Benefit Points in Scrum: A Design Science Study



Project Group: HCC 1011

Supervisor: John Stouby Persson

Spring 2018



Cassiopeia House of Computer Science Selma Lagerlöfsvej 300 9220 Aalborg Ø Tlf. 9940 9940 http://www.cs.aau.dk/

Title: **Benefit Points in Scrum: A Design Science** Abstract: Study This report aims to resolve the problem **Project Period:** of value based prioritization of the Product Backlog for the Product Owner in Scrum, by changing it from cost to value. We take a Design Science research approach, by building and evaluating an artifact, to address our research question. We created an extension to the project management tool, named Jira, which incorporates the theory of Benefit Points. Patrick Grønhøj Benefit Points is a simple way of using value in the prioritization of the Product Backlog. Our solution has been evaluated with three practitioners, and we found that our solution is interesting for the resolution of our research question. In conclusion, our results shows that using Benefit Points for prioritization is beneficial and the integration into the widely used Jira, is fruitful to practitioners.

Spring 2018

Project Group: HCC 1011

Participants: Martin Haaber

Supervisor: John Stouby Persson

Copies: 2

Number of pages: 55

Appendix: A - C

Date of Completion: June 8, 2018

Summary

The foundation of this thesis is the problem of prioritizing the Product Backlog. The Product Backlog is a living artifact, which requires continuous refinement and reprioritization. This thesis takes a Design Science research approach, to build and evaluate a solution, to the problem. A possible solution is suggested and presented, to be used by practitioners, to resolve the problem of prioritizing the Product Backlog.

The solution is based on a theory named Benefit Points, which is used to estimate value of the items in the Product Backlog. Benefit Points exist to shift the focus from cost based prioritization to a more value based focus. Specifically it is concerned with Benefit/Cost ratio and working on the items which produce the most value for the organization and stakeholders. As Benefit Points is used as the theory for this thesis, the problem of value creation is addressed in the discussion, as value is being used to prioritize the Product Backlog.

The solution is an extension to the project management tool, named Jira, and has incorporated the theory of Benefit Points. Benefit Points revolves around Epics, Objectives, and Returns. Epics are high level requirements, Objectives are the business visions in a project, and Returns are the strategic planned returns for the organization/stakeholder. These elements are used in the solution to calculate the balanced Benefit Points, which are used for the prioritization of the Product Backlog.

Our solution is a workable plugin for practitioners to use in their projects, which use value as a prioritization factor. It exist to make the prioritization simpler, to create more value in a project, and justify the choices made by Product Owners. The solution is based on new theory about value based prioritization, and it enables the practitioners to enact the theory in a practical way. By conveying the theory of Benefit Points in a simplistic manner, it should make it easier for practitioners to make use of our plugin and enable a more value based project for them.

Preface

Aalborg University, Jun 8th, 2018

This thesis was written by two 10th semester software engineering students at the Department of Computer Science at Aalborg University. The duration of this semester was from February 2018 to June 2018. It builds upon the knowledge gained from our 9th semester project, which was about the different problems a Product Owner in Scrum can encounter during a project, and focuses on the prioritization of the Product Backlog, to be more value based, by using a theory named Benefit Points.

We would like to say thank you to the practitioners who participated in our evaluations in this project.

We also wish to say thank you to our supervisor, John S. Persson, for his continued assistance and guidance throughout the project.

Page iii of 55

Contents

1	Introduction	vi
2	Theoretical Background2.1Scrum2.2Benefit Points	1 1 2
3	Research Methodology3.1 Design Science Research	7 7
4	Design4.1Existing Technology4.2Jira4.3User Interface4.4Architecture	13 13 16 17 22
5	Implementation5.1Data Structure	26 26 28 29
6	Evaluation6.1IT-House Evaluation6.2WindIT Evaluation	32 32 36
7	Discussion7.1Contribution7.2The Theory of Benefit Points7.3The Artifact	38 38 39 40
8	Conclusion	43
9	Limitations & Future Work9.1Benefit Points for Stories9.2Benefit Points for Bugs9.3Dependency Inclusion9.4Backlog Integration	44 44 45 45

Page iv of 55

Bil	bliography	48
Lis	st of Figures	50
Lis	stings	51
A	Jira	52
B	Balanced Benefit Points	54
С	Testing Our Plugin	55

Chapter 1

Introduction

Scrum is a lightweight process framework, which can be used to manage complex adaptive software product development (Sutherland and Schwaber 2016). Scrum, as well as other agile development methods, was created to accommodate rapid and continuous change (Conboy 2009). However, Scrum is not always adopted by the book, but is assimilated, adopted and re-adjusted to fit the organizational needs (Pries-Heje and Baskerville 2017). Scrum consist of Artifacts, Events, and the Scrum Team (Sutherland and Schwaber 2016), but organizations may have adopted practices from other agile development methods (Pries-Heje and Baskerville 2017).

