where one could say they are merely informed to be able to make the right decisions. Also, the development team bears the most responsibility during the process.

Main conclusions

All the preceding made it possible to create the Ideal model, which is an (unreachable) benchmark for the planning of software development for Polder Valley. For the Ideal model the number of fulltime employees has been chosen which brought us the closest to the current situation (3). Analyzing the Ideal model, we find that only 12% of total time available is spent on planning activities. The rest can be spent on functional tasks.

8.1.3 How does the current situation differ from the ideal situation?

To answer this question the gap analysis has been performed in paragraph 4.2. This was done by comparing the As-Is model and Ideal model, which served as data sources. Doing this, we were able to answer the two following questions. Both questions are addressed throughout the paragraph. The conclusions regarding both questions are presented separately.

How does the As-Is model differ from the Ideal model?

First, extra steps taken, and missing steps have been identified. Extra steps have been dealt up in categories: Sprint design is done by the Product Owner and is a part of his tasks. The same counts for updating artifacts (Development team and Scrum Master). Furthermore, there are some automated processes. Capacity estimation and meeting preparation are however not of vital importance for the planning process. From the missing steps it is seen the first four steps of the Ideal process are absent in the As-Is model. Also, activities related to product improvement aren't done due to the absence of continuous development.

Where in the As-Is model 40% of all time is spent on planning activities, this is 12% in the Ideal model. About 18% of all time spent in the current situation is due to the extra steps identified. From all this, only a small fraction is spent on capacity estimation and meeting preparation. So, its mostly time spent on functional tasks. When we compare the activities which overlap with one another we notice two main things. First, regular activities such as meetings cost more time in the As-Is model. This is due to the students in the development team, which result in more people present, which means more time is spent as a team. Second, the daily Scrum meeting structure which is meant to be less constraining for students costs the team more time than a structure in the Ideal situation would have. Even though in the Ideal situation there are 5 meetings a week, where in the current situation this is two meetings.

Focusing on the division of roles we first see that due to the combination of a Scrum Master and developer role the Scrum Master is sometimes absent in the current situation. Activities that would ideally be performed by the Scrum Master are now performed by the Product Owner in some cases. Also, Business Developers aren’t involved actively in many processes. Furthermore, stakeholders outside of the development, such as users, team aren't involved in backlog refinement.

Regarding planning and scheduling variables which can be understood through metrics, we have seen that not many metrics are being used. Possible metrics can come from the following 5 categories:

- Sprint and project planning
- Sprint and project progress tracking
- Understanding and improving quality
- Fixing software process problems
- Motivating people

Which differences are unique for Polder Valley?

When identifying problems in the gap-analysis we have seen that steps can be redesigned in the To-Be model, as well as deviating (RACI) roles. To make more use of metrics, a planning tool has been designed. Matters that aren’t solved by the two preceding solutions can be seen as unique for Polder Valley. So first of all, this is the full involvement of students in the Agile software development team, which leads to the necessity of a thorough capacity estimation and a deviating daily Scrum meeting structure. Second, this is the absence of a dedicated Scrum Master which leads to complications with the division of roles. For these unique aspects original solutions have to be designed.

8.1.4 How can the planning process of Polder Valley be improved?

To answer this question solutions have been designed and a To-Be model has been created in chapter 5. All preceding research has been combined with data collected through interviews with external companies. The interviews with experts at external companies gave answers to the first question. The two following questions are answered throughout the chapter. The To-Be model is a model similar to the As-Is and Ideal model and is presented in paragraph 5.3.

How is the planning/scheduling done in comparable situations/companies?

To answer this question interviews have been done with experts from external companies, which are summarized in paragraph 5.1.

It became most clear that Agile is used very differently. Usually teams adapt Agile to their specific needs. Mostly this provides a nice overview which reflects the way successful companies apply software development. It also portrays how experimental the field of Agile software development still is.

How can differences/sources of waste can be resolved using interventions?

To answer this question interventions have been designed related to VSM and RACI literature in paragraph 5.2.

First of all, various capacity estimations were created and tested. However, none of the created estimations was very accurate. The best scoring estimation was the one based on the UT-weeks with a Mean Abs. Error of 26,62% and a Mean Error of -1,67%.

Second, clear descriptions have been created for the planning meeting, daily Scrum meeting and refinement meeting. There shouldn't be any preparation necessary. Proper execution of functional tasks should be enough for the meetings to take place. The retrospective meeting should be prepared, as it has been seen that a lack of routine leads to lesser communication. Therefore, variety is necessary. Going into depth on the daily Scrum meeting, we see that updating the team as well as artifacts can be combined. Therefore, the following structure is proposed:

- Scrum meetings will be held daily using the option to call in from elsewhere at a fixed time.
- Students only join if they have done work the previous day and/or are planning on doing work the current day.
- Otherwise students can view changes on the task board on a dashboard (chapter 6).
- This is the only moment where the Scrum Master makes changes to artifacts.

This should save the team 575 minutes per sprint.

Third, combining steps from the As-Is and Ideal model steps for the To-Be model have been created. Steps that are left over from the As-Is model but can be viewed as functional tasks. Where the Product Owner should always prioritize the backlog in VSTS and prepare the retrospective, when is decided that he/she is responsible. Also, VSTS should be checked by the Product Owner. The Scrum Master does this as well. The Product Owner and Scrum Master should have a clear understanding about this, so that two people don’t check the same thing. Finally, the development team should always update their work in the sprint backlog (e.g. remaining hours).

Fourth, through combining the ideal division of responsibility with the newly designed steps of the To-Be model an overview of the roles has been created where it becomes clear that in some cases the Scrum Master needs to be the Scrum Master as well as a developer. For the item division and task creation, during the planning meeting, the Product Owner could take over this role. In other cases it's

advised to pass on this role within the development team per sprint. Then, the load of being the Scrum Master is shared, and a learning experience is provided for the entire team.

Last, a To-Be model has been created containing all proposed interventions. To compare the To-Be model the percentage of hours spent on planning activities has been calculated at 27% (13% lower than in the As-Is model). The To-Be model also enables us to analyze the drop in overhead hours (related to planning) when a sprint would be 4 weeks (instead of 3). Calculating the difference shows a drop of 1%. Shortening the sprint length to 2 weeks only causes overhead time to rise.

Which variables should be used for the planning and scheduling at Polder Valley?

To answer this question several variables have been identified in paragraph 5.3. Here it has been decided that the focus should be on metrics that support the planning process, as the scope of this research is on planning.

8.1.5 How can the collected knowledge be used in a planning/scheduling tool?

To answer this question a planning tool has been designed in chapter 6. Knowledge from answers of preceding questions, especially regarding planning and scheduling variables has been combined with interventions that require certain information to create a conceptual planning tool. Ideas for various metrics have been identified by studying literary sources.

A conceptual application has been designed which provides insights in the capacity estimation proposed, sprint progress, release progress and task changes. The conceptual planning tool has been evaluated using the UTAUT by Venkatesh et al. In this evaluation Business Owners, the Product Owner and the Scrum Master have been identified as users with the highest potential of adopting the tool.

8.2 Recommendations

Several recommendations can be made to Polder Valley. The recommendations are grouped in relations to their type.

8.2.1 Planning steps

Regarding the planning steps, it's first of all important to consider all levels of the planning onion, starting with strategic planning. Also, goals should be set often during the planning process. The Golden Circle by Simon Sinek can be used to structure related goals. Second, the capacity estimation and meeting preparation activities should be improved in line with the conclusions of this research. Stick to meeting descriptions that have been agreed upon by the team. This provides clarity. When something should be changed in a meeting this can be discussed during a retrospective meeting. Third, when continuous development becomes relevant, activities that add to product improvement should be clearly defined. This can also be done according to the conclusions of this research. Fourth, the daily Scrum meeting structure should be revisited. In this case the artifact updating process should also be taken into account as the two can be combined. The redesign of these activities can again be done in line with the conclusions of this research. Fifth, don't make the sprints longer, as it doesn't decrease overhead significantly. If time would have to be saved, perhaps the retrospective meeting could be done every other sprint. But this should only be done when the team doesn't require a sprintly retrospective meeting. In general, implement the changes in the To-Be model. They bring the organization a step closer to the Ideal model.

8.2.2 Division of responsibilities

All roles should be clearly defined (e.g. Business Owners, Business Developers, other Stakeholders and Users) so it is known who should be involved in certain situations. There can be overlap within roles. There is no continuous development being done in the current situation. This means the Stakeholders are still present, only from within the company. Therefore, they can still be identified and involved. Also, it's important to create consensus regarding the division of roles (regarding responsibility) within the organization. This could be done by assessing everybody's view on the ideal role division and comparing that to the current division. Furthermore, start passing on the Scrum Master role within the development team and evaluate this after some time.

8.2.3 Planning tool

Regarding the capacity estimation part of the tool, the estimation related to the UT-weeks can be used but improved. The best way would be to collect more data and use Machine Learning to create the best possible prediction. This is also related to the division of responsibilities but identify who would be responsible for different kind of metrics. Create the opportunity for people to create dashboards or find information that is relevant for Polder Valley when it adds to process or product improvement. This can also be seen as a 'part of the job'. It fits in the Agile methodology which says teams should be able to divide their attention to different aspects in a short time-frame. Regarding the tool design in general, the tool should be developed using input of Business Owners, the Product Owner and the Scrum Master. They have shown to be keen to adopt the tool.

8.2.4 Agile methodology

Regarding the Agile methodology there has been committed to Scrum. Then, it is important to start focusing on implementing Scrum 100%. Later on, elements can always be dropped when seen fit and there can be focus on integrating other methodologies or even switching. This does not mean that elements that are present now from other methodologies should be dropped. With regards to this, lessons can be learned from other companies, but never try to copy a different company. In the end, everybody does in in their own way, which works for them.

8.2.5 Expanding the team

It has been shown that compared to students, fulltime employees are more efficient. If management would want to expand the team's capacity, it would be advisable to hire more fulltime developers.

8.2.6 Self-assessment and evaluation

Self-assessment and evaluation can always provide insights in a company state. However, when re-assessing the CMM-level this should be done when continuous development is a part of the process. This is a better point to start measuring, as the companies state becomes more stable it this point. Furthermore, when management would decide to implement one or more interventions these should be evaluated. The To-Be model can be evaluated by re-assessing the time costs after implementation. The planning tool can be evaluated by calculating the usage and using a similar evaluation done for the conceptual planning tool. The meeting structures and circulating Scrum Master role can be evaluated if necessary during retrospective meetings. Also, time costs can be considered for these two interventions.

8.3 Discussion

In this paragraph there is first, a discussion about validity and reliability. Then, limitations and opportunities for future research are identified.

8.3.1 Validity and reliability

In order to execute a correct research validity, reliability and limitations need to be taken into account where necessary. For the definitions of validity and reliability we take the following (Cooper & Schindler, 2014, p. 257):

- *Validity* is the extent to which a test measures what we actually wish to measure.
- *Reliability* has to do with the accuracy and precision of a measurement procedure.

Validity can be split up in internal and external validity. Where internal validity is "the extent to which the experiment is free from errors and any difference in measurement is due to independent variable and nothing else". External validity is "the extent to which the research results can be inferred to world at large" (Surbhi, 2017).

This research can be seen as internally valid, but not externally. As the research was very specifically done for Polder Valley, chances of conclusion applying to other organizations are quite small. Furthermore, we can discuss any threats to this internal validity. Starting with the CMM assessment, it can be seen that only a small assessment has been done. A more elaborate assessment would provide better and more clear information regarding the state of the company. So, when the assessment would be more elaborate, the internal validity would be better. Then, there are the methodologies that have been researched. There are many literary definitions of dozens of Agile methodologies which all could not be assessed. A more thorough research taking in more methodology would be more reliable that the one performed. The following threat has been very present in the research. There weren't a lot of subjects to collect data from. This means one person's assessment of aspects (e.g. time costs) ways heavy on the average. Therefore, bias could threaten the reliability. Also, due to changes which effected the process, data sometimes was collected from people who had to refer to a past state. This means it was collected from a secondary source and could be less reliable. In general, this research shows a past state. Furthermore, validation was performed by a single person who could be biased.

Then, there is the creation of the Ideal model. As explained before, the model was created taking into account a single development team and is therefore not likely to be externally valid. Furthermore, the

model has been created from the viewpoint of the researcher who could be biased. As well as the Product Owner, who was consulted for mapping the ideal roles and validating models. This is a threat to reliability, even though it isn't possible to find a real optimal state. Still, this model served as the foundation for interventions. Therefore, interventions could have been created with a certain amount of bias. This doesn't necessarily mean that the interventions lack quality.

Finally, there are two constricting viewpoints in the research. The CMM states that measuring performance is a higher-level characteristic. However, when creating metrics, performance is being measured. However, it is still possible to start doing this and value is still added. Therefore, the choice has been made to pursue this viewpoint among others.

8.3.2 Limitations and further research

Several limitations have been identified during the research. All limitations provide an opportunity for future research which would be of value to the company. First of all, as explained in paragraph 8.3.1 data was often collected indirectly. Surveys were used instead of actual measurements. An opportunity for further research is measure actual time spent on different activities for a more reliable analysis. Also, in that case all activities can be taken into account. In this research (for the As-Is model) several activities weren't taken into account as they were outside of the scope of the research. Second, Polder Valley has committed to Scrum. Therefore, the focus of this research was on Scrum as well. In further research other Agile methodologies could be researched to discover if any other methodology would be applicable for Polder Valley. Third, not all recommend variables which could have an influence on the success of the team have been completely uncovered. There could still be research done regarding (the number) of acceptance criteria and the number of bugs that occur. Fourth, management focuses on time rather than costs. A focus on costs could yield different results when researched. Fifth, the situation couldn't be compared to a comparable development team or organization. It is therefore also a future research opportunity to make a comparison with a (successful) team to create a benchmark. This could be combined with a more thorough research where the To-Be model is actively compared to the business processes at other companies. Perhaps to create a better ideal division of responsibility. Sixth, there wasn't any time to actually design the planning and scheduling tool. When an MMP of the application would be created this would provide an opportunity to collect more useful feedback from potential users compared to the evaluation that has been done. Last, because changes haven't been implemented, a proper evaluation has not been possible. When changes are implemented, research in the consequences of this implementation would provide useful insights that could lead to process or product improvements.

8.4 Contributions for Polder Valley

Via this research, no significant contributions to literature have been made. However, several contributions have been made to the planning process of Polder Valley. First of all, the As-Is and Ideal model have been created. These models provide Polder Valley with a detailed insight and analysis regarding the current situation and the state that would ideally be reached. Both models focus on steps, roles, time costs and planning and scheduling variables. Second, interventions and re-design of the As-Is model via the Ideal model enabled us to create the To-Be model. Together with the interventions the To-Be model can directly be applied and used to assure a similar view of the planning process within the organization. Third, the conceptual planning tool provides a good basis for the creation of an actual tool as well as the general use of variables. Fourth, recommendations often focus on creating a united company with a clearly defined planning process. Which is the research objective. Therefore, the research objective has been met.

8.5 Reflection

First of all, it's important to say that Polder Valley is doing a great job as an organization. Of course, as a young organization there are many problems. However, due to the Agile mindset that Polder Valley has, these aren't seen as roadblocks, but rather as opportunities for improvement. This attitude has ensured that, during my time at the company, the organization has developed rapidly. This made it very difficult for myself to find obvious improvements that could be made and forced me to dig very deep to find the smallest possible improvements. I do still feel, that the 100% involvement of students in the Scrum process provides an amazing learning opportunity for students. But, this also brings complications to the process. When in the future there are more fulltime developers at the company, the current team structure should be evaluated.

Personally, I had an amazing learning experience at Polder Valley. I arrived at the company with an interest in the world of IT. And I learned a lot about it! At the beginning of my time at the company it was quiet in the office, and sometimes I felt a little bit lost. But the more time passed, the more I felt at home at the company. I feel like I've grown closer to people from Polder Valley, the Backbone, and even ExplainiT and IT2IT. I'm also grateful for the opportunities provided to me. I've had many interesting talks where I enjoyed keeping people from working. I definitely want to thank everybody for these conversations which were sometimes even useful for the research. Furthermore, I had the opportunity to discuss Business Models of Invinitiv with the CEO, facilitate a retrospective session for the development team and towards the end aid in improving the Backbone's support team's working process together with the director, which we will continue in the upcoming year. In enjoyed all these (side-)activities which made me feel appreciated and welcome. In summary, my time at Polder Valley was informative, challenging and fun!

References

Agile Alliance. (2018, April 11). *Agile 101*. Retrieved from Agile Alliance: https://www.agilealliance.org/agile101/

AltexSoft. (2017, September 5). *Agile Software Development Metrics and KPIs that Help Optimize Product Delivery*. Retrieved from AltexSoft software r&d engineering: https://www.altexsoft.com/blog/business/agile-software-development-metrics-and-kpis-that-help-optimize-product-delivery/

Arkilic, G., Reijers, H. A., & Goverde, R. H. (2012). How Good Is and AS-IS Model Really? *Business Process Management Workshops*, 12.

BiZZdesign. (2018, May 30). *Enterprise Studio*. Retrieved from BiZZdesign: https://bizzdesign.com/products/enterprise-studio/

Cabanilas, C., Resinas, M., & Ruiz-Cortés, A. (2011). Mixing RASCI Matrices and BPMN Together. *VII Jornadas en Ciencia e Ingeniería de Servicios (JCIS'11)*, (p. 14).

Cohn, M. (2006). Agile estimating and planning. In M. Cohn, *Agile estimating and planning* (p. 308).

Cooper, D. R., & Schindler, P. S. (2014). Business Research Methods. In D. R. Cooper, & P. S. Schindler, *Business Research Methods* (p. 692). New York: McGraw-Hill/Irwin.

Heerkans, H., & van Winden, A. (2012). Geen Probleem. In H. Heerkans, & A. van Winden, *Geen Probleem* (p. 144).

Johnson, J., & Henderson, A. (2012). Conceptual Models: Core to Good Design. In J. Johnson, & A. Henderson, *Conceptual Models: Core to Good Design* (p. 96).

Khan, P. M., & Quraishi, K. A. (2014). Impact of RACI on Delivery & Outcome of Software Developmen Projects. *Fourth International Conference on Advanced Computing & Communication Technologies*, (p. 9).

Krush, A. (2017, November 14). *Agile Framework Comparison: Scrum vs Kanban vs Lean vs XP*. Retrieved from Objectstyle: https://www.objectstyle.com/agile/agile-scrum-kanban-lean-xp-comparison

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies. *Information and Software Technology*, 143-163.

Management Mania. (2016, March 3). *RASCI Responsibility Matrix*. Retrieved from Mangement Mania: https://managementmania.com/en/rasci-responsibility-matrix

Meyer, B. (2014). Agile! The Good, the Hype and the Ugly. In B. Meyer, *Agile! The Good, the Hype and the Ugly* (p. 170).

Microsoft. (2018, April 11). *Visual Studio Team Services*. Retrieved from Visual Studio: https://www.visualstudio.com/team-services/

Morgan, R. (2008). *How to Do RACI Charting and Analysis: A Practical Guide.*

Mountain Goat Software. (2018, April 11). *Daily Scrum Meeting*. Retrieved from Mountain Goat Software: https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum

Mountain Goat Software. (2018, June 25). *User Stories*. Retrieved from Mountain Goat Software: https://www.mountaingoatsoftware.com/agile/user-stories

Object Management Group. (2009). *Business Process Model and Notation (BPMN) version 1.2.*

Object Management Group. (2011). *Business Process Model and Notation (BPMN) version 2.0.*

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). *Capability Maturity Model for Software, Version 1.1.* Pitssburgh: Software Engineering Institute Carnegie Mellon University.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information System Research. *Management Information Systems*, 53.

Pichler, R. (2013, October 9). *The minimum viable product and the minimum marketable product*. Retrieved from romanpichler: https://www.romanpichler.com/blog/minimum-viable-product-and-minimal-marketable-product/

Planview. (2018, April 11). *What is a Kanban board?* Retrieved from Leankit: https://leankit.com/learn/kanban/kanban-board/

Radigan, D. (2018, June 30). *Five agile metrics you won't hate*. Retrieved from Atlassian Agile Coach: https://www.atlassian.com/agile/project-management/metrics

Rasmusson, J. (2018, June 25). *Burndown Charts*. Retrieved from Agile In A Nutshell: http://www.agilenutshell.com/burndown

Rouhani, B. D., Mahrin, M. N., Nikpay, F., Ahmad, R. B., & Nikfard, P. (2015). A systematic literature review on Enterprise Architecture Implementation Methodologies. *Information and Software Technology*, 20.

Schwaber, K., & Sutherland, J. (2018, June 25). *The Scrum Guide*. Retrieved from Scrum Guides: http://www.scrumguides.org/scrum-guide.html

Skrabanek, B. (2017, August 30). *Difference Between Vision and Mission Statements: 25 Examples*. Retrieved from ClearVoice: https://www.clearvoice.com/blog/difference-between-mission-vision-statement-examples/

Surbhi, S. (2017, March 10). *Difference Between Internal and External Validity*. Retrieved from Key Differences: https://keydifferences.com/difference-between-internal-and-external-validity.html

Taghizadegan, S. (2006). Essentials of Lean Six Sigma. In S. Taghizadegan, *Essentials of Lean Six Sigma* (p. 275).

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. 55.

VersionOne. (2018, June 22). *The Ultimate Guide to DevOps in 8 Foundational Concepts*. Retrieved from VersionOne: https://www.versionone.com/devops-101/what-is-devops/

Weske, M. (2012). Business Process Management. In M. Weske, *Business Process Management* (p. 403).

Wester, J. (2016, September 25). *Flow Efficiency: A great metric you probably aren't using*. Retrieved from EVERYDAY KANBAN: http://www.everydaykanban.com/2016/09/25/flow-efficiency/

Tables and figures

9 Appendix

9.1 Resources

Polder Valley, and I, make use of several resources. Of course, I'm provided with a space to work. But, for a software development company the most important resources are people. An overview of the Polder Valley team can be seen in the table below.

Table 28 Polder Valley team

Part	Focus	Approximate hours/week
Nico Kienhuis & Peter Klijndijk	Business development	16
Gert Kienhuis	Lead developer, Product Owner	16
Edwin Tangenberg	Software development, Scrum Master	36
Wim Holterman	Software development	32
5 part time developers	Software development	40
1 part time developer	Graphic design	4
1 marketer	Marketing	8

If I'm able to convince my company supervisors of the necessity, I can make use of the people working at Polder Valley as much as I wish. The developers use various applications and tools with respect to a task they are working on. I will only explain tools provided to me by Polder Valley or third companies that I will make us of.

Microsoft Office:

- Standard range of applications (e.g. **Word, PowerPoint, Excel**).
- **Visio** is an application or making technical and logical schemes.
- **Teams** is an adaptable chat environment for teams which is used for communication.
- **Power BI** enables the user to create interactive, dynamic and interesting dashboards from that data that can be shared within the company. An example of a Power BI interface can be seen below.

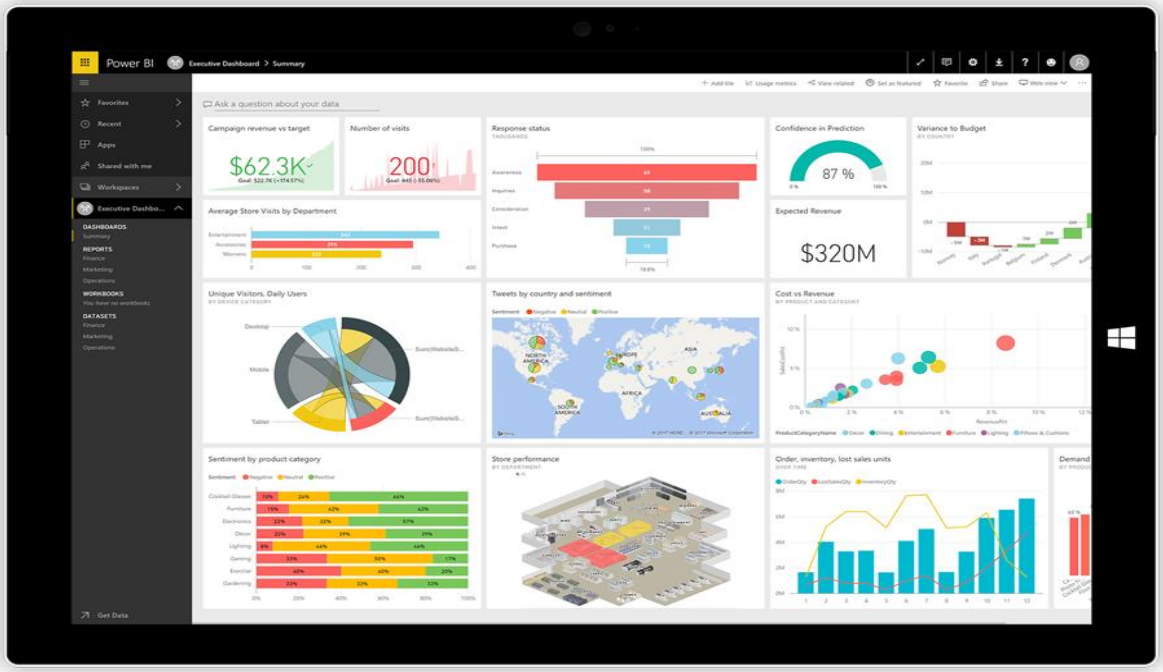


Figure 39 MS PowerBI example - https://powerbi.microsoft.com/en-us/features/

Furthermore, I have access to Microsoft Visual Studio Team Services (VSTS). VSTS is the coding, planning, overview environment used at PV. This environment contains Kanban boards containing tasks and backlog items. This is also the place where code branches can be merged, and code can be tested (Microsoft, 2018). VSTS has a very present role in the planning process and can be used a source of data for my research. An example interface of VSTS can be seen in the figure below.

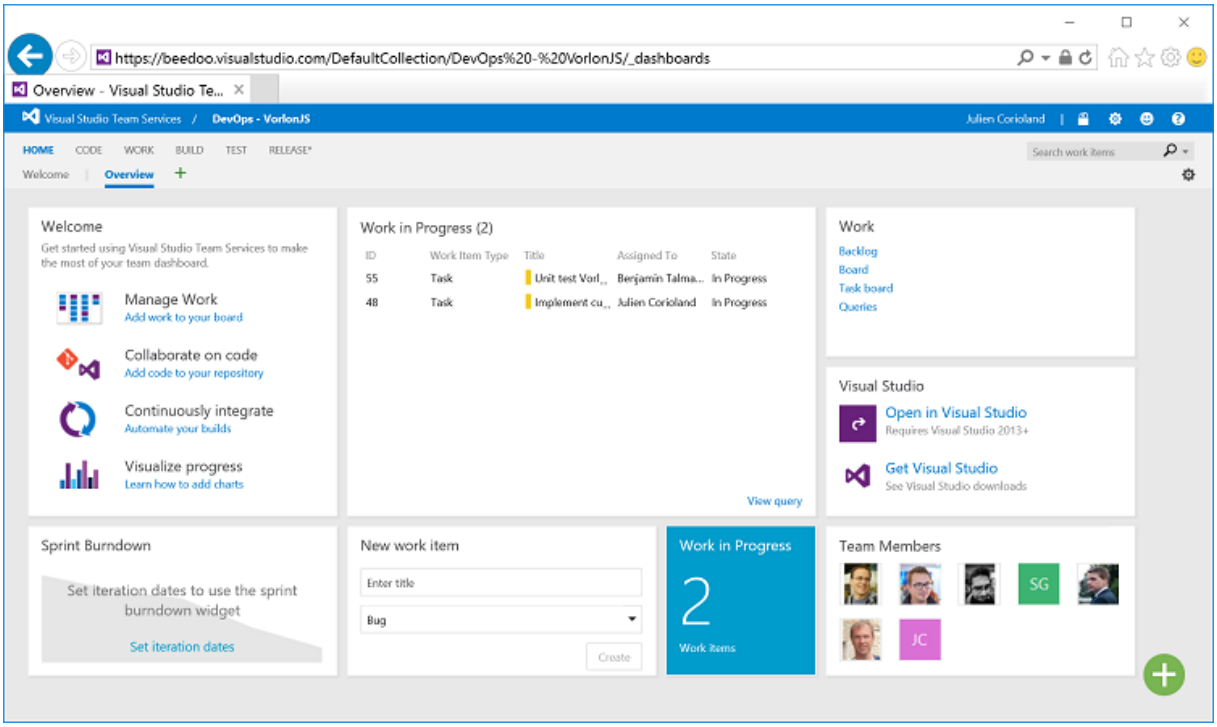


Figure 40 Visual Studio Team Services example - http://www.visualstudioresources.com/overview/visual-studio-team-services/

Besides all these Microsoft applications, I have also been provided a license from BiZZdesign. Via my UT supervisor Adina Aldea, I can make use of BiZZdesign Enterprise Studio. Which is a “collaborative business design platform that offers powerful, integrated modeling across multiple disciplines. It provides all the capabilities needed to seamlessly plan, track and execute change in a single software platform (BiZZdesign, 2018)”.

Last, I have access to worked hours which employees have declared via the Time Writer application. To access I have to approach Peter Klijndijk.

9.2 Stakeholder analysis

There are several stakeholders which I should take into account at my company and one from the University of Twente. These are mostly people with far more experience than me. Therefore, I should respect their input and try to use it in a correct way. I consciously don't consider myself into as a stakeholder. Because, personally I want the best result for all stakeholders. That would please me the most. An overview of my stakeholders can be found in the table below.

Table 29 Stakeholders

Group	Name	Position	Stake/role regarding PV
Business development	Nico Kienhuis	CEO Invinitiv and Business Developer PV, also my company supervisor	Officially the only stakeholder of PV and as 0,2 fte involved with the development of PV
Business development	Peter Klijndijk	Director the Backbone and Business Developer PV, also my company supervisor	Representative of The Backbone and 0,2 fte involved with the development of PV
Business development	Luuk IJland	Director operations ExplainiT	Representative of ExplainiT
Development team	Gert Kienhuis	Lead Developer PV and Product Owner	0,5 fte leading the development team of PV
Development team	Edwin Tangenberg	Fulltime developer and Scrum Master	As a Scrum Master he has an explicit role in the tea
Development team	Fulltime developers	Besides Edwin Tangenberg also Wim Holterman	Are always present at the office and working
Development team	Part time developers	The students which are a part of the development team	Working an average of 8 hours a week and often from a remote location
University of Twente	Adina Aldea	My supervisor from the University of Twente	None

My stakeholders contain three distinctive groups which I will assess separately. There are Business developers, the development team and the University of Twente. The stakeholders are also addressed in several moral issues later in this chapter

The Business developers are also my company supervisors. They saw the need for some changes within the company. I believe I have to take their opinions into account. But, they are mostly focused on end-results. It doesn't matter very much to them how I do my work. They just want me to add value to the development of Polder Valley as a company. I've already noticed that they appreciate it when I disagree with one of their opinions as they like to engage in a discussion. Therefore, I think that when I make well-made decisions, this group will agree with them.

The development team is the main subject of my research. I aim to change the way they work (for the better). Therefore, I can have a significant influence on the team as a whole, or groups within the team. The team is mostly focused on changes with a direct positive influence on the development process. It's mostly Gert who collects input and makes decisions regarding changes in the way the team works. If I were to suggest significant changes I will have to obtain buy-in within the team. Otherwise I don't believe a change will have a positive effect. As explained before, it's of the highest importance that Gert will support any change.

Finally, there is Adina Aldea, my University supervisor. My experiences with Adina so far have shown me that her advice is all to help me obtain a higher grade. She doesn't question any content but merely steers me in a good direction. Therefore, I should accept all the advice she gives me. Because, a high grade is the end-goal for me. As a side note, that doesn't mean I shouldn't engage in any discussion!

9.3 Moral issues

All these cases could possibly occur during the execution of my Bachelor assignment. In my different cases, active, as well as passive, responsibility play an important role. Where active responsibility is taking responsibility over something before it happened, passive responsibility is taken after something has happened. In all cases I'm actively responsible for the avoiding of bad outcomes. This means I will always try my hardest to ensure a good outcome for everybody. If something happens due to a personal decision or action I will be passively responsible.

9.3.1 Which of my stakeholders should I listen to?

Having conversations with people working at the company I've already noticed that there are very different views. For example, the owner of the company has a clear vision and wants to ensure the right product is produced. Whereas the product owner focuses on making the product in a correct fashion and the director of course is aversive against high costs. And adding the development team and my University supervisor makes a lot of different voices to listen to. And perhaps I would have to disappoint some when favoring others.

In this case I must consider many actors. And choose from whom I will take an opinion into account, and who's opinion to discard. It is possible for me to make the wrong decision based on the information I have at a certain point in time. Then, I will have to take passive responsibility for a possible negative outcome when this has been the case. I presume most people will understand why I made a certain decision at the time.

9.3.2 The University versus my company

It's somewhat predictable there is a discrepancy between wishes of the University and the company I'm working for. It's true that certain aspects of M11 and 12 add more to my personal development than to a good recommendation regarding the problem of the company. I will have to find a good balance between these two stakeholders.

I see this case as one of the most difficult conflicts. I have chosen to do my thesis at my company because I believe I gain useful experience. And I want to provide them with the best recommendations possible. However, the University (my supervisor) will be the one grading me. Therefore, I will have to ensure that the University is content with the work I'm doing. This requires me being actively responsible when it comes to tempering my enthusiasm regarding doing the best job possible for my company.

9.3.3 Making a recommendation that directly influences a person (negatively)

It's a possibility I will conclude that somebody (or multiple) people aren't functioning in the best way for the company. Or perhaps even that a person should be let go. As it would be possible that a decision I make could have significant (negative) impact on the life of a person raises difficult questions. Should I blindly go for the maximum result solving the problem of the company? Or is it unacceptable to leave a person out of consideration?

In this case I consider myself to be actively, as well as passively, responsible. Because I think it's important to never forget that there are actual people working. I believe to consider human welfare in all cases (also for your own employees). I would wish to safeguard this by always looking for a solution that does not negatively influence a person. However, this could be unavoidable. And it could even be so that the person in question has not taken responsibility in some way for his own actions. Then, I would not find myself responsible. When I would consider myself responsible is when it would turn out I have set the wrong person accountable for something. Then, passive responsibility would be mine.

9.3.4 Conducting personal business on company time

It can easily happen that I find myself conducting personal business during my time at the company. For example, I do some work on the side for the Student Union and am organizing a trip for my fraternity. Of course, I'm supposed to spend the time at the company working for the company. So, there shouldn't be too much time spent on 'side activities'. But what is too much?

I believe this is something everybody does to some extent. It could also be very personal to what extent somebody things this is acceptable. Regarding effectiveness and efficiency, I hold myself actively responsible for ensuring this does not negatively influence my work for the company. Even though I'm not able to pinpoint exactly when this would be the case.

9.4 Conclusions from first interviews

Table 30 Conclusions from first interviews

Interview matrix	**Conclusion**
Way of working	
Choice for Agile	Everybody agrees that Agile is a good fit. There are several arguments given supporting this.
Other ways of working	The traditional method (in combination with Agile) still appears to be used. This is most likely still because everybody has to get used to the new situation. Because of this, sometimes, unnecessary work is done.
Sprint length	There have only been two sprints so far. The first sprint was extended by a few weeks. The current sprint is going to be extended by one week. So, at this moment there is still too much loss of time. This leads to planned items that are not being done in the desired time. People also mention this is because of the unpredictable resources.
01-07-2018 for the MMP	This is a wish. Nico and Gert are going to discuss whether this is possible.
Daily business	
Planning/way of working	At this moment the planning is not being revised. Code review often hasn't been done yet when the end of a sprint is near. There is no clear division of tasks and for people outside of the development team it is unclear what is going on. A product demo after every sprint could solve these problems.
Meetings	For the University students it's hard to be physically present. In the past there was a weekly moment where everybody was present, but at this moment this appears to be impossible. There are no meetings regarding the planning or the backlog. Diederik meets with Gert and Daniel every few weeks to discuss the planning. Nico, Peter and Gert discuss everything on a strategic level (Luuk is also a part of this, but not very active at the moment).
Communication in general	At the moment the communication goes via MS Teams. Diederik always check what the University students are doing. Neither Nico or Peter are involved in any direct communication within the team. Nico especially wants to obtain a better overview regarding what's going on. It's clear the communication needs to improve. There is a difference between the view of Gert and Diederik about Diederik's role. Communication isn't flowing, and the SCRUM platform should be used in a better way.
Communication personal	It's unclear who the product owner is, Gert or Nico/Peter/Luuk. There is also a difference between wat Gert and Diederik think about Diederik's role.
Capacity	The capacity of Polder Valley (the development team) has to be known.
Demand	There is a vision. In two year all the products should be self-sustaining (with bigger teams).
Product	
Requirements	Now the focus is only on the systems architecture. There is a wish to sell 100 units of each product.
Involvement stakeholders	After the MMP products can be tested at stakeholders.
Software development	
Planning and prioritizing	At this moment there is no system regarding planning or prioritizing. There is looked at what should be possible during a meeting at the beginning of a sprint. Furthermore, there is no clear overview.
Loss of time and vision	Loss of time is mostly caused by the combination of fulltime and part-time employees.
Waste	Waste arises due to a difference between skills/knowledge and because people create more functionality than is asked for.
Other issues	Communication, planning, difference between employees, nobody has 100% focus, no clear task division, very different sprints, not enough overview for people outside of the development team.
Use of software to communicate	Present.
Collection of data for forecasting purposes	Gert doesn't think PV is mature enough to start measuring the performance of employees. Diederik thinks this would add value, but it has to be communicated in the right way. Peter doesn't think this is necessary, and Nico agrees you shouldn't want to influence everything. Data/KPI's would be happily used to improve process performance.
Restrictions (costs or time)	The University students.

9.5 Twelve principles of Agile

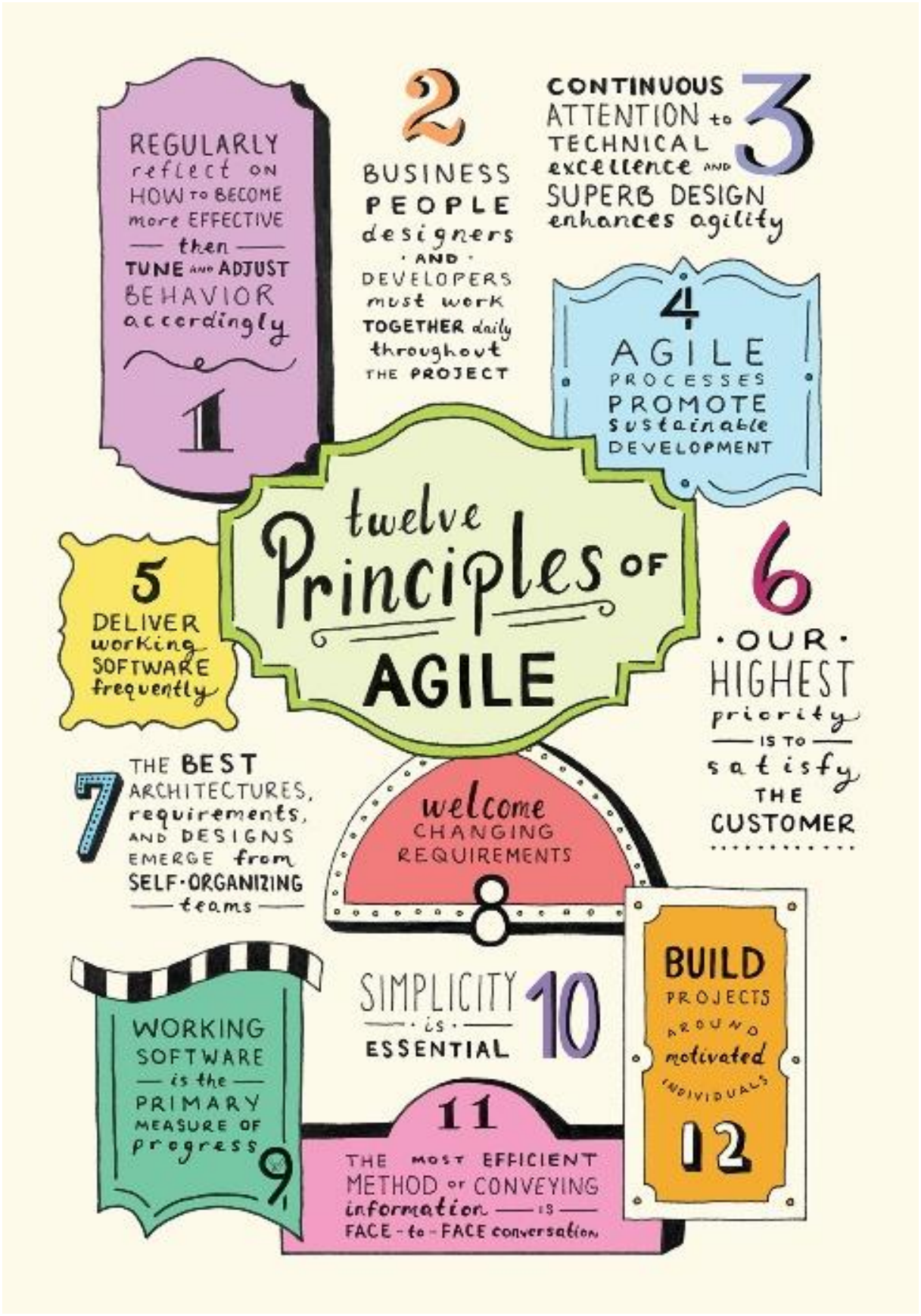


Figure 41 Twelve principles of Agile https://www.behance.net/gallery/28702877/12-Principles-of-Agile-Poster-Walt-Disney-Studios

9.6 Systematic Literature Review

9.6.1 Search string

String 1: "Agile software development" AND ("planning strategy" OR "planning method" OR "planning framework" OR "planning approach" OR "planning policy" OR "planning procedure" OR "planning system"), searched on 17-04-2018

String 2: (TITLE (Agile AND planning)) AND (software) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)), searched on 18-04-2018

9.6.2 Exclusion criteria

Table 31 SLR - exclusion criteria

Criteria	Reason for exclusion
Pre-2014 articles	As Agile software development is becoming more and more popular it has also become a more popular research topic. To maintain a feasible scope, it helps to include this criterion.
English language	To ensure I can understand the literature and terms are used in the same way English (the most used language)
No mentioning of "planning" in the abstract of a paper. No mentioning of "planning" in an introduction chapter of a book. Planning should be in a business working environment.	Often planning is merely touched upon in literary sources.

9.6.3 Set of literature

Table 32 SLR - set of literature

Action	Entries
Search string 1 Google Scholar	1090
Search string 1 Scopus	10
Search string 1 Web of Science	5
Total	**1105**
2014-present	-702
English language	-21
Duplicates	-3
Total	**379**
>4 citations/year	48
No planning	-29
Inaccessible	-5
Total	**14**
Search string 2 Scopus	23
>4 citations/year	-17
No planning	-4
Search string 2 Web of Science	12
>4 citations/year	-12
Duplicates	-1
Wrong topic (after reading)	-4
Total	**11**

9.6.4 Final set of literature

Table 33 SLR - final set of literature

Number	Authors	Title	Year
S1	P Serrador, JK Pinto	Does Agile work?—A quantitative analysis of Agile project success	2015
S2	EC Conforto, F Salum, DC Amaral…	Can Agile project management be adopted by industries other than software development?	2014
S3	A Moran	MANAGING AGILE.	2016
S4	A Scheerer, T Hildenbrand…	Coordination in large-scale Agile software development: A multiteam systems perspective	2014
S5	HF Cervone	Improving strategic planning by adapting Agile methods to the planning process	2014
S6	JF Tripp, C Riemenschneider…	Job satisfaction in Agile development teams: Agile development as work redesign	2016
S7	T Suomalainen, R Kuusela…	Continuous planning: an important aspect of Agile and lean development	2015
S8	VT Heikkilä, M Paasivaara, K Rautiainen…	Operational release planning in large-scale Scrum with multiple stakeholders–A longitudinal case study at F-Secure Corporation	2015
S9	BP Douglass	AGILE systems engineering	2015
S10	D Leffingwell	SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering	2016
S11	Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M.	Estimating, planning and managing Agile Web development projects under a value-based perspective	2015

9.6.5 Conceptual matrix

This conceptual matrix is a summary of a larger conceptual matrix containing all found relevant information from the sources.