In Scrum there exist an artifact called Product Backlog, which is a list of features to be developed for the product, and it exists to provide transparency, order, and structure to the development of a product in a project (Sutherland and Schwaber 2016). The Product Owner is responsible for the Product Backlog and he/she must also possess the traits of order and structure to insure the items in the Product Backlog are prioritized, clear, and transparent (Baumgart, Hummel, and Holtenn 2015).

There exists a problem on how to prioritize the Product Backlog, as the Product Backlog is a living artifact, which requires continuous refinement and re-prioritization as development proceeds (Haaber and Grønhøj 2017). Priorities should be based on the wishes of the stakeholder, but also from the interpretation, by the Scrum Team, of the wishes (Kristinsdottir, Larusdottir, and Cajander 2016).

Sprints are a central part of Scrum. The end of a Sprint is also an opportunity to gather feedback from stakeholders, and the wishes of the stakeholders may or may not have changed, and hence priorities may be changed based on this feedback (Kristins-dottir, Larusdottir, and Cajander 2016).

As stakeholders may wish for the world, while they most likely do not require it, there exists a task of defining the necessary requirements for the stakeholders. Usually requirements are based on what the wishes are and the budget available, which provides a reason to add a cost to each requirement. Throughout a project, cost is usually being monitored routinely and used to narrow the scope of a project, to keep it within the budget (Hannay, Benestad, and Strand 2017a). Priorities are often determined by

Page vi of 55

what the cost is to develop a specific item and not on on the value an item would create. (Hannay, Benestad, and Strand 2017a)

The Product Owner must be able to take the vision of the stakeholders and the expertise of the Development Team into account when creating and prioritizing the Product Backlog. The stakeholders' desire is to get value, while the Development Team's wish to provide this value. The Product Owner must provide order and transparency of the Product Backlog to enable the value to be created (Sutherland and Schwaber 2016). As prioritizing items in the Product Backlog are a necessity and a continuous task throughout the lifespan of a project, it possess a problem with how to prioritize the Product Backlog in a clear and optimal way (Haaber and Grønhøj 2017). This report takes a Design Science approach. Design Science is a research methodology which offers guidelines for building and evaluating an artifact within research projects. The first step of Design Science is problem identification and the motivation, which leads to the research question of this report:

• RQ: How can the Product Owner's value based prioritization of the Product Backlog be improved?

This report proceeds as follows. In Chapter 2, we present our theoretical background of Scrum and Benefit Points, to create a knowledge foundation. In Chapter 3, our research methodology is described. Chapter 4 presents our design choices. In Chapter 5, our implementation is presented. Our evaluation is outlined in Chapter 6. Chapter 7 and Chapter 8 presents our discussion and conclusion. Lastly, Chapter 9 describes our limitations and future work.

Chapter 2

Theoretical Background

Section 2.1 presents the Scrum theory and the specific elements used in this report. Section 2.2 presents the theory of Benefit Points, and its elements, which are the main theoretical background for this project.

2.1 Scrum

Scrum is an agile framework, but Scrum is not always followed as the theory is prescribed and the adoption of agile development methods affects how the elements of these are used (Pries-Heje and Baskerville 2017). The goal of an agile development method is to provide agility to an organization, but a standardized agile development method is rather contradictory to the aspect of agility (Conboy 2009). However, elements of Scrum are rather challenging, specifically the role of the Product Owner (Kristinsdottir, Larusdottir, and Cajander 2016), and the tasks around the Product Owner, which can cause a number of problems (Haaber and Grønhøj 2017).