Table 34 SLR - conceptual matrix

Number	Source	Authors	Year	Research subjects	Conclusion	Mentioned strategies	Key findings
1	Does Agile work?—A quantitative analysis of Agile project success	P Serrador, JK Pinto	2015	859 participants from several countries with various managerial positions	The level of Agile used in a project does have a statistically significant impact on all three dimensions of project success, as judged by efficiency, stakeholder satisfaction, and perception of overall project performance	Traditional planning, iterative methodologies, upfront planning, planning games, replanning	Agile as an iterative methodology has a positive impact to project success relative to traditional planning methods. Also, it is explained that upfront planning is required, and planning games are explained
2	Can Agile project management be adopted by industries other than software development?	EC Conforto, F Salum, DC Amaral…	2014	19 medium and large-sized companies from different industry sectors considering innovative projects	The companies surveyed have some characteristics and organizational enablers similar to companies from the software industry, which is considered a source of motivation to develop and pursue the application of Agile management practices	Traditional planning, iterative methodologies, project planning responsibility, intermediate approach	It shouldn't necessary be a manager who is responsible for planning. There can also be shared responsibility within a team. This can be done in an intermediate approach which is positioned between the traditional and Agile approach
3	MANAGING AGILE.	A Moran	2016	na	na	Traditional planning, iterative methodologies, upfront planning, planning games, adaptive planning, customer engagement, Extreme Programming (XP), Dynamic Systems Development Methods (DSDM)/Timeboxing, Scaled Agile Framework (SAFe), Forecasting based planning, multi-tiered planning, MoSCoW prioritization, Configuration Management Planning, Increment planning	Planning poker is introduced as an example of a planning game. Furthermore, many planning methods are explained
4	Coordination in large-scale Agile software development: A multiteam systems perspective	A Scheerer, T Hildenbrand…	2014	A large enterprise software development organization	Coordination strategies lie on a continuum between organic and mechanistic coordination types. If the communication network is completely interconnected, dividing into individual teams is ineffective. A purely mechanistic strategy contradicts the lean and Agile principled of empowered teams and embracing change	Upfront planning, planning games, organic planning, communication focused, strategic planning	The explanation of the continuum between organic and mechanistic planning as well as communication focused and strategic planning

5	Improving strategic planning by adapting Agile methods to the planning process	HF Cervone	2014	na	In order to adapt Agile planning methods, one should allow for gradual change, facilitate the adoption, obtain frequent feedback, gain trust by showing value and track progress using tools and methods	Traditional planning, iterative methodologies, Scrum model	That the use of tools and methods are important as usually Agile focuses more on communication
6	Job satisfaction in Agile development teams: Agile development as work redesign	JF Tripp, C Riemenschneider…	2016	252 software-development professionals	Agile software development and planning management practices improve job satisfaction	Traditional planning, iterative methodologies	Agile software development and planning management practices improve job satisfaction
7	Continuous planning: an important aspect of Agile and lean development	T Suomalainen, R Kuusela…	2015	Three large Finnish-based ICT companies with more than 1,000 employees	The research findings highlight the importance of continuous planning throughout an entire organization including the elements of continuous planning (organizational planning, strategic planning and business planning) and their tight interrelation	Traditional planning, strategic planning, continuous planning, organizational planning, roadmapping, business planning	How even with continuous planning there are different planning levels such as organizational planning and business planning. Also, that a roadmap should be a living document
8	Operational release planning in large-scale Scrum with multiple stakeholders–A longitudinal case study at F-Secure Corporation	VT Heikkilä, M Paasivaara, K Rautiainen…	2015	A large Finnish software company	We identified the following ways the method ameliorates the difficult characteristics of the release planning problem: the communication between the development organization and the Product Management enabled by the events allows both of them to better understand the requirements from the business and the technical points of view	Traditional planning, iterative methodologies, release iteration planning method, model-based release planning	With planning for a release there has to be an understanding between the development team and management. This is actually a prioritization problem which can possibly be solved by a model
9	AGILE systems engineering	BP Douglass	2015	na	na	Traditional planning, iterative methodologies, planning games, model-based release planning	None
10	SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering	D Leffingwell	2016	na	na	Iterative methodologies, Scaled Agile Framework (SAFe)	None
11	Estimating, planning and managing Agile Web development projects under a value-based perspective	Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M.	2015	A project team of four members (multi-disciplinary)	Using an Agile approach to plan, estimate and manage Web projects is a fit.	Iterative methodologies, planning games, Extreme Programming (XP), continuous planning	Business value isn't considered when playing a planning game like planning poker, this should be addressed in a different way

9.7 Scrum elements explanation

Considering the roles within the Scrum process, Meyer explains them as following (Meyer, 2014):

Product owner	"Concretely, the principal responsibility of the product owner is to define and maintain the product backlog: the list of features" (Meyer, p. 80).
Team	"A self-organizing group of developers and others (such as customer representatives), responsible for the ongoing assignment of development tasks to individual members" (p. 7). This refers to the development team.
Scrum Master	"Agile methods raise frequent problems in their daily application and require enforcement, lest the team stray from the recommended principles" (p. 84). It's the Scrum Masters role to facilitate the dealing with this.

Now we will look into the various moments which are also referred to as ceremonies:

Sprint planning meeting	As is stated in figure 11, the team decides how much work to commit two. The mentioned second part focuses on creating a detailed sprint backlog.
Retrospective	"A sprint retrospective reviews what went well and less well during the latest sprint, with a view to identifying what can be improved for the next one" (Meyer, 2014, p. 99).
Review	"The review meeting mirrors, at the end of a sprint, the planning meeting performed at the beginning. Its purpose is to assess what has actually been done" (Meyer, 2014, p. 99).
Daily Scrum meeting	Its focus is precisely defined: answering the "three questions". What did you do on the previous working day? What will you do today? Any impediments?

Artifacts are also mentioned. Artifacts can be virtual or material. Examples of virtual artifacts are the User Stories or a Burndown chart. An example of a material is a physical task board.

9.8 Visualizing metrics

Looking for a more pragmatic use of these variables I found various applications. The most common tool is the Burndown chart which has been explained before. Looking at the highest rated metric, Velocity, we can find various applications. The most frequently used one through a Velocity chart as shown in the example below.

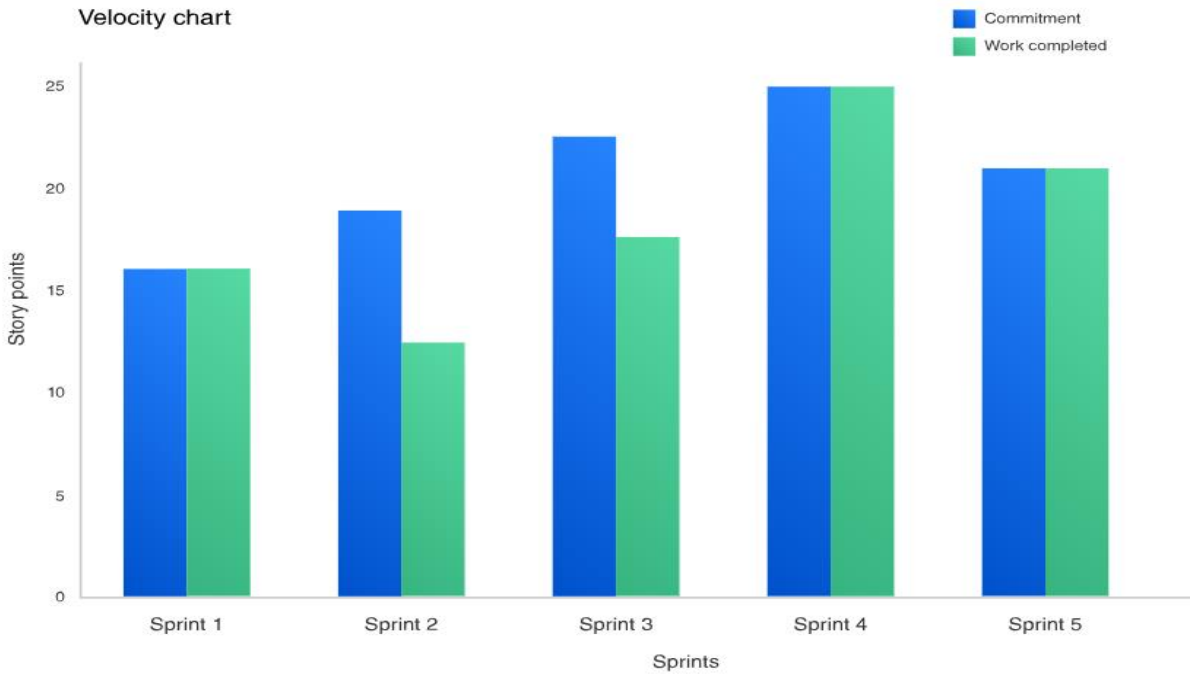


Figure 42 Velocity chart example according to Atlassian - https://www.atlassian.com/agile/project-management/metrics

There is also a more dynamic visualization possible, an Epic and release burndown. “Epic and release (or version) burndown charts track the progress of development over a larger body of work than the sprint burndown, and guide development for both Scrum and kanban teams” (Radigan, 2018).

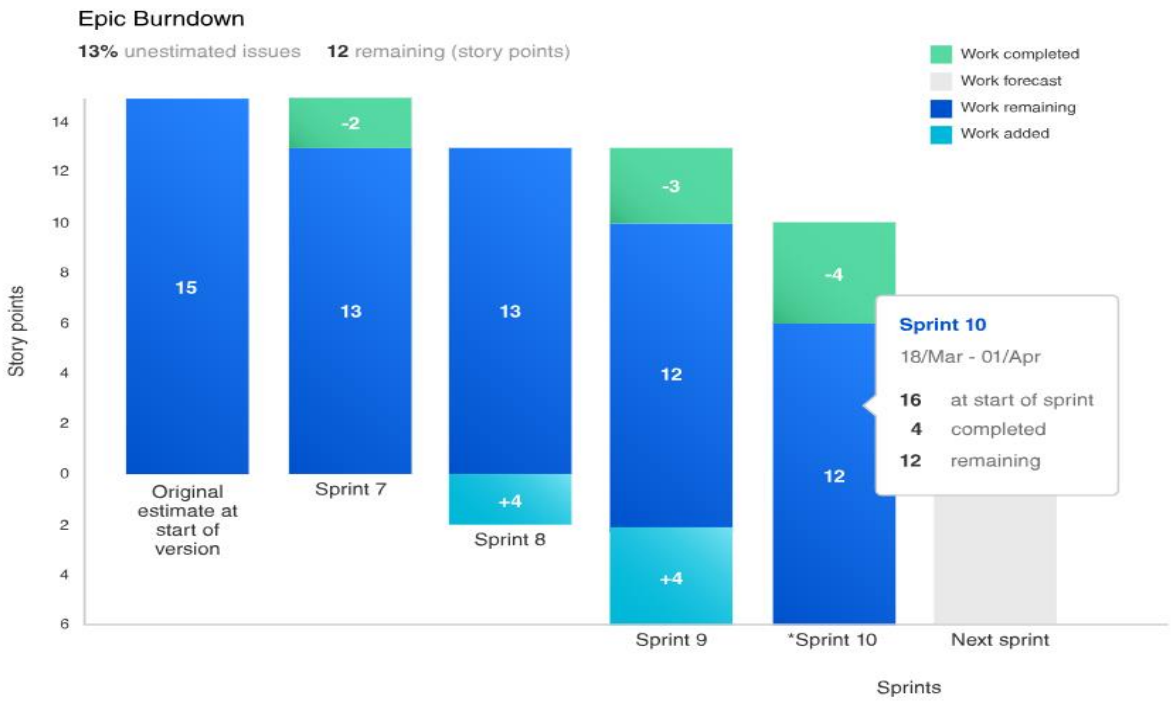


Figure 43 Epic burndown example according to Atlassian - https://www.atlassian.com/agile/project-management/metrics

A decent way to track the build status is through a Cumulative Flow Diagram (CFD). “The cumulative flow metric is described by the chart area showing the number of different types of tasks at each stage of the project with the x-axis indicating the dates and the y-axis showing the number of story points” (AltexSoft, 2017). Story points are the same as feature points.

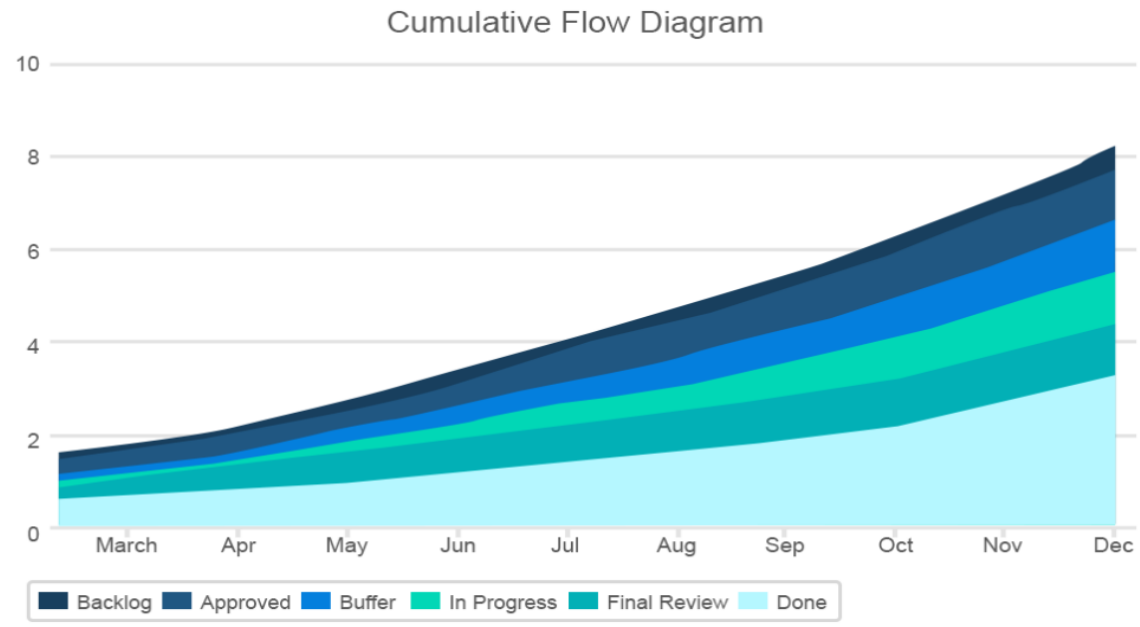


Figure 44 - CFD example according to AltexSoft - https://www.altexsoft.com/blog/business/agile-software-development-metrics-and-kpis-that-help-optimize-product-delivery/

To visualize the lead time is one thing. But, lead time can also be related to the time items spend ‘waiting’. When we are doing this we are assessing the flow efficiency. This appears to be better than

purely taking work-in-progress into account. Because, “work-in-progress isn’t always actually in progress. Flow efficiency tells us how often that is true” (Wester, 2016).

Flow efficiency is measured as following: $\frac{Work\ being\ done}{Work\ being\ done + Work\ waiting} \times 100\%$

When using this one should decide what a minimum acceptable measurement is. “Some say that the 15 percent mark is okay for most projects, which basically means that a story point or another item of work waits 85 percent against 15 percent processing time” (AltexSoft, 2017). Lastly, below there’s an example of a Flow efficiency chart.

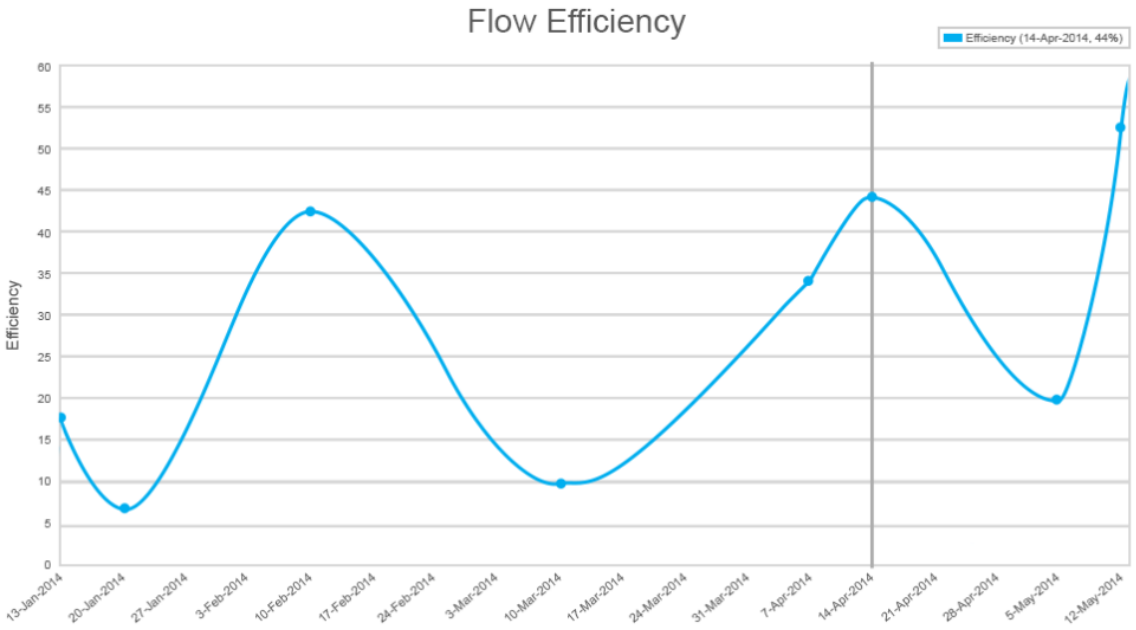


Figure 45 Flow efficiency chart according to Atlassian - https://www.altexsoft.com/blog/business/agile-software-development-metrics-and-kpis-that-help-optimize-product-delivery/

Besides these progress focused metrics there are also several visualizations available regarding code quality or tests run. In chapter 5 I will revisit metrics when deciding which variables should be measurable.

9.9 CMM survey and results

CMM assessment Polder Valley

The following questions should be answered with respect to Polder Valley and the work on the Productivity Performer

*Vereist

Fill in to what extent you agree with the following statements *

	I don't agree	I somewhat agree	I agree	I strongly agree
The product requirements are clear	○	○	○	○
Activities are planned and managed	○	○	○	○
Progress is visible and trackable	○	○	○	○
There is room for skill and knowledge development of individuals	○	○	○	○
New technologies are researched	○	○	○	○
Code is tested appropriately	○	○	○	○
Software integration is managed in a standardized manner	○	○	○	○
Software is produced in a consistent manner	○	○	○	○
Code is reviewed by a different person	○	○	○	○
Processed are clearly defined	○	○	○	○
There is a focus on process improvement	○	○	○	○
Causes of defects are identified to prevent recurrence	○	○	○	○
Processes are continuously improved to improve quality, increase productivity or decrease throughput time	○	○	○	○
Performance is analyzed for the purpose of optimization	○	○	○	○
The quality of work is quantitatively scored	○	○	○	○
Specific and measurable goals are often set	○	○	○	○

Figure 46 CMM assesment 1/2

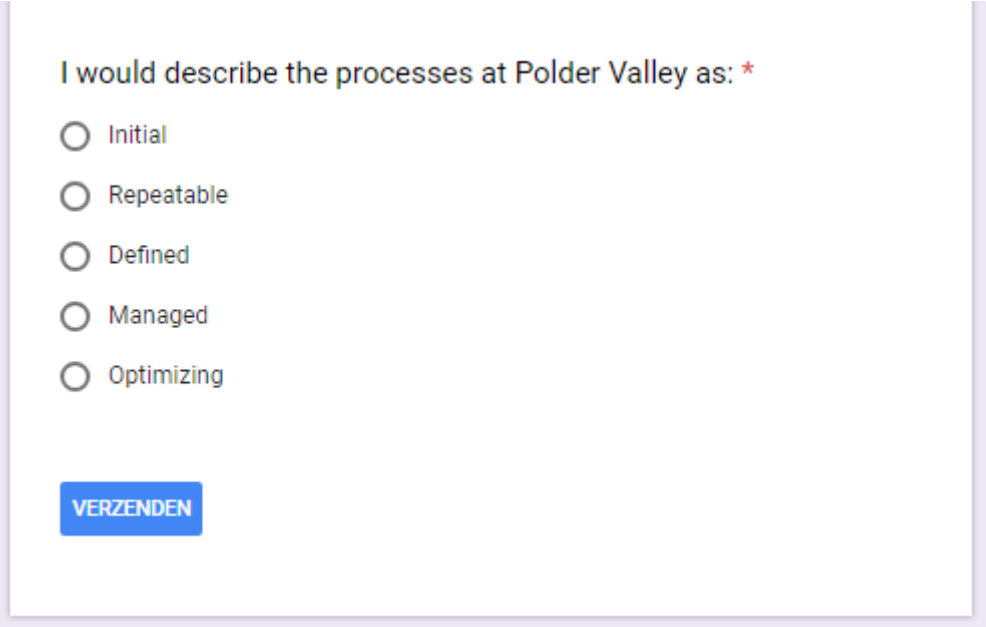


Figure 47 CMM assessment 2/2

One question per subject has been asked. Answers have been operationalized where a score of 1 = I don't agree and 4 = I strongly agree, 0 = unaddressed.

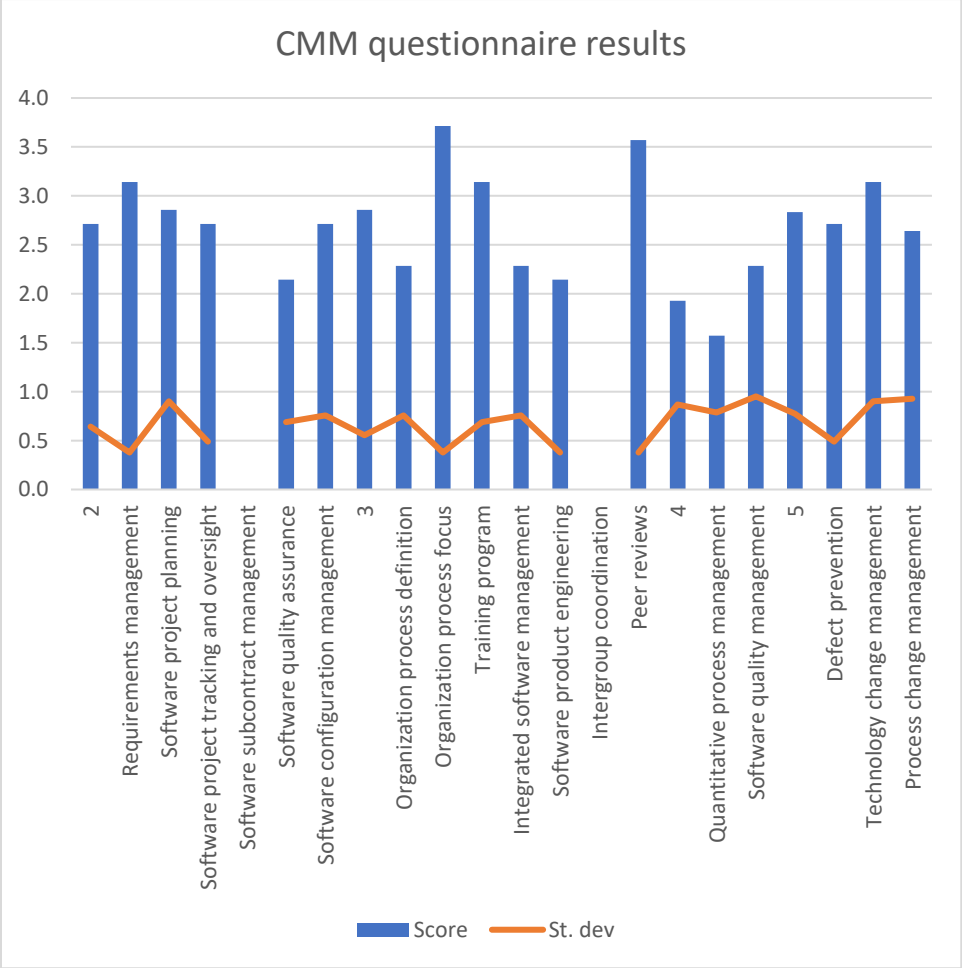


Figure 48 CMM-assessment results 1/2

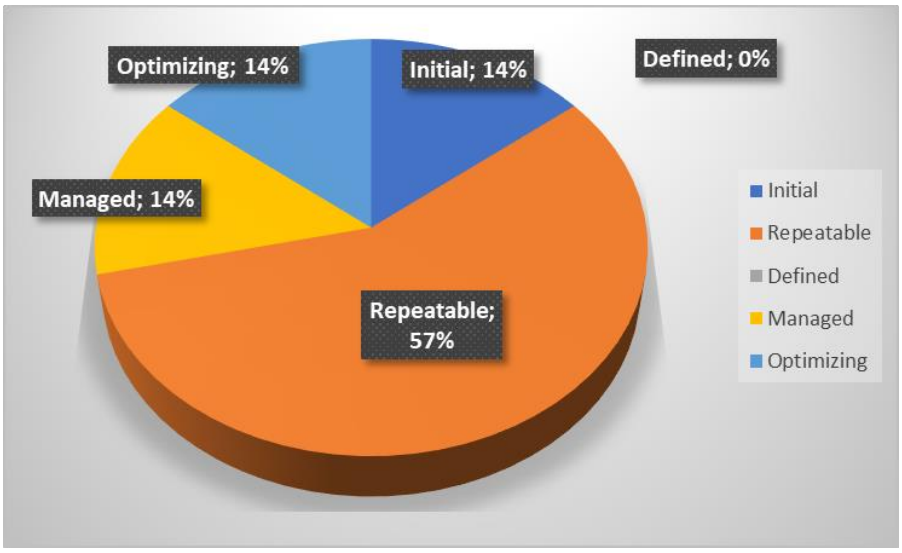


Figure 49 CMM-assessment results 2/2

9.10 Other Agile terms

Table 35 Various Agile terms

Term	Explanation
Epic	An Epic is a large piece of functionality of an application which contains several features.
Feature	A Feature is a smaller piece of functionality which contains several backlog items.
User story	Polder Valley's backlog items are User Stories. "User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system" (Mountain Goat Software, 2018).
Global sprint planning	A global sprint planning is a roadmap of a certain number of sprints which are related. The current global sprint planning contains the path to the MMP.
Product backlog	"The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product" (Schwaber & Sutherland, 2018).
Sprint backlog	"The Sprint Backlog is the set of Product Backlog items selected for the Sprint" (Schwaber & Sutherland, 2018).
Acceptance criteria	Acceptance criteria are criteria that should be met in order to close or finish a backlog item.
Burndown chart	"The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories. It shows the total effort against the amount of work we deliver each iteration" (Rasmusson, 2018).
Effort estimation	An Effort estimation is done in order to measure the time needed for a backlog item. The effort can be 1, 2, 3, 5, 8, 13, 20, 40 or 100. Effort should be benchmarked against items from the past.