2.1.1 Product Owner

The Product Owner is a singular person, but may represent the desires of a committee (Sutherland and Schwaber 2016). The Product Owner is the most complex role in Scrum (Kristinsdottir, Larusdottir, and Cajander 2016), as he/she should be selfconfident when facing stakeholders and the Development Team (Baumgart, Hummel, and Holtenn 2015), and possess the correct competences, both technical and business, as his work may influence the quality of work performed by the Development Team (Oomen et al. 2017). As Scrum is adopted in numerous ways (Pries-Heje and Baskerville 2017), the role of the Product Owner is inevitable adopted in varies ways too, be it scaling the role of the Product Owner (Paasivaara, Heikkilä, and Lassenius 2012), to having a Project Manager and Scrum Master performing some of the tasks a Product Owner should do (Moe, Dingsøyr, and Kvangardsnes 2009), or how

Page 1 of 55

the prioritization of the Product Backlog is done (Moe, Dingsøyr, and Kvangardsnes 2009).

2.1.2 Product Backlog

The Product Backlog is a singular, ordered list of items, which are used to develop a product (Sutherland and Schwaber 2016). The Product Backlog can be represented from a simple text document to an advanced tool for the Product Owner to prioritize and visualize the items in it (versionone 2017). Adopting the Artifact of the Product Backlog can vary, as an organization can decide to limit or extend the fundamental idea of the Product Backlog, depending on how the organization would adopt Scrum (Pries-Heje and Baskerville 2017). The Product Owner should provide sufficient detail to items in the Product Backlog, and providing too much detail can be an issue (Moe, Dingsøyr, and Kvangardsnes 2009). A Product Backlog also has to be maintained throughout the lifespan of a project, and the Product Owner's allocation of time to maintenance can vary a great amount of the available time the Product Owner has (Sverrisdottir, Ingason, and Jonasson 2014).

2.2 Benefit Points

Benefit Points can be used as a solution to the problematic situation of the Product Backlog and how the Product Owner prioritize the items in the Backlog. The theory of Benefit Points is about adding business value estimates -Benefit Points- to Epics and Stories in the Product Backlog (Hannay, Benestad, and Strand 2017a). There exists other ways to achieve the same result

Benefit Points is a far less complex approach than Scaled Agile Framework's (SAFe) Weighted Shortest Job First. Benefit Points is designed to be intuitive and straightforward to use and maintain throughout a project, with a focus on a clear separation of the cost and benefit. (Hannay, Benestad, and Strand 2017a)

Before assigning Benefit Points for Epics, it is required that an Epic's cost and business value are estimated in relation to the business case of the project. A project may not have a well defined business case, but project objectives will also suffice to estimate the Benefit Points.

To assign Benefit Points to each Epic, one must assess how much an Epic contributes to the objectives of the project. It is recommended to estimate the benefit of an Epic to each objective, however it is possible to estimate the Benefit Points based on all

Page 2 of 55

objectives. It would be beneficial to use a size scheme, which the users are familiar with e.g. the same size scheme used for estimating cost could be fruitful. Onward with this description of Benefit Points, the size scheme used will be the Fibonacci numbers, known from planning poker, to estimate business value of an Epic

An organization usually have long term goals, and linking project objectives to the business return of the strategic goals is beneficial. Figure 2.1 shows how Returns, Objectives, Epics etc. are linked.



Figure 2.1: Blue rows are enterprise level and red are project, grey rows are technical levels. *Figure obtained from (Hannay, Benestad, and Strand 2017a, p. 75)*

Page 3 of 55

We will use our own example based on the theory of Benefit Points. These examples are based on an example logistics organization, which would have, in cooperation with project management, determined the project objectives and returns in the strategic period.

- Objective 1: Reduce the package sorting time by 50 percent.
- Objective 2: Reduce the number of damaged packages by 80 percent.
- Objective 3: Reduce the delivery time by 20 percent.

The Objectives are used in the following equations, and can also be seen on Figure 2.2.

- Return 1: Increased package capacity Value 8 million.
- Return 2: Reduced financial compensation Value 5 million.
- Return 3: Improved corporate relations Value 7 million.

In our example, the total strategically planned return for Returns 1, 2, and 3 is 20 million. Figure 2.2 is an adapted version of the one found in (Hannay, Benestad, and Strand 2017a, p. 78).