9.11 Time spent and value-added survey

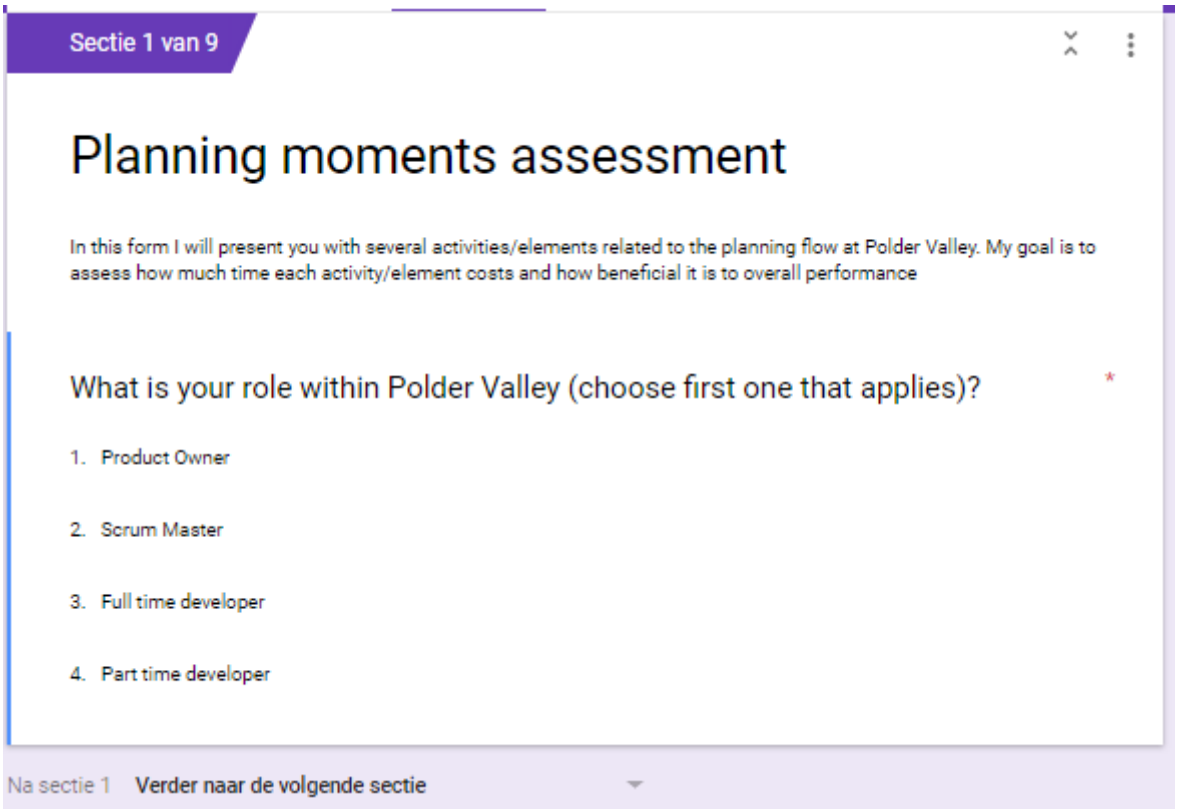


Figure 50 Planning moments assessment 1/9

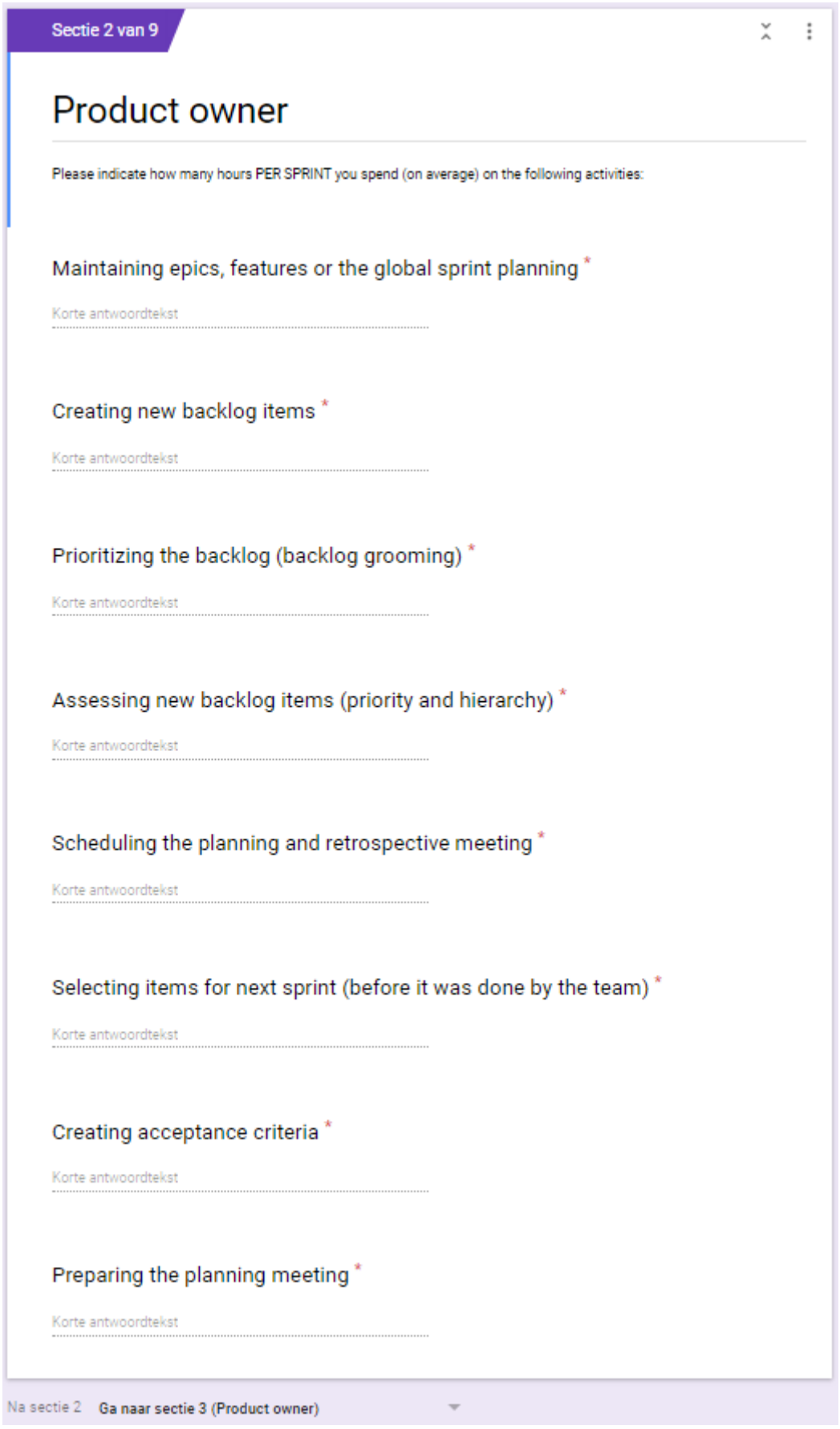


Figure 51 Planning moments assessment 2/9

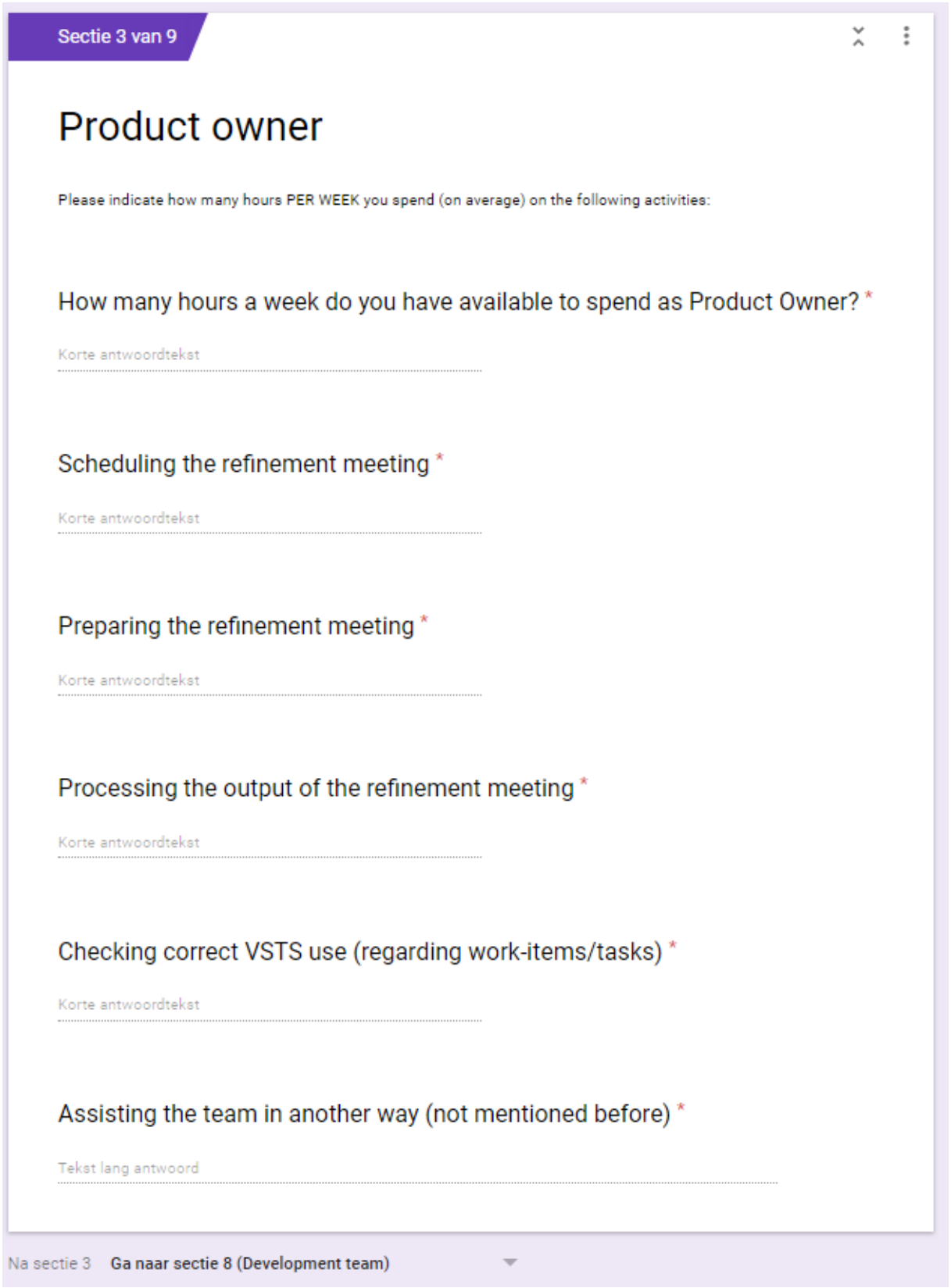


Figure 52 Planning moments assessment 3/9

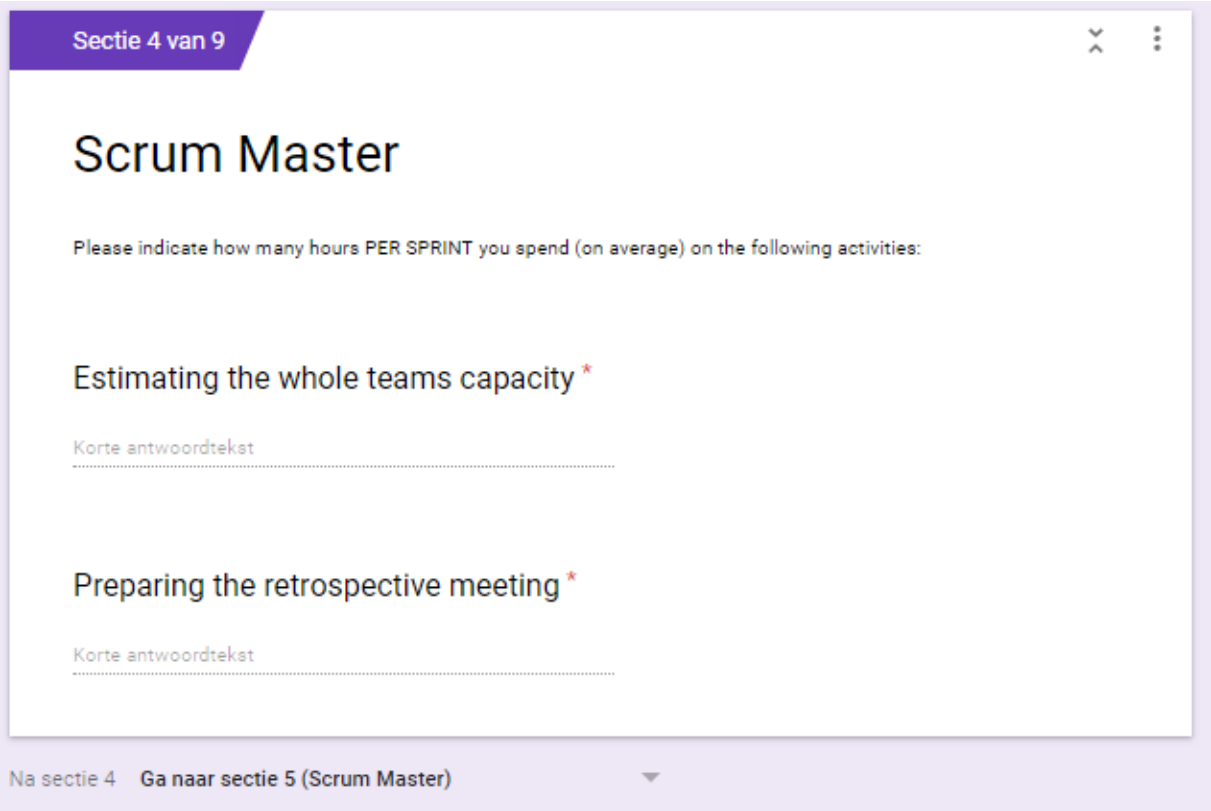


Figure 53 Planning moments assessment 4/9

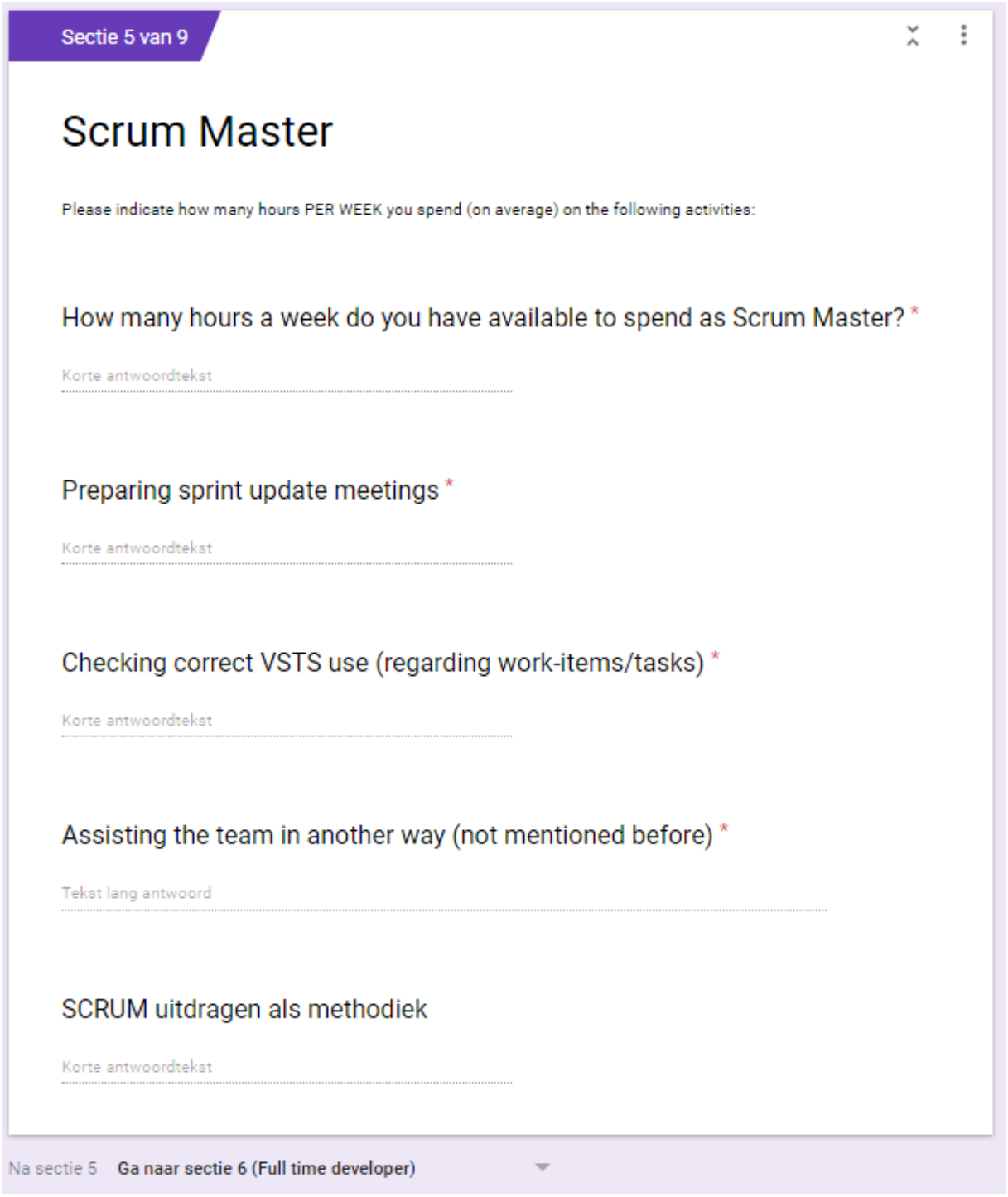


Figure 54 Planning moments assessment 5/9

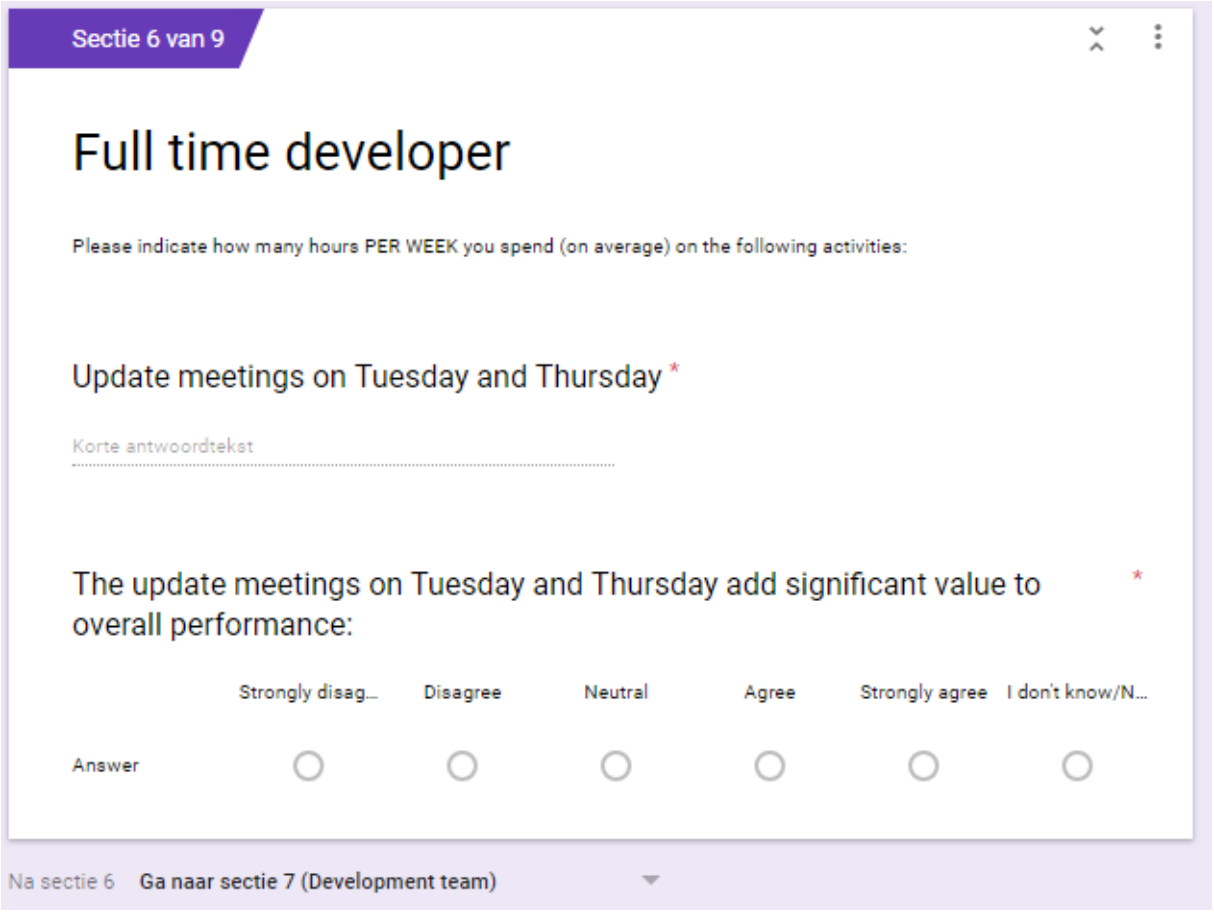


Figure 55 Planning moments assessment 6/9

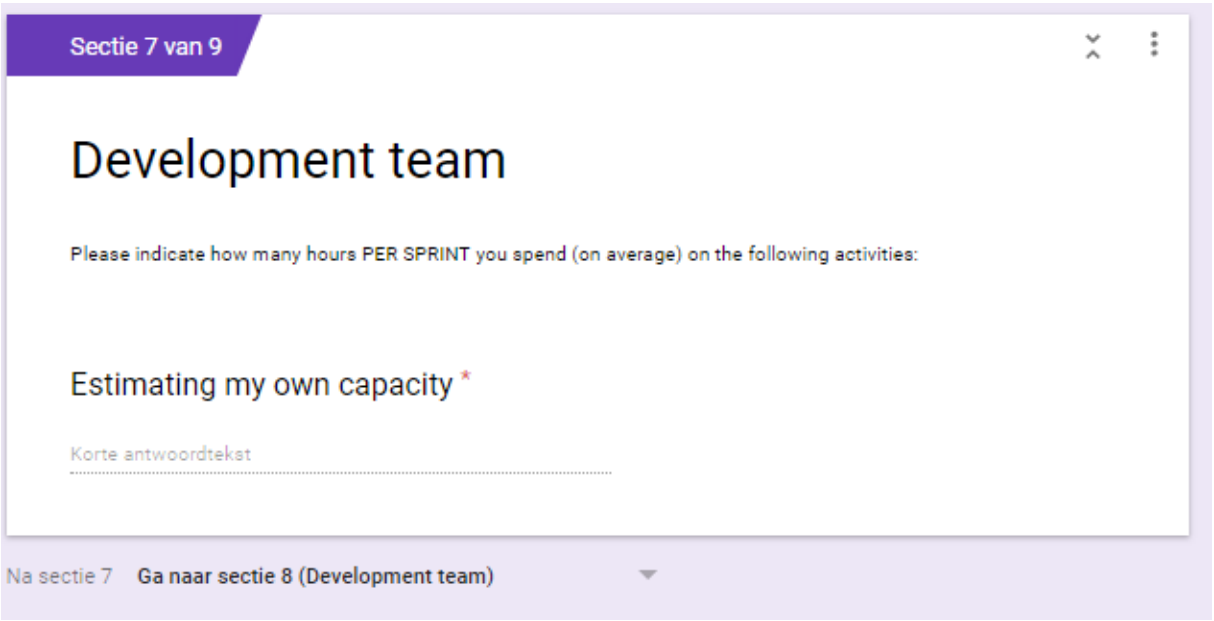


Figure 56 Planning moments assessment 7/9

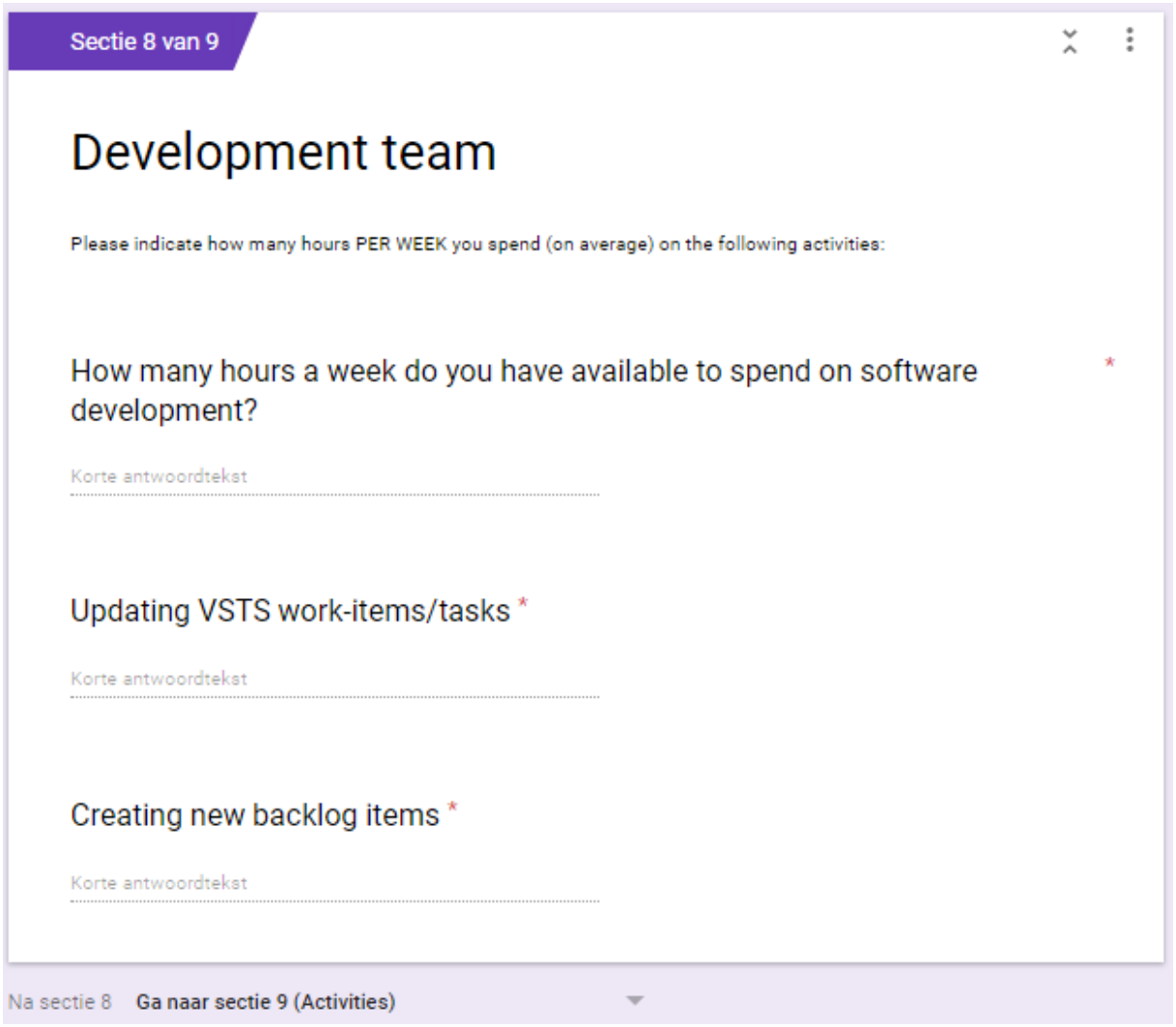


Figure 57 Planning moments assessment 8/9

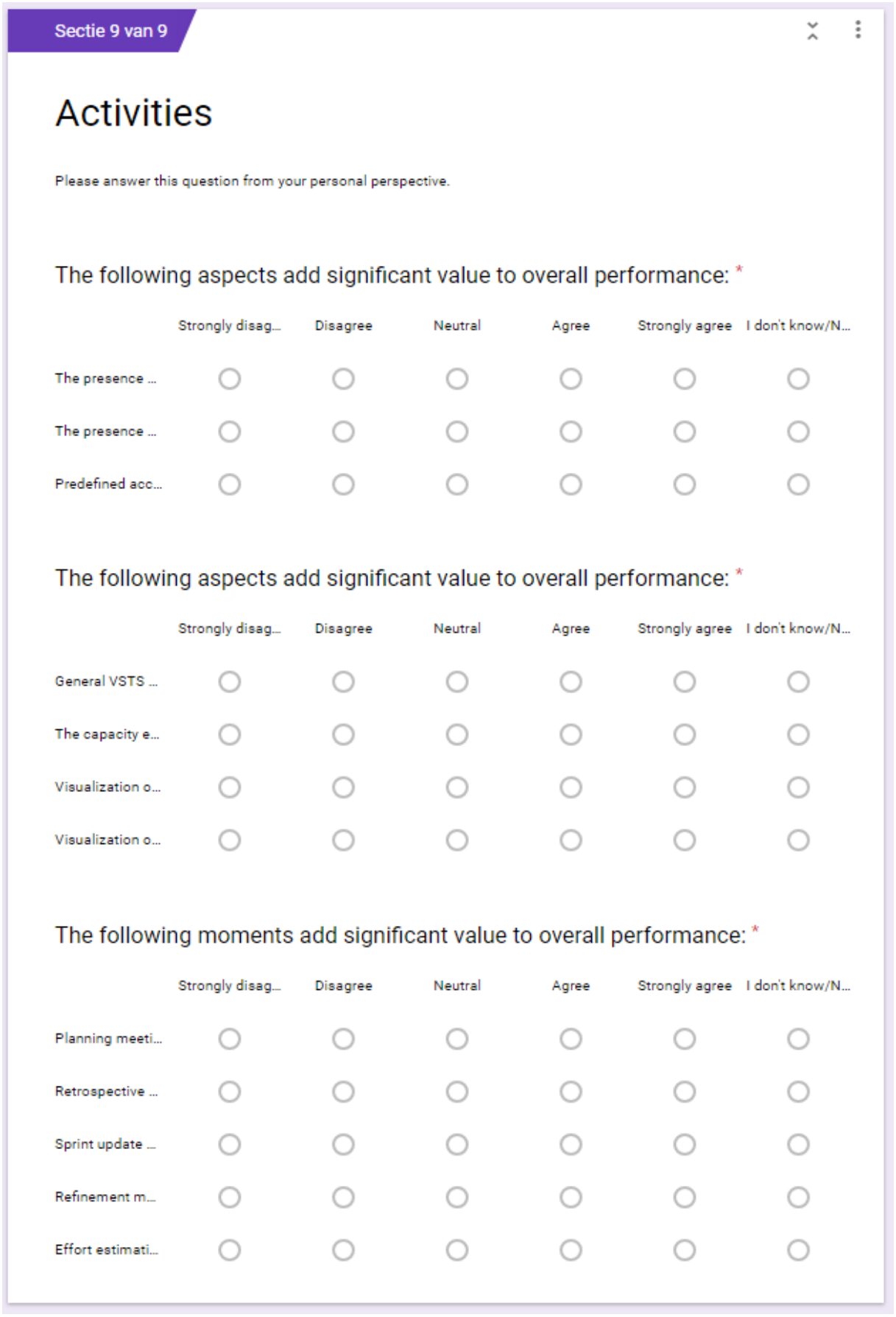


Figure 58 Planning moments assessment 9/9

9.12 Data collection As-Is model

9.12.1 Planning meeting

The planning meeting takes place at the beginning of the sprint. The planning meeting for sprint #5 has been observed. The following was seen in chronological order:

Sprint 4 is assessed shortly, there are still unfinished items even though the sprint was extended with an extra week. There is some discussion. However, it does not become clear what is to happen with these items. The backlog items for the upcoming sprint have been entered in VSTS by the Product Owner. There has already been some preparation, there is an effort estimation. Overhead isn't considered in the capacity estimation. But the team has committed to 80% of total available hours for tasks.

All the items selected for the upcoming sprint are presented by the Product Owner. Sometimes there is a short discussion about an item. Some items have already been assigned to someone. A new item is created during the meeting and added to the backlog. Also, an item is added related to the sprint 4 retrospective.

Then, the development team takes the lead from the Product Owner and items are discussed in more detail. Each item is broken down into tasks which are sometimes assigned to individuals based on:

- Time
- Expertise
- Randomness (seemingly)

Tasks receive an estimated amount of remaining work (in hours). Every backlog item already has somebody who is responsible Finally, the team refers to the burndown chart and concludes 20% might not be enough buffer due to overhead tasks. At this moment it's > 30 %, this is not assessed per person. But, that is seen as a good amount of spare time.

9.12.2 Retrospective meeting

The aim of this meeting is to evaluate the past sprint. This is done through writing things on post-its that should be continued, improved or stopped. Then, there is a vote about the most important items which are then assessed by the team. The meeting is led by the Scrum Master. At the end the team has committed to certain improvements to the way of working. It is less clear what happens with these commitments after the meeting.

9.12.3 Refinement meeting

In the refinement meeting the Product Owner uses input from the development team for upcoming sprints. This is done every week. A few items are selected which are discussed during this meeting. In the end the team should have committed to the items and an effort estimation. When members of the development team have a different effort estimation they can resort to planning poker which works as following:

Everybody has cards with the numbers 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100. There are references available to aid in the estimation. Every member of the team chooses one card and then the entire team turn around their card at the same time. Then, the person with the lowest and highest estimation state their case, facilitating discussion. After this, another round is played, and the process is repeated until consensus is reached.

It is noticeable that it is hard for the students to make time for this meeting every week. A member of the team is going to try to improve this.

9.12.4 Sprint update meeting

This moment is meant to be a daily stand-up. In a stand-up each person answers the following questions:

- What did you do yesterday?
- What are you going to do today?
- Is there anything blocking you?

But to reduce overhead time for the students a sprint update meeting is held every Monday and Wednesday. On days when there is no sprint update meeting, the people present at the office (usually the fulltime employees) do a stand-up. For this meeting most of the students have to join using Skype. Each backlog item which is in progress is shortly discussed. The meetings don't last longer than 15 minutes. There is no clear structure like: Did yesterday, doing today, anything holding me back. In some cases there is background noise coming from the Skyping students.

9.12.5 Interview Product Owner

In order to get a complete overview, an interview as held with the Product Owner, Gert Kienhuis. Gert is involved with, or has an overview on, almost all activities related to planning. He does this even though he is very busy. He only has half of his time to focus on Polder Valley, the other half of his time he is working as a consultant for the Backbone. He is always supportive of the research, but also believes in an organic way of organizational development. Therefore, he sometimes questions new ideas and is, for example, not a fan of measuring performance.

Current planning cycle (MMP 1/7)

There has to be a strategic plan that looks 1 or 2 years ahead, a tactical plan or roadmap that shows the upcoming six months and sprints of 3 weeks. The current roadmap was created by the PO and the CEO. The roadmap is changed continuously, the main aspects do remain the same. Backlog items are created by the development team or out of feedback from stakeholders.

Next planning cycle

The main plan is as following:

1. The focus for 1/7 is on early adaptors (α)
2. The application will be made user ready (β) by the fall
3. The application should be ready for continuous development around May 2019

Now there is a roadmap with 6 months that moves with the time (always a view for the upcoming time). Gert wishes to move to thematical planning cycles containing 5-8 sprints. The PO is always processing all the information from stakeholders and the development team. There is continuous contact within Polder Valley.

Sprints

The challenge for the Product Owner is to always have two sprints prepared. The backlog items are operationalized roadmap features which have been made sprint ready. First these items are added to a sprint, then if there is any room left items with the highest priority are added. This leads to the top backlog items always ending up in the sprint. During sprints there are the following planning moments:

- Planning meeting in the beginning
- Update meetings every Monday and Wednesday
- A refinement meeting every Thursday

- A review (internal) and a demo (for stakeholders) every last Thursday of the sprint
- The first day of the next sprint there is a retrospective

Right now, backlog items are prioritized according to the following:

- Feeling
- Added value to early adapting

In the future the following will be added:

- Business value (defined by stakeholders)
- Effort

9.13 Separate overview of As-Is BPMN model

9.13.1 Strategic and release planning

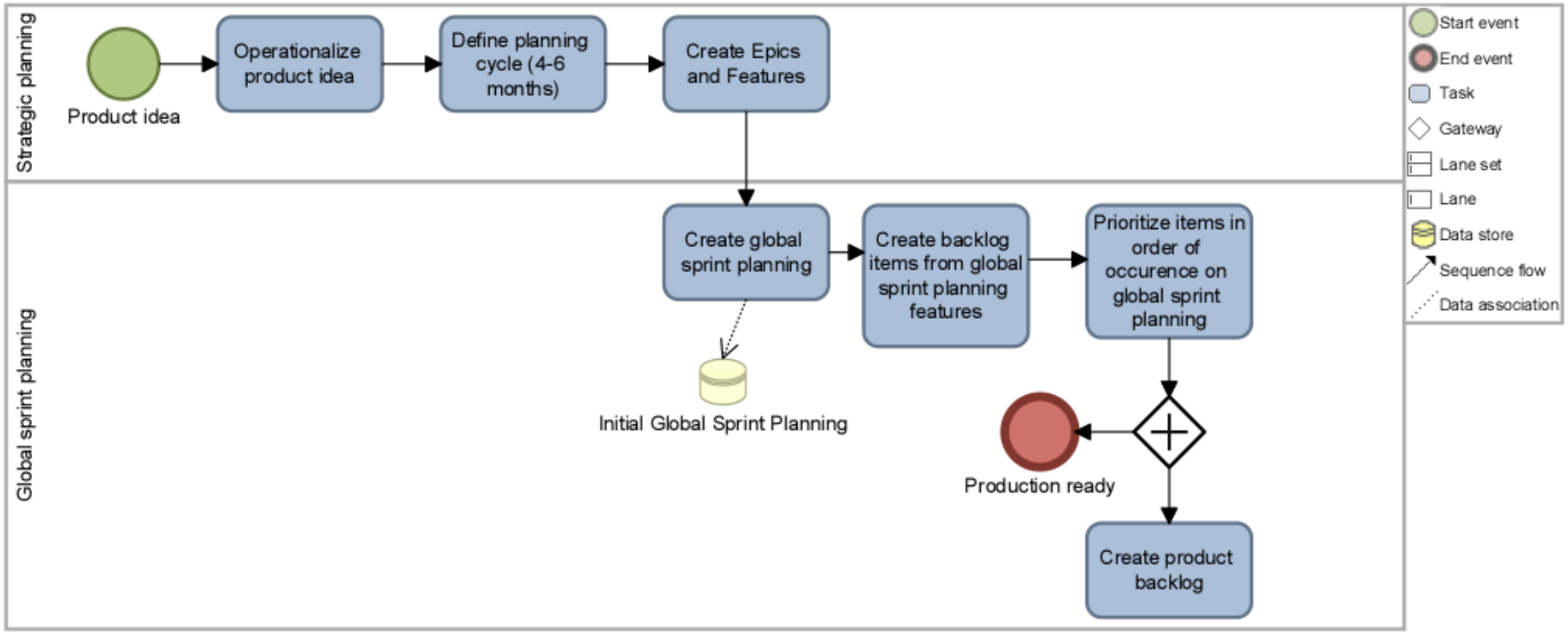


Figure 59 As-Is BPMN Strategic and release planning

9.13.2 Sprint planning

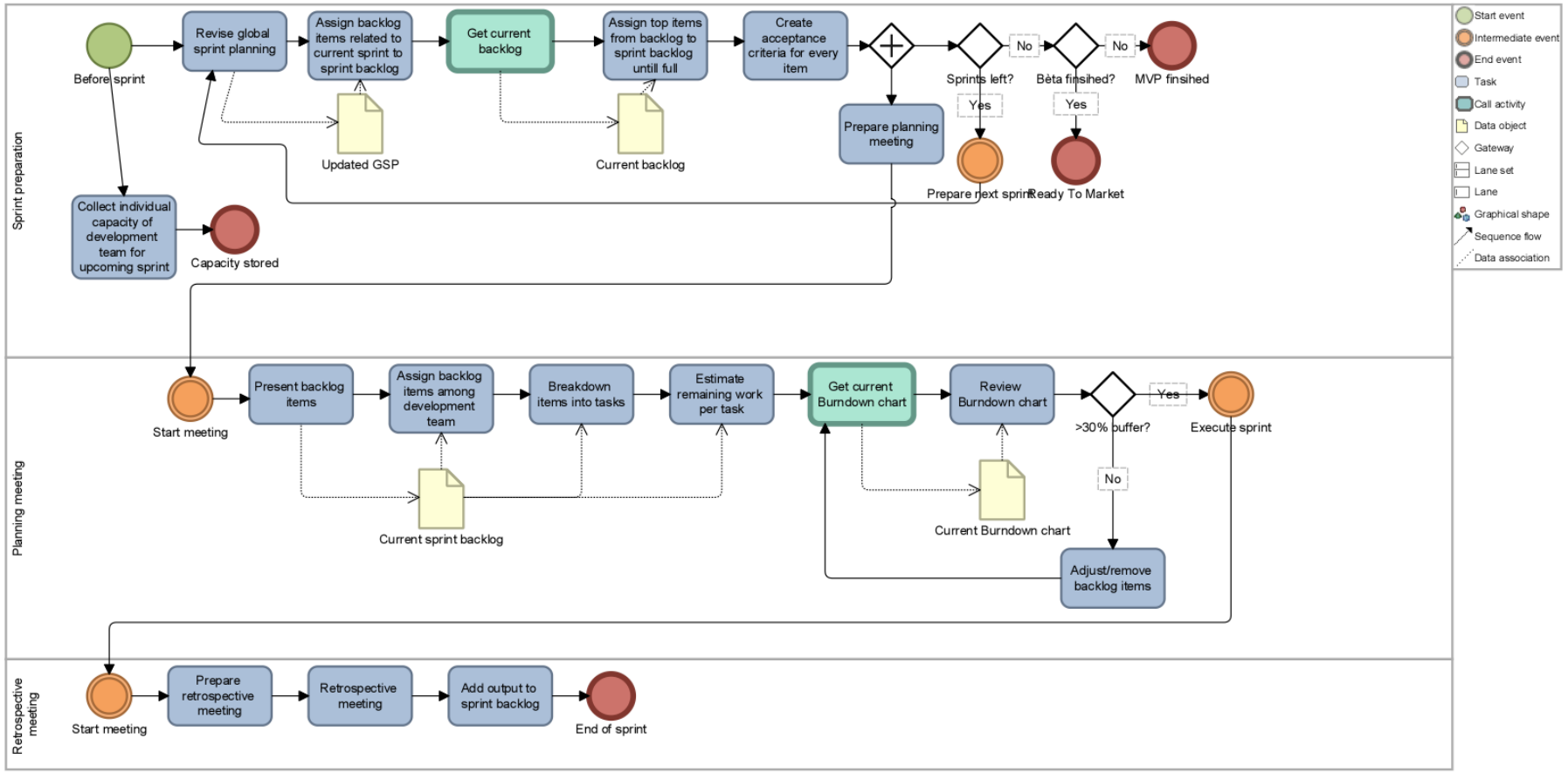


Figure 60 As-Is BPMN Sprint planning

9.13.3 Weekly planning

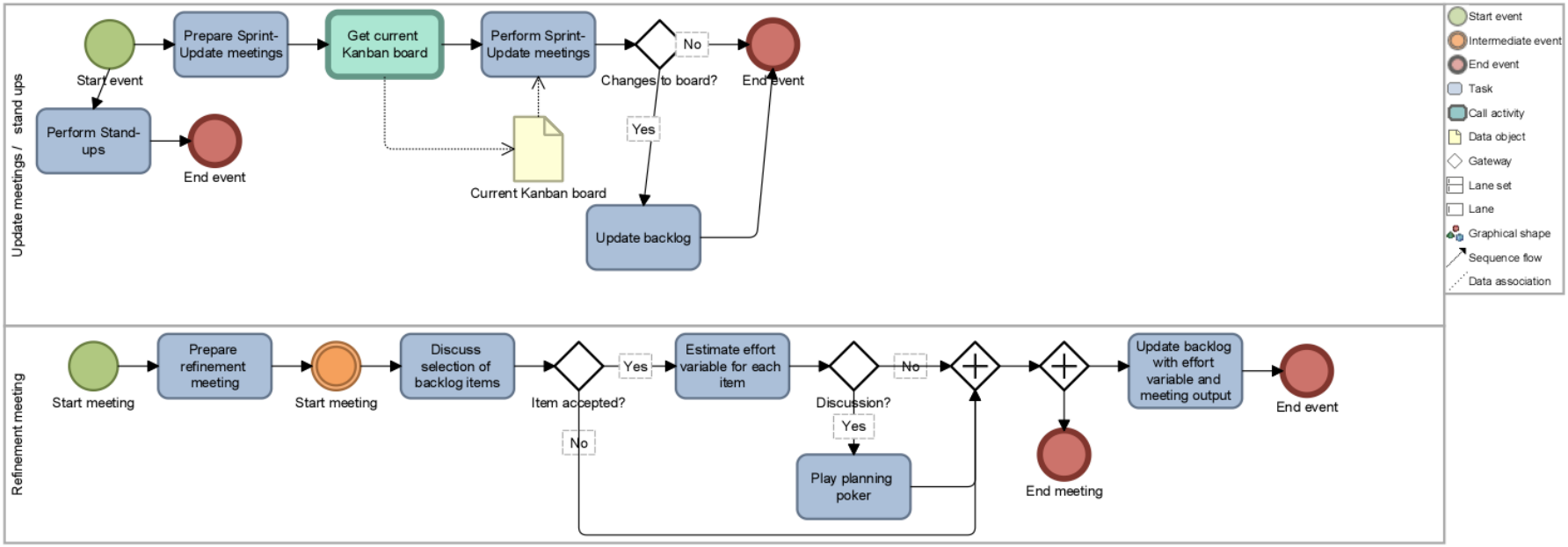


Figure 61 As-Is BPMN Weekly planning

9.13.4 Daily planning

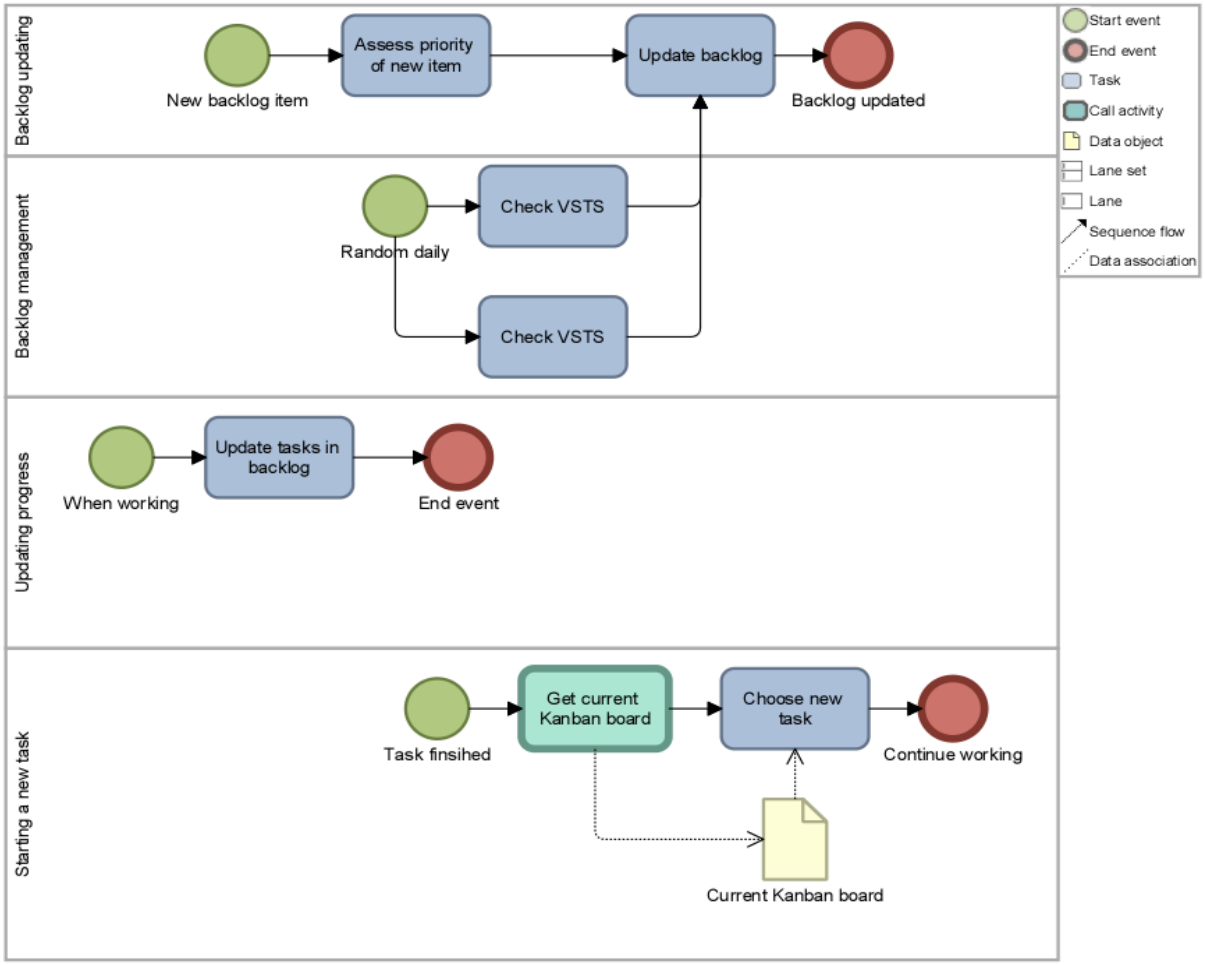


Figure 62 As-Is BPMN Daily planning

9.13.5 Visual Studio Team Services (VSTS)

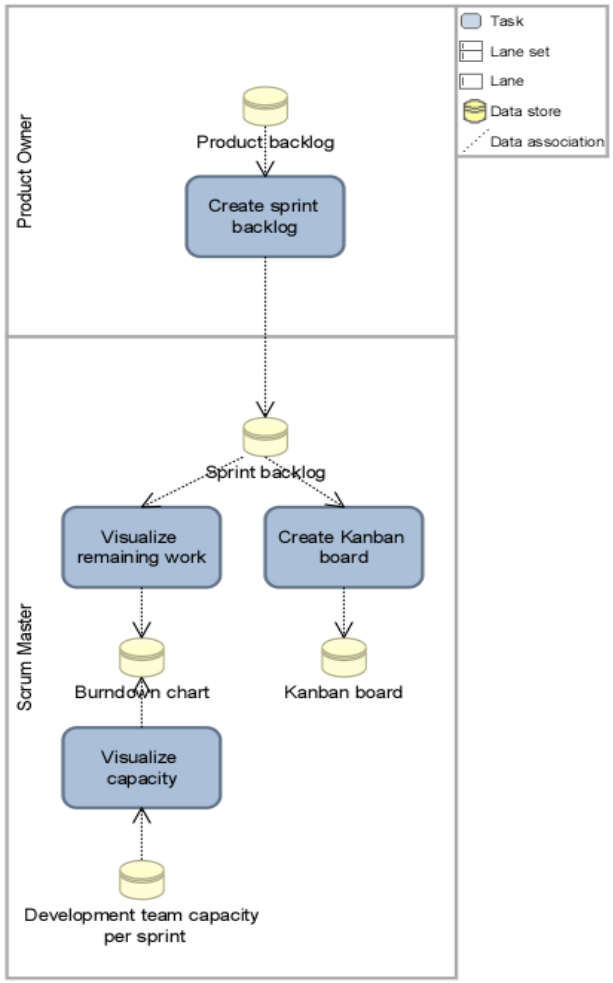


Figure 63 As-Is BPMN VSTS

9.14 Analysis added value

In Figure 64, the results of the questionnaire are shown first in full. And, with consecutively the Product Owner results dropped, the Scrum Master results dropped, and the Fulltime development team results dropped. The amount of value elements or activities add to overall performance has been operationalized on scale from 1-5 where 1 is the lowest. Dividing this value by the amount of work (in hours) it costs per sprints gives a useful overview as can be seen in Figure 65.

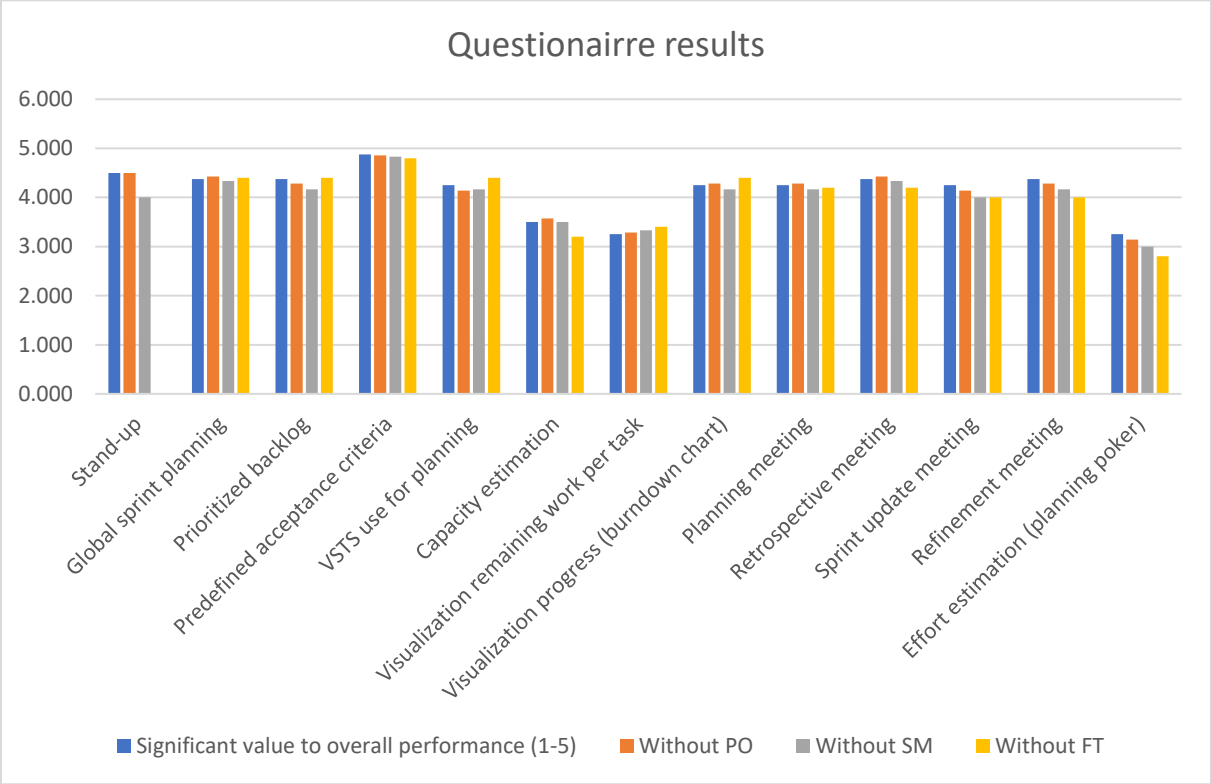


Figure 64 Value added questionnaire results

It's clear that in some cases students value activities or elements less than their fulltime colleagues. This can be seen by occasional relatively low bars. This is clearly seen regarding the capacity estimation and the effort estimation. However, students also score relatively low regarding various meetings (except for the planning meeting)

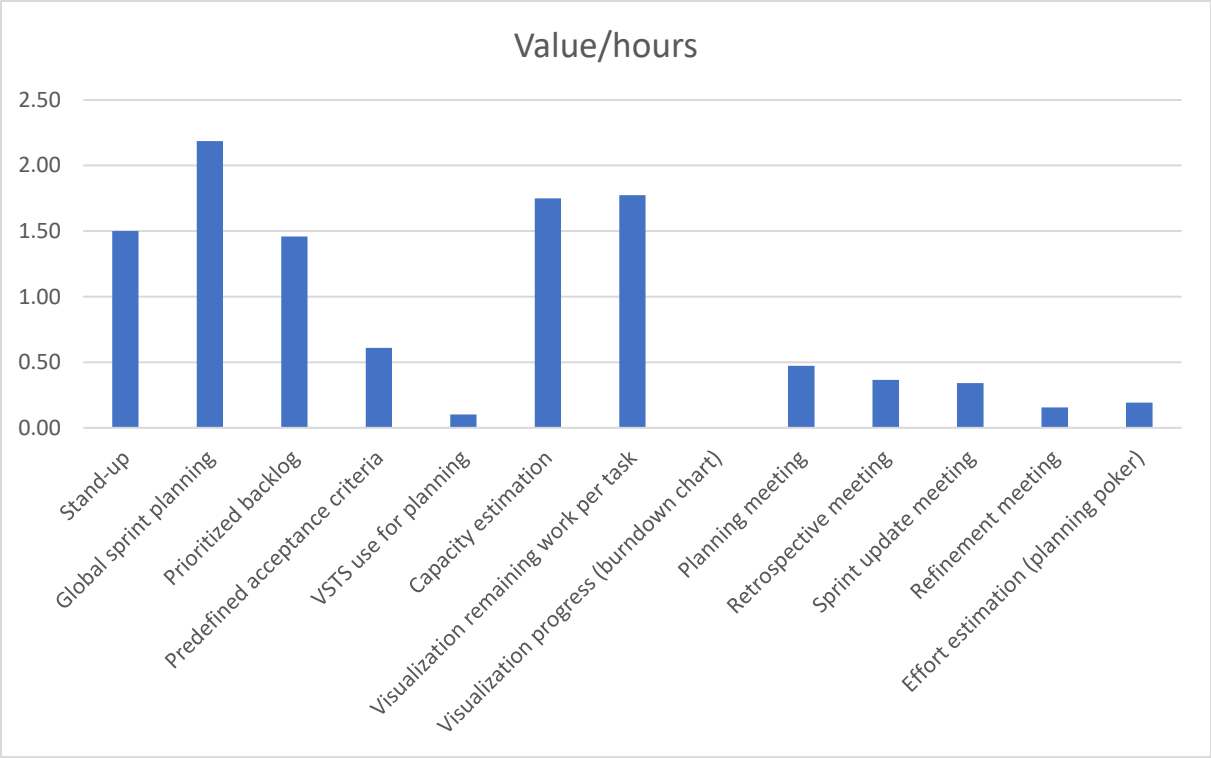


Figure 65 Value added to overall performance per hour spent

Knowing this, it makes sense to focus on improving either the lower scoring elements or activities either by improvement or decreasing the amount of time necessary to provide the element or activity. It is good to focus on the lower scoring elements. When an element has a score of 0. This means no works is necessary to provide for it. When considering all the elements and activities that score lower than 0,50 we find the following:

- VSTS use for planning
- Planning meeting
- Retrospective meeting
- Sprint update meeting
- Refinement meeting
- Effort estimation (planning poker)

As we distinguish a difference in measurements for students as opposed to the entire team. We review the same graph for only students in Figure 66.

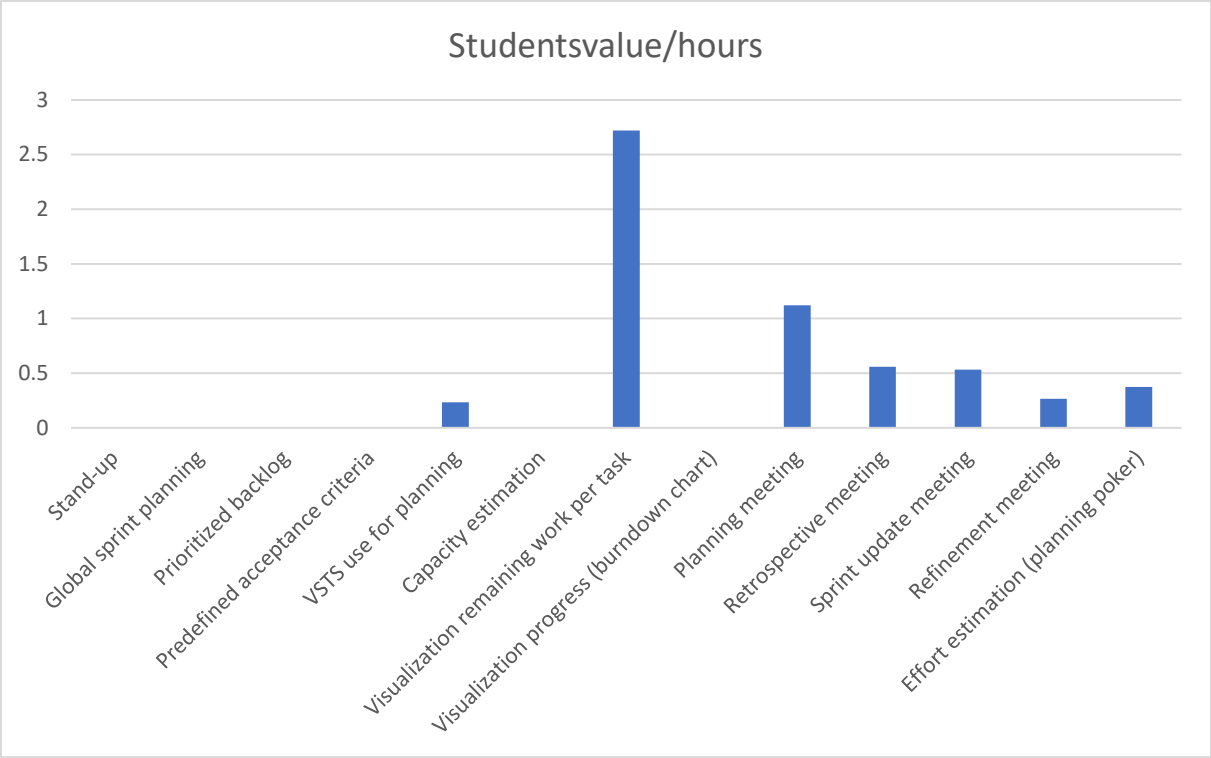


Figure 66 Value added to overall performance per hour spent (for students)

When reviewing this figure, we see that for almost half of the elements and the activities no working hours from students are necessary. We also see that the elements wherefore work is required score higher relatively to the scores for the entire team. This means that the low scoring elements defined in the previous parts are mostly caused by work done and/or value given by fulltime working employees. Still we identify the following items to be low-scoring:

- VSTS use for planning
- Refinement meeting
- Effort estimation (planning poker)

9.15 Roles related to activities

Table 36 Roles As-Is model

level	name	R	A	C	I
Strategic planning	Operationalize product idea	PO	BO	BD	DT
Strategic planning	Define planning cycle (4-6 months)	PO	BO	DT	
Strategic planning	Create Epics and Features	PO			
Release planning	Create global sprint planning	PO		DT	BO
Release planning	Create backlog items from global sprint planning features	PO			
Release planning	Prioritize items in order of occurrence on global sprint planning	PO			
Release planning	Store items in Visual Studio Team Services	PO			
Sprint preparation	Revise global sprint planning	PO			BO, DT
Sprint preparation	Assign backlog items related to current sprint	PO			
Sprint preparation	Get current backlog				
Sprint preparation	Assign top items from backlog to current sprint until full	PO			
Sprint preparation	Create acceptance criteria for every item		PO		DT
Sprint preparation	Collect individual capacity of DT for upcoming sprint	SM		DT	
Sprint preparation	Prepare planning meeting		PO		
Planning meeting	Present backlog items	PO		DT	
Planning meeting	Assign backlog items among DT	DT			PO
Planning meeting	Breakdown items into tasks	DT		PO	
Planning meeting	Estimate remaining work per task	DT			PO
Planning meeting	Get current Burndown chart				
Planning meeting	Review Burndown chart	DT		PO	
Planning meeting	Adjust/remove backlog items	DT		PO	
Retrospective meeting	Prepare retrospective meeting	SM	PO		
Retrospective meeting	Retrospective meeting	DT		PO	
Retrospective meeting	Add output to sprint backlog	PO			DT
Meetings	Prepare Sprint-Update meetings	SM			
Meetings	Get current Kanban board				
Meetings	Perform Sprint-Update meetings	SM		DT	
Meetings	Update backlog	SM		DT	
Meetings	Perform Stand-ups			FDT	
Weekly planning	Prepare refinement meeting		PO		
Refinement meeting	Discuss selection of items for upcoming iteration	PO		DT	
Refinement meeting	Assess effort variable for each item	DT			PO
Refinement meeting	Play planning poker	DT	SM		PO
Weekly planning	Update backlog with effort variable and meeting output	PO			DT
Backlog management	Assess priority of new item	PO		DT	
Backlog management	Update backlog	PO			
Backlog management	Check VSTS	SM			
Backlog management	Check VSTS	PO			
Daily planning	Update tasks in backlog	DT	SM		
Daily planning	Get current Kanban board				
Daily planning	Choose new task	DT			

9.16 Ideal division of responsibility (roles)

Table 37 Roles Ideal model

BO = Business Owners, BD = Business Developers, PO = Product Owner, SM = Scrum Master, DT = Development Team, US = Users

	R	A	C	I
• Strategy planning				
○ Company vision (Why)		BO	BD, PO	SM, DT
○ Company mission (How)		BO	BD, PO	SM, DT
• Portfolio planning				
○ Goal (What)	BD	BO	PO	SM, DT
• Product planning				
○ Product vision	PO	BO	BD, DT	
• Release planning				
○ Release goal (Epic)	PO	BO	BD, DT	
○ Product backlog (Features)		PO	BD, DT	BO
• Iteration planning				
○ Sprint goal	DT	PO		SM
○ Product backlog refinement		PO	DT, US, BD	BO
○ Planning meeting (item selection)	PO, DT			SM
○ Planning meeting (task creation)		DT	SM	
○ Sprint backlog	DT		PO, SM	
• Daily planning				
○ Daily scrum meeting	DT		SM	
○ Artifacts update	DT		SM	
• Sprint result				
○ Review meeting	DT		PO, US, BO, BD	
○ Product increment		PO		BO, BD, DT, US
○ Feedback from users		PO	US	DT, BO, BD
○ Retrospective meeting		SM	PO, DT	
• Information from metrics				
○ Sprint and project planning				
▪ Velocity/capacity	DT		SM	BO, PO
▪ Effort estimate	DT	PO	SM	
▪ Remaining work per task	DT		SM	
○ Sprint and project progress tracking	SM		DT	PO
○ Understanding and improving quality		SM	PO, DT	
○ Fixing software problems	PO			
○ Motivating people				

9.17 Agile software development practices at other companies

9.17.1 Nedap

Software engineer / Product Owner

At Nedap I was shown around all the different departments Nedap has and gave me the chance to interview various development teams. All the development teams have a different way of working related to the team size, the application being developed or the presence of remote developers. Some teams also have students. But these students just get an assignment to make something and are let go. The atmosphere at Nedap is very free. Everything is possible if there is a good reason to do something.

It quickly became clear that nobody does Agile or Scrum exactly as it should be done. One person even explains that if you do it exactly the right way, you are doing it wrong. Every team takes elements from Agile they see fit. A trend is, that when teams are more complex (larger or scattered) there is more use of Scrum frameworks (e.g. GitHub or Jira). Smaller teams can still get around with talks around the coffee machine. Teams very randomly use planning or retrospective meetings. However, every team does have moment where they can present their progress to others (as in a demo or review). So, functionality is added constantly. Product owners are always a part of the team. Therefore, decisions are also made with an entire team. Even regarding prioritizing of the backlog. Teams that don't have a roadmap (I believe because nobody takes the time to make one) do wish they had one. Scrum masters are very scarce. Some teams do have one (sometimes strict, sometimes not so much). One team used to have a Scrum master, but now that everybody is so familiar with the way of working, the necessity disappeared. This team is also able to do a stand-up (including a 360 camera for remotes) with 22 people in 10 minutes. After these 10 minutes people can have discussion in pairs or smaller groups. Prioritization is done randomly. If someone thinks something is important, he/she can give an item a high priority.

Meeting all the different teams I noticed the following regarding various Agile as shown in Table 38.

Table 38 Agile element summary interview Nedap

Element	Remarks
Scrum	Used very often with a Kanban board to keep track of tasks
Jira	Used as a digital Scrum board
GitHub	Can contain a Scrum board (is where all the code is like VSTS etc.)
PO	A part of the team who ensures he/she has all the information from stakeholders necessary to build the right product.
Scrum Master	Not very present at Nedap. But can solve arguments and timebox activities.
Stand-up	Very often done (at a set time, as short as possible <15 minutes)
Retrospective	Most often done quarterly. Not as a part of a sprint.
Review/Demo	At Nedap every 6 weeks all teams can present their progress internally. Furthermore, some aspects of the application are updated weekly, others real-time)
Sprints	Teams do not work in sprints often. This is very up to the state of the application being worked on. For continuously (real-time, cloud updated) developed software sprints aren't necessary.
Sprint kick-off	Only done by one team including a planning meeting.

9.17.2 Frontwise

Partner

Frontwise has its own way of working which strongly resembles the traditional approach. However, they have ensured that they obtain feedback from their customers while working. Usually they only work on one final version of a project. They have a few customers for whom they provide support. For these customers there are backlogs, issue trackers and such. For the other projects this is usually not the case. This shows that when working towards a v1 (or MVP/MMP) it isn't always necessary to work fully Agile. This had been done once, where every design step was an iteration, the effect of this approach was very high costs (e.g. more manhours). Regarding Agile elements, Frontwise has a daily standup. This was not always the case. But, when one member of the team was abroad this appeared to be very good. There is also a weekly meeting.

Usually they are able to stick to their planning. This is due to experiences from the past.

The situation at the customer is also important. Their process should be in order, sometimes the client also has certain tasks, these should be performed in time. Clear process steps are also important, as well as evaluation. It's also good to define as much as possible upfront to not make mistakes (or set the right expectations).

For one customer, for whom they provide support, they let the customer prioritize all the 'tickets'.

Frontwise has their own flow and an UX (User eXperience) flower (https://www.frontwise.com/) are key methodologies/elements for the development.

9.17.3 Moneybird

Software engineer

Moneybird doesn't have a very tight planning. However, they do work Agile. They don't have an elaborate planning and don't spend much time on documentation. They develop small things. Therefore, they can continuously integrate. So, Agile is viewed as releasing continuously. And, even though there isn't a detailed planning, there's always a clear point on the horizon. This point is related to the vision, which is safeguarded by the founders. Every quartile there is a possibility to bring in new ideas (which should be in line with the vision). Moneybird works with OKR's (Objectives and key results). These are goals set per quartile which are always a bit too ambitious. This provides them with a clear focus. Then, people in the team work on something together. Ivo explains that if there is too much freedom things can become complicated. Working on something together makes it go faster.

Every Monday there's a meeting which strongly resembles a stand-up. There is a focus on limiting work in progress. Doing so, nobody focuses on more than three tasks a week. Regarding Scrum, Ivo explains you shouldn't necessarily follow all the rules. Moneybird also doesn't work with user stories. He does agree that in a younger company structure can be very useful. But later on, when there is more experience, this becomes less important.

Moneybird always has students work on a different part of the application (than the development team is working on), or a small part of it. Because students can't work as much and are very variable. They don't communicate a deadline to the students. Their work shouldn't be blocking for the rest of the team. Sometimes, when it is known a student has the time, students can help with the main focus. The projects for students are usually of a middle-large size and aren't necessary.

9.17.4 Company X

Technology consultant

As Company X works on different projects, it depends on the customer if Agile or Scrum practices are applied. Usually, this is the case and they work with user stories and follow Agile practices by the book. The sprint length can still differ per customer. The size of teams is also dependent on the wishes of the customer, and therefore variable. Preferably, there are always a defined Product Owner and Scrum Master. It's even crucial that there is a PO. Usually, the PO is provided by the customer. This does sometimes lead to a knowledge gap. Within Company X it doesn't matter how or when you spend your hours. If the work isn't done at the end of sprint it is moved to the next. And if there's a hard deadline, more hours should be put in.

A discovery sprint is always done (sprint 0). When there's a backlog it is prioritized by the customer. But it is also possible for somebody from the team to add a ticket (with a high priority). Criteria for prioritizing are business value and the effect on the software (for example, does it negatively affect other parts of an application). There is worked according to acceptance criteria which are defined by the product owner. The given, when, then format is always used. There is also a clear goal and a roadmap which continuously evolves. This roadmap can be for a year or perhaps even for five years ahead.

Work often isn't finished at the end of a sprint. This can be due to too much work or unclear stories. This can lead to an incorrect effort estimation (amount of points). Remaining work isn't estimated for tasks. The average velocity is assessed in order to plan ahead.

9.17.5 Trimm

HR manager

Trimm has been working on software for over 25 years already. They mostly work in continuous development projects. 6/7 years ago they started working with multidisciplinary teams (vertical as well as horizontal integration). This also led to Scrum and/or Agile working teams. Previously to his they worked according to the traditional software development method. For 3/4 years they have been working with self-organizing teams (these teams do have hierarchy).

From 120 employees about 20/30 are students. Students are free to participate in Agile planning events, so it is not obligatory. Students are used as is seen fit, there are no standing expectations or frameworks regarding this aspect. Trimm attracts student through PIT (talent program).

Every team uses Kanban and/or Scrum artifacts. But It is very clear that every team can decide for itself how it wants to work. Sprint lengths also differ. Every team has a SM (who is also a developer). The frequency of meetings is also up to the teams.

The product owner role is always fulltime. The product owner's week looks as following:

- 2/3 days at customers (stakeholders)
- 1 day always goes to meetings and such
- He makes the requirements

Every team has stand-ups, retrospectives and refinements, and demo's. The retrospectives and refinements are aimed at the project. Besides this they have 6-8 weekly meetings across teams where people with the same role can discuss progress and share information. Prioritization is done together by the senior designer senior developer and the PO. The PO doesn't do this alone. Yet again, it all depends on the team.

Tom often explains it's hard for people (especially in IT) to make contact with others. It's often hard to make contact, set expectations and apply changes to an organization. Assumptions, interpretations and expectations can be deadly. A difficult thing about planning is that people are afraid of commitment. Things such as capacity and velocity are assessed. But VERY difficult to control. You should always review your current (Scrum) process. This should be facilitated in a smart way but watch out that you don't slip into a routine.

9.17.6 Vanderlande

Vanderlande supplied me with seven editions of their monthly Lean-Agile newsletters. These newsletters contain interesting insights which are taken into account in the concept matrix.

9.18 Concept matrix interviews companies

Table 39 Concept matrix interview companies

Concept	Nedap	Frontwise	Moneybird	Company X	Trimm	Vanderlande	**Findings**
Agile software development	Every team can decide their own way of working. Teams apply Agile elements where they see them fit. Most teams don't see the necessity because of continuous development	They have their own approach which more or less resembles the traditional approach	Not a very tight planning but the work Agile. They don't spend much time on documentation and develop small things. Therefore, they can continuously integrate	Always adapt to the wishes of the customer (work project based). They work according to Scrum standards	Trimm started with continuous development 6/7 years ago (this led to Scrum/Agile). For 3/4 years they have self-organizing teams (with hierarchy)	From Lean-Agile Monthly newsletters with interviews. Apply Agile when requirement are far from clear.	**Agile/Scrum can be implemented in different ways. Often teams can organize themselves.**
Team size	3 - 22	3	10, out of 25 employees total	Also up to the customers wishes. With large teams the Spotify model can be applied	8 - 12		**Ranges from 1-22. 8-12 appears to be an average size.**
Number of teams	>30	1	1		7 innovation teams	Reduced amount	-
Product Owner	Every application has one PO which is a part of a team. This person ensures he/she has all the information from stakeholders necessary to build the right product	Each project has a product owner	There is no defined PO. Feedback is collected from support. Testing different things (A vs. B) is difficult, individual customers can be approached or surveys can be used. Everybody can take initiative to do this	In the ideal situation this is done by the customer, the presence is crucial. Sometimes, there's a knowledge gap when the PO is from the company	Every team has a FT PO: 2/3 days at customers 1 day for meetings and such Create requirements	Yes	**In companies with more application there is always a clear PO. Companies with one product don't necessarily have a PO. It can be done by the customer, but this doesn't seem ideal.**
Scrum Master	Not often used. Sometimes used for argument solving or timeboxing. Teams notice that as they gain more experience the necessity of a SM becomes less and less	NA	NA	Preferably filled in by a customer	Every team has a SM who is a part of the team	Yes	**Not always present, seems to be good when a team is not yet completely familiar with Agile.**
Students	Students are usually just given tasks separate from the core work. So, they cannot be blocking for progress	NA	Students always work on a different part of the application than the development team is focusing on (or a small part). They don't communicate deadlines to students and their	Some departments have students working for them. But at Company X it doesn't matter when (and how) you spend you hours	20/30 out of 120 employees total. Students are free to participate in events, not obligatory. Students are used as is seen fit, there are no expectations or		**Students are used in a very free way. They don't often work on core tasks.**

			work shouldn't be blocking. They can focus on middle-large size projects		frameworks for them. They attract students via PIT (talent program)		
Sprint Length	Usually no sprints. When there are sprints 1-3 weeks.	NA	NA, OKR	What the customer wants	Differ per team	2 weeks	**1-4 weeks.**
Tools	Jira, GitHub	Frontwise uses random or self-created tools	NA	Customers often have Jira. And Company X has some own tools	JIRA, Slack	JIRA	**Jira is used a lot.**
Planning meetings	Not often used, most teams do have weekly team meetings. One team does an actual planning meeting	Weekly meeting	They have a Monday meeting which strongly resembles a stand-up. The founders safeguard the vision. Every quartile somebody can bring in new ideas	Done	Frequency of all meetings is up to the team	Joint planning every 10 weeks	**Usually teams have a weekly meeting, more dedicated Scrum teams do have this.**
Retrospective meetings	Usually every 3 months, unrelated to sprints	NA	Is done every quartile	Every 2 sprints	Yes. You should always be critical towards you own (Scrum) process. This should be facilitated in a mart way, and watch out not to slip into routine		**More often done per quartile.**
Review meetings (internal)	Every six weeks all teams of the Healthcare department can give a presentation to the entire department	NA	NA	Done	Yes, 6-8 weekly meetings across teams where people with the same role can share information		**Done every 6-8 weeks.**
Product demo's (external)	Some aspects are updated weekly, other real-time	They continuously ask their customer for feedback. However, they don't let the customer decide to much as they are a creative agency which needs their freedom	NA	Done	Yes		**Sometimes other ways are used to collect feedback from customers. But there is always a structure in place to do so.**
Refinement meetings	NA	NA	NA	Done	Yes		**Only done by dedicated Scrum teams.**

Daily meetings	Usually done at a set time, as short as possible. The larger the team, the higher the necessity	Standup	No daily stand-up. Nobody should work on more than 3 tasks a week. These tasks should always bring you closer to the goal	Done	Yes		**Smaller teams have this less than larger/more complicated teams.**
Backlog management	Decisions are made as a team regarding prioritization. This is done intuitively	The customer gets to prioritize	NA	The customer prioritizes, but somebody developers can also create a ticket and give it a high priority. Prioritization is done according to business value and effect (does it break anything)	Prioritization is done by the senior designer, senior developer and the PO. But it does depend on the team.	Break down tasks	**With project-based development the customer decides. Otherwise, the team has a fitting structure.**
Roadmap/themes	Teams that don't have one would like to have one	NA	There isn't always a clear planning. There's a clear point on the horizon. The team usually works on larger projects which they always finish.	There is a roadmap and a clear goal. The roadmap also moves with the time. Sometimes it's for a year, could also be for 5 years		Start with an MVP	**A clear goal seems more important. A roadmap is always good to have.**
Scrum/Kanban	Used often with a Kanban board to keep track of tasks. Don't do it exactly following the rules	NA	Don't necessarily follow the rules. Kanban boards are used and there is a focus on minimizing WIP. Structure flow into experience	Done according to the book	Every team decides for itself how it want to work.	Scrum/SAFe. Seems to be very dedicated	**Kanban are used very often. More experienced teams need less structure. Either teams are dedicated or cherry pick.**
Continuous development	Always	Only for a few customers	Somebody is always assigned to maintenance. This role is passed on often	No, project bases. Discovery sprint (0) followed by iterations. It stops when the customers says its done (due to budget or finished work). Sometimes there are service contracts	Yes	Yes	**Somebody needs to be assigned to maintenance. Continuous development is a standard.**
Acceptance criteria	Some teams have a standardized format		NA	Created by the PO	Created by PO		

Achieving sprints	NA	NA	As Moneybird works with OKR's they always set slightly too ambitious goals per quartile. This provides clear focus	Planned items are often not finished during sprints. This can be due to too much work, wrong estimations or something else. You should always stay realistic and honest towards customers and re plan these items. If items are left, they are picked up in the next sprint. If there's a hard deadline more work has to be done.	NA	Problematic	**Quite often not all items are finished at the end of a sprint. This should be handled with care, so no work gets left behind.**
Positive influence on planning	NA	NA	Working on something together, there is a fixed amount of time (3 months) keep a variable scope. Everything comes down to details: planning 3 months in detail isn't possible. 1-2 weeks should be possible	Clear stories, capacity (enough), a good PO (knows Business Value, fast decision maker), DevOps is applied, small number of bugs	NA		**Clear stories, capacity (enough), good PO (assess BV and fast decisions), prevent bugs from occurring, working together, variable scope, plan micro not macro (too much).**
Negative influence on planning	NA	NA	When there is too much freedom things can become more complicated	Too much overhead and negative side of positive points	Its har for people (especially in IT) to make contact with others, set expectation and apply changes. People are afraid of commitment		**Overhead, too much freedom (individualism), bad communication, expectations, fear of commitment and downside of positive points.**
Capacity estimation			NA	Velocity, no remaining works for tasks	Capacity and velocity are assessed, but very difficult to control	Velocity is assessed	**Velocity is used for an indication, but hard to control.**

9.19 Capacity estimation analysis

To gain insights in the capacity of students results of an analysis of worked hours (01-01-2018 until 31-05-2018) are shown in Figure 67 and Figure 68.

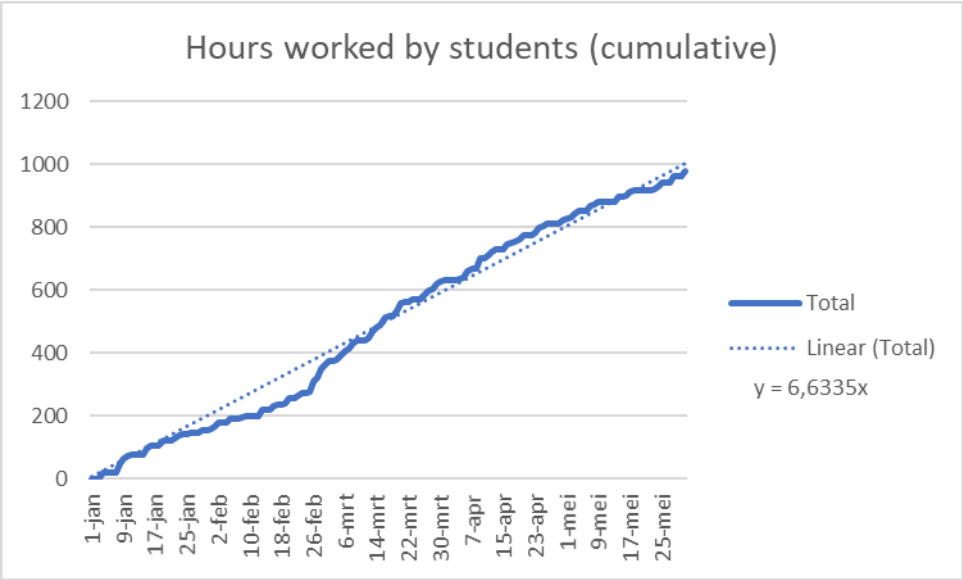


Figure 67 Total hours worked by students (cumulative)

We can see that the cumulative flow of hours is relatively linear. And in this linear flow every day the students (combined) work 6,335 hours. It's important to realize that students aren't bound by a normal working week. They work in weekends as well. Therefore, 46,4345 hours a week, and 139,3035 hours a sprint are done according to this linear prediction.

We can also look at the daily average. The daily average is 6,487 hours, which is slightly lower than the trendline prediction. We can assess the trustworthiness of this results through confidence intervals. Confidence intervals allow to say how sure you are that the actual value will be within a certain interval. A confidence interval is calculated as in the following equation:

$$Confidence\ interval = \left(\bar{x} - \frac{z^*\sigma}{\sqrt{n}}, \bar{x} + \frac{z^*\sigma}{\sqrt{n}}\right)$$

$$\bar{x} = population\ mean$$
$$z^* = critical\ value$$
$$z^* = \phi^{-1}\left(1 - \frac{\alpha}{2}\right)$$
$$\phi = Cumulative\ Distribution\ Function\ (C.D.F)$$
$$\alpha = degrees\ of\ freedom$$
$$\sigma = population\ standard\ deviation$$
$$n = population\ size$$

For the critical value the normal distribution is used as the standard deviation is known and the sample size is more than 30. When we calculate the confidence intervals for an interval of 90%, 95% and 99% we get the results as shown in .

Table 40 Confidence intervals average capacity prediction

α	**0,1**		**0,05**		**0,01**	
Boundaries	[Low bound	High bound]	[Low bound	High bound]	[Low bound	High bound]
Confidence	5,483	7,490	5,291	7,683	4,915	8,058

So, with a certainty of 90% we can say students will work at least an average of 5,483 hours a day and no more than 7,490 hours a day. How these hours are divided across sprints is shown in Figure 68 where each color is a different student employee.

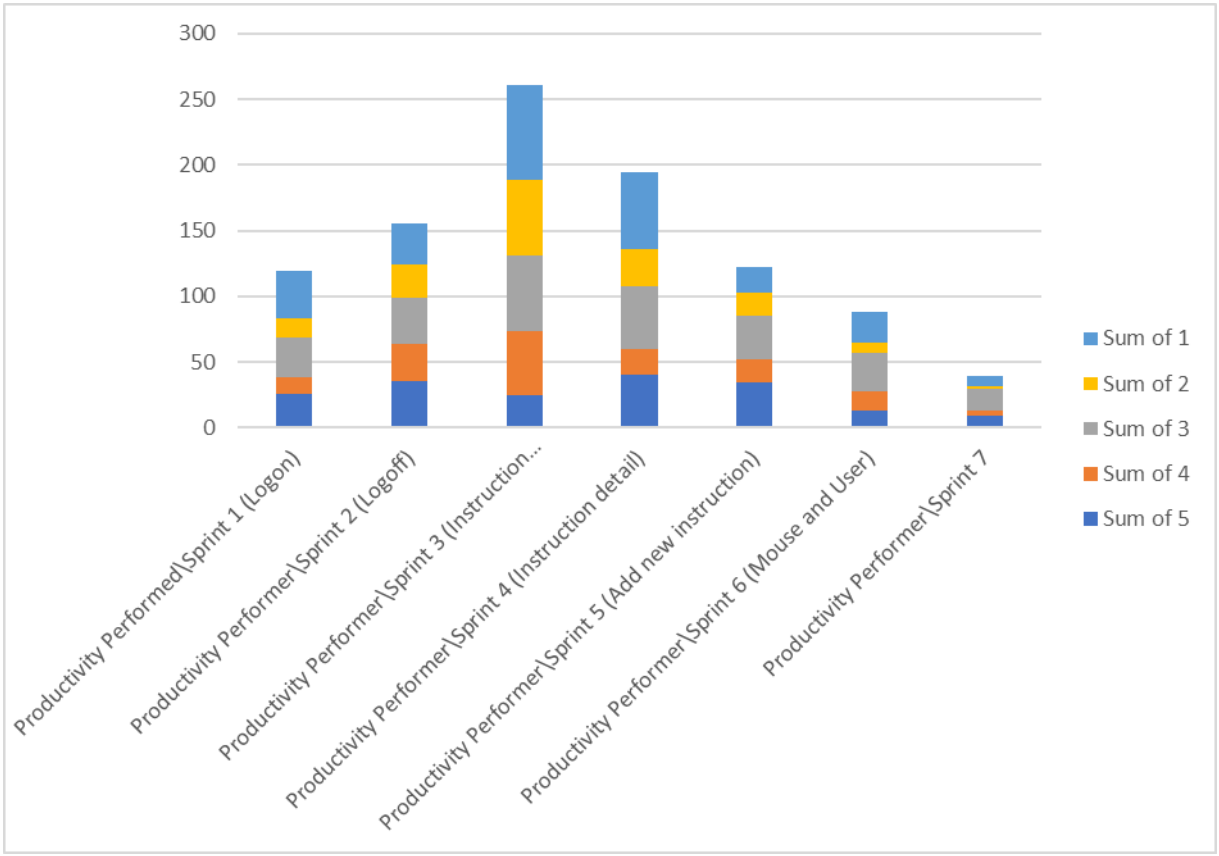


Figure 68 Hours worked per sprint, per student employee

Here we see that there is no trend per sprint related to the hours worked. Also, there is no student who work a significantly higher number of hours compared to others. The reason for the peak in sprint 2 is because the length of the sprint was extended by almost 3 weeks. Sprint 4 was extended with one week.

In sprint 4 and 5 the capacity was estimated as shown in the As-Is model from chapter 3. We can analyze the worked hours with the capacity from these sprints to gain a view on how accurate the capacity estimation is. Analysis with this is done in regarding Figure 69.

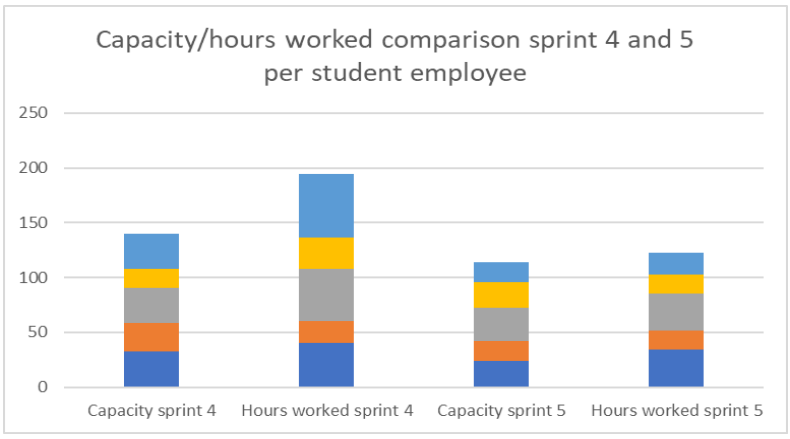
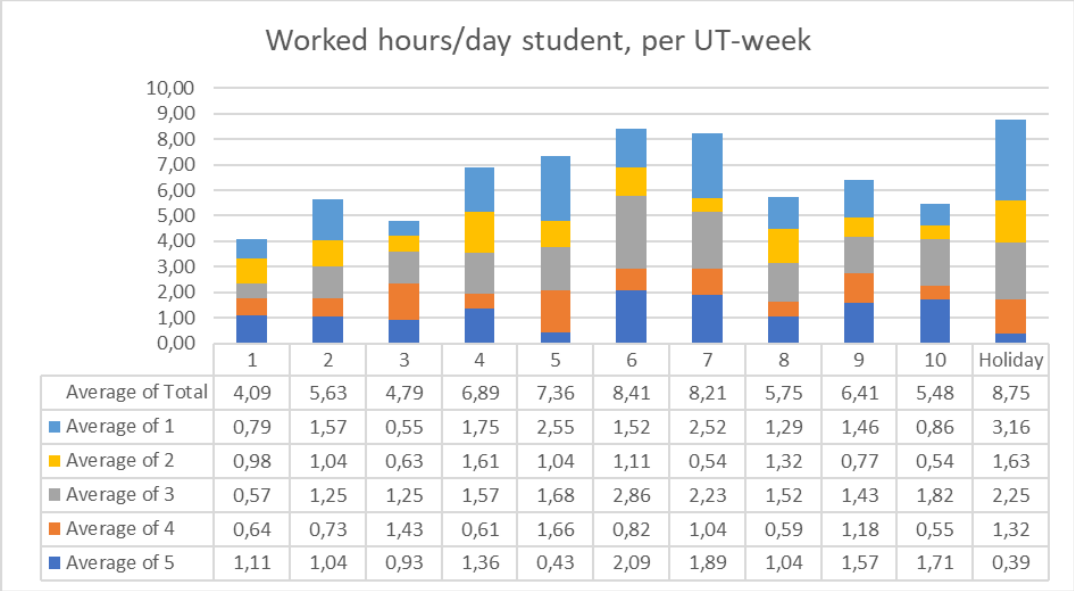


Figure 69 Capacity/hours worked comparison sprint 4 and 5, per student employee

Here we see that in sprint 4 there were significantly more hours worked then estimated in the capacity. In sprint 5 there were only a few hours more worked than estimated. However, it can be said that this way of estimation is quite volatile to errors. The capacity estimation is 17,44% lower on average than the actual worked hours.

As explained before in this paragraph capacity is related to study pressure. Therefore, we compare the hours (01-01-2018 until 31-05-2018) with the schedule of the University of Twente in Figure 70. We determine how many hours are worked in each quartile as defined by the University of Twente (including holiday weeks).



	1	2	3	4	5	6	7	8	9	10	Holiday
Average of Total	4,09	5,63	4,79	6,89	7,36	8,41	8,21	5,75	6,41	5,48	8,75
Average of 1	0,79	1,57	0,55	1,75	2,55	1,52	2,52	1,29	1,46	0,86	3,16
Average of 2	0,98	1,04	0,63	1,61	1,04	1,11	0,54	1,32	0,77	0,54	1,63
Average of 3	0,57	1,25	1,25	1,57	1,68	2,86	2,23	1,52	1,43	1,82	2,25
Average of 4	0,64	0,73	1,43	0,61	1,66	0,82	1,04	0,59	1,18	0,55	1,32
Average of 5	1,11	1,04	0,93	1,36	0,43	2,09	1,89	1,04	1,57	1,71	0,39

Figure 70 Average worked hours per day, per student, per UT-week

In this overview it can be seen that the most hours are worked in the middle of a period of the University and if there's a week of holiday. It's surprising to see that in the beginning of a period less hours are available. There is no clear explanation for this. Now we can compare the results of various estimations to the actual worked hours in the first five sprints. This gives the overview shown in Figure 26 and Table 14.

9.20 Roles To-Be model

Table 41 Activities To-Be model with roles

Activities (in order)	**R**	**A**	**C**	**I**
Review company vision (why)		BO	BD PO	SM DT
Review company mission (how)		BO	BD PO	SM DT
Review company goal (what)	BD	BO	PO	SM DT
Review product vision	PO	BO	BD DT	
Set release goal	PO	BO	BD DT	
Create product backlog		PO	BD DT	BO
Set sprint goal	DT	PO		SM
Select related items from approved items	PO DT			
Commit to items (total effort)	PO DT		SM	
Assign responsible person per item		DT	PO	
Create tasks per item		DT	PO	
Create remaining work estimation per task		DT	PO	
Daily Scrum meeting	DT		SM	
Update artifacts	SM		DT	
Assess (new) product backlog items priority	PO			DT BO BD
Create acceptance criteria for product backlog items	PO			
Present items to Stakeholders		PO	US BO BD	
Present items to Development team		PO	DT	
Adjust/remove item	PO			BO BD
Adjust/remove item	PO			DT
Estimate effort for item	DT	PO	SM	
Have Review meeting	DT		PO BO BD	
Increment product		PO		BO BD DT US
Collect feedback from Users		PO		BO BD DT US
Prepare Retrospective meeting	SM			
Have Retrospective meeting		SM	PO DT	

9.21 Planning tool detailed development

9.21.1 Planning tool design

The first part considered designing the planning tool is what it should visualize. As explained we focus on capacity, remaining work and effort. Starting with the capacity, the goal is to render the individual capacity estimations unnecessary. So, we want a weekly estimation of the hours that will be worked. As it is always good to add known information it should still be entered when somebody works a significantly different number of hours than the average. For example, if somebody plans to work either fulltime or not at all during a week. We also want to check this estimation afterwards to ensure correctness of the prediction. So, a comparison with the actually worked hours should be made. Last, it would be useful to discover a correlation between the velocity and the worked hours.

Then, through adding the remaining work to the capacity a burndown chart can be created. This is done in VSTS automatically. However, it would be useful when a line is added which shows the expected flow of finished tasks. Through analyzing past weeks of sprints an estimation of the burndown can be created.

Last, the effort estimation can be used for different estimations. First of all, as is enabled in VSTS, the velocity per sprint can be shown. However, this doesn't tell a person much on itself. When this is used as a metric to estimate how much can be done during the next sprint some calculations are necessary. Therefore, a prediction can be added based on all previous sprints, the last 3 sprints and based on the capacity, as explained before. Furthermore, and Epic and feature burndowns can be created to show progress and estimate a finishing date, mainly with the Epic burndown as for features it's dependent on item selection.

Furthermore, as proposed in paragraph 5.2.1, a dashboard can be created which shown changes to the task-board. This is for members of the team who weren't present at the daily Scrum meeting to stay informed.

All this can be created using Microsoft PowerBI. This is tool where data can be transformed (similarly to MS Excel) and visualizations can be created using real-time data. Therefore, manual updates aren't often necessary.

9.21.2 Collecting data

To be able to visualize the contents of the tool as defined in paragraph 9.21.1 data needs to be collected from several sources. The collection of the main data is shown in Table 42.

Table 42 Data collection for planning/scheduling tool

Visualization	Data	Source
Capacity	Historic working hours	TimeWriter
	Capacity as entered in VSTS	Visual Studio Team Services
	University planning	Manual, selecting the correct academic calendar via https://www.utwente.nl/nl/ces/planning-roosters/jaarplanning/jaarcirkels/
Remaining work	Remaining work per task (historically)	Visual Studio Team Services
Effort estimation	Effort per backlog item	Visual Studio Team Services
Epic/Feature burndown	Status per item	Visual Studio Team Services
Task-board updates	Changes to tasks	Visuals Studio Team Services

MS PowerBI can be connected to Visual Studio Team Services for the collection of data. Then, per item or task there is a lot of data available. An example of this is shown in Figure 71. Data that isn't necessary can be left out in order to speed up the updating process. Furthermore, data can be transformed and combined in various ways. Examples of this are addressed in paragraph 9.21.3. In some cases, data is manually entered in VSTS (for example the capacity). This is no different from the current situation. Regarding VSTS no extra work hours are required.

For data from TimeWriter further research is necessary do determine of this process can be automated. Otherwise, manual updates will be necessary. It is possible to export predetermined datasets from TimeWriter which could easily be uploaded to the PowerBI environment.

No clear way has been identified to automatically update the UT-schedule in PowerBI. Therefore, that should have to be done at the beginning of every academic year.

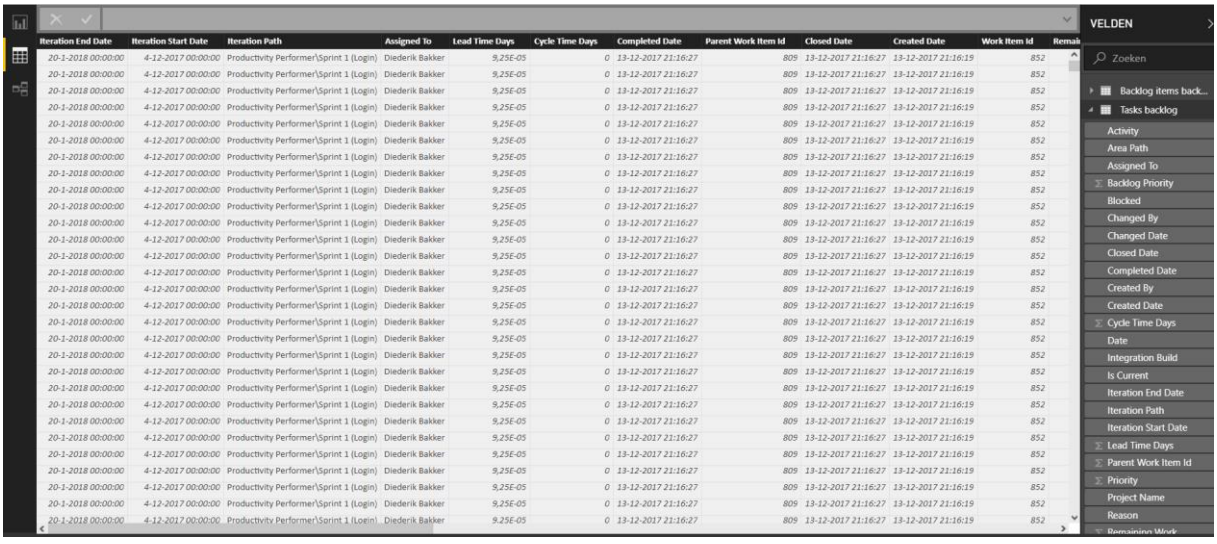


Figure 71 Example of data set in PowerBI

9.21.3 Visualizing metrics

In this paragraph visualizations for the capacity estimation, sprint progress, release progress and task changes. These visualizations contain various predictions and accuracy measurements. From this point, where visualizations of metrics are designed, it can be considered that there are real-time updated tables containing historic data regarding all backlog items, tasks and hours worked. Also, the UT-schedule is present as a table on PowerBI.

Capacity estimation

To estimate the capacity we first have to know which UT-weeks are present in the upcoming sprint. Therefore, we create a table from VSTS with the contents of Table 43. To do so the 'Items backlog' can be used.

Table 43 Iteration overview

Iteration Path	Iteration Start Date	Iteration End Date
Unique value	Unique value	Unique value

We create Table 44 at the beginning of the academic year.

Table 44 UT-week overview

Week Number	Week Start Date	UT Week
36	3-9-2018 00:00:00	1
…	…	…
27	1-7-2019 00:00:00	10

Combining both tables based on Iteration Start Date and Week Start Date enables us to assess which UT-week is active during every planned sprint. Then, the hours from TimeWriter are enriched with week numbers and based on this the UT-week is added.

Now that the data has been prepared we can calculate the average work done per UT-week by 1 student, store this in a table and create a graph similar to Figure 70. Then, we predict the weekly estimated hours a student will work for the upcoming sprint. First, we determine the current sprint and take the next. Looking through the combined table provides 3 UT-weeks. The average of these 3 weeks should be clearly shown and filled in for each student in VSTS (unless there is a significant difference).

To verify the accurateness of the prediction the total capacity per sprint from VSTS can be compared to the working hours of the entire team (assuming differences aren't caused by fulltime employees). The working hours of the team should be grouped by sprint (can be done by week number of the combined table) before calculating the total amount of worked hours. A graph showing the estimation and worked hours per sprint alongside a KPI showing the mean absolute error is a good measurement of the accurateness. It should be noted that the estimation, as well as the realization contain all the time spent on planning activities.

Sprint progress

For this, a Burndown chart can be created which is similar to Figure 14. The capacity is shown by starting with the total capacity for the sprint at day 1, then subtracting the capacity for day 1 at day 2. Until at day 15 the capacity reaches 0. Then, the remaining work is calculated daily as the sum of all the remaining work in the sprint. For the use of this the table containing capacity is filtered for each day between the 'Iteration Start Date' and 'Iteration End Date'. Then, a column is added containing the described calculation. The table containing data on all tasks is filtered to show only 1 unique 'TaskID' per date (again between the start and end date). For every day the sum of remaining work is taken.

At this moment there is an ideal trend line. This line is a linear line moving from the remaining work at the first day of the sprint to 0. This seems inaccurate, is it makes sense that this is in line with the daily capacity. For example, in the current situation there are no fulltime employees working on Friday. So the capacity is less then. To improve this line which serves to track progress we can apply the following:

$$RemainingWork_n = RemainingWork_1 - \frac{Capacity_{n-1}}{Capacity_{total}} \times RemainingWork_1$$

$$n = day\ (1, 2, .., 15)$$

$$Capacity_{total} = \sum_{n=1}^{15} Capacity_n$$

$$Capacity_0 = 0$$

Added to this, it's important to include all known information in the remaining work. Therefore, there should be tasks showing the amount of work remaining for meetings. For example, there can be a task "Planning meeting". At the planning meeting 7 people of the development team spend 1,5 hours. Then, this task is to be 10,5 hours of remaining work and is finished on Monday.

Release progress

To assess the velocity of a team an effort estimation per item is done the effort estimation is done in 'points'. A team should then know its own capacity, so they can actively commit to a certain number of items. To do this, the standard velocity chart can be used with a few extensions to predict future sprints. Different lines and predictions can serve as a helping factor when committing to a certain number of items. Various possibilities and their explanations are shown in Table 45. For all these estimations the number of sprints taken into account is variable. The estimation can, for example, be done for all available data, the current release, or a certain number of sprints.

Table 45 Estimation factors for effort estimation

Estimation factor	Calculation
Average	The average number of points done per sprint
Prediction	A prediction (including a 100-n% confidence interval)
Minimum	A constant line showing the minimum number of points done during a sprint
Maximum	A constant line showing the maximum number of points done during a sprint
Hours/point	The hours worked during a sprint divided by the number of points done. When doing this an estimation of this factor can be made and multiplied by the number of available hours in the upcoming sprint. All above factors can be calculated using this.

To create an overview of the effort estimation the table containing all backlog items is used where the column 'Is Current' is 'True'. Then, the sum per 'Iteration End Date' is selected. The aiding lines are a built-in feature.

For the Epic Burndown, as shown in Figure 34, a new table has to be created. A table is created containing all Features related to the Epic as in Table 46. This is imported directly from VSTS.

Table 46 Feature ID with name table example

Feature ID	Title
Unique value	string

Then, we create another table containing every work item and the information as shown in Table 47. This is also imported directly from VSTS. We also create from the table containing all data from work items.

Table 47 Work Item ID info for Epic Burndown example

Work Item ID	Parent ID	Created Date	Finished Date	Effort
Unique value		Date	Date	Integer

Using this table we create Table 48 which contains the date per sprint.

Table 48 Epic burndown input example

Iteration Path	Iteration Start Date	Iteration End Date	Effort left	Effort added
Unique value	Date	Date	From table 32 = sum of 'Effort' when 'Finished Date' > Iteration End Date OR 'Finished Date' = "" AND 'Created Date' < Iteration1'Iteration Start Date'	From table 32 = sum of 'Effort' when 'Iteration Start Date' < 'Created Date' < 'Iteration End Date' + *Value previous row*

Then, in PowerBI we create a column chart showing the total amount of points that is left and also providing an insight in how much points are being added during sprints. Creating a projected trendline then gives an insight in how many sprints appear to be necessary to finish an Epic. Linking this to Table 46 where 'Feature ID' = 'Parent ID' the overview can also be shown per feature.

Task changes

Last, an overview of tasks that have changes is created using the table containing data on all tasks and filtering it as shown in Table 49. After the table is created, all double rows are deleted.

Table 49 Task changes overview example

Title (Sort)	Changed Date	Reason	State	Remaining Work
String	=Last week	String	String	Integer

This overview shows any changes that have been made the past week and enables anybody to stay up to date regarding planning progress.