Figure 2.2a shows each Epics contribution to the objectives, and the orange shaded fields are just a summation of the assigned Benefit Points. Each Objective is estimated to contribute to the strategic return in varies degrees. Figure 2.2b illustrates how each objective would contribute and Equation (2.1) shows how to calculate TReturn, which is the total return for each objective. SReturn is the strategic return and CReturn is the contribution return for an objective.

$$(SReturn_{1} * CReturn(Obj_{i})) + (SReturn_{2} * CReturn(Obj_{i})) + (SReturn_{3} * CReturn(Obj_{i})) = TReturn(Obj_{i})$$
(2.1)

As an example of TReturn for objective 1, we have: (8*0.35)+(5*0.05)+(7*0.15) = 4.1. Project weight is the proportion relative to the projects total return, and the enterprise weight is the proportion relative to the total strategic return. As an example of the calculation of the project weight we have 4.1/13.55 = 0.30.

As objectives rarely would be equally important, the distribution of the Benefit Points should be balanced to reflect this accordingly. To calculate the balanced Benefit Points, information from Figure 2.2a and Figure 2.2b is required. Figure 2.2c represents the balanced Benefit Points which has been based on the project weight. It is possible to use the enterprise weight instead of the project weight. With the use of

Page 4 of 55

	Return					
Objective	1	2	3		Wei	ight
Contribution Return	8	5	7	Total	Project Enterpri	
1	0.35	0.05	0.15	4.1	0.30	0.21
2	0.15	0.60	0.10	4.9	0.36	0.25
3	0.20	0.10	0.35	4.55	0.34	0.23
Total	0.70	0.75	0.6	13.55	1.00	0.68

(0)	
· /	

	0	Objective	S	
	1	2	3	
Epic	bp's	bp's	bp's	Total bp's
1	1	3	13	17
2	3	5	2	10
3	8	5	21	34
4	21	1	8	30
5	1	2	13	16
6	34	8	5	47
7	5	21	13	39
8	13	21	8	42
Total	86	66	83	235
(a)				

	1	2	3	
Epic	BP's	BP's	BP's	Total
1	0.83	3.86	12.36	17.05
2	2.48	6.44	1.90	10.82
3	6.61	6.44	19.97	33.02
4	17.36	1.29	7.61	26.26
5	0.83	2.58	12.36	15.76
6	28.11	10.30	4.75	43.17
7	4.13	27.04	12.36	43.53
8	10.75	27.04	7.61	45.39
Total	71.11	84.98	78.91	235.00
(c)				

Figure 2.2: Assignment of Benefit Points. (a) Epics' contribution to the Objectives. (b) Objectives' contribution to the Returns. (c) The Balanced Benefit Points. The cells in white are input cells, and the orange shaded cells shows the automatic generated results of the inputs.

the enterprise weight it is possible to see an Epic's or Story's contribution to the entire enterprise return and not only the project return.

To calculate the balanced Benefit Points, a formula presented in (Hannay, Benestad, and Strand 2017a) is displayed in Equation (2.2). E_i is an Epic and Obj_j is an Objective.

$$balancedBP(E_i; Obj_j) = BP(E_i; Obj_j) * weight(Obj_j) * total(BP)/total(BP; Obj_j)$$
(2.2)

To illustrate Equation (2.2), we provide an example for Epic 1 as seen on Figure 2.2c. The example is 1 * 0.30 * 235/86 = 0.82, where we only provide two decimals, hence if you try to replicate this exact example, it will be off by a fraction. After all the balanced Benefit Points have been automatically calculated, the Epics which will provide the most benefit should be considered to be developed first.

Page 5 of 55



Figure 2.3: *The Iron, Agile, and Benefit/Cost triangles (Hannay, Benestad, and Strand 2017b, p. 61)*

The theory of Benefit Points shifts the focus of project management. The iron triangle of project management, as seen on Figure 2.3a, considers quality as the result of balancing scope, schedule, and cost. According to (Hannay, Benestad, and Strand 2017b), the iron triangle has been discussed and most likely been abandoned by the agile community. Figure 2.3b is the agile triangle, which have replaced the iron triangle. The agile triangle is trying to balance benefit, quality, and some constraints which include scope, cost, and schedule.

(Hannay, Benestad, and Strand 2017b) argue that the existing triangles do aim to create a balance between their factors, but given the distinct division of the factors, it may be tempting to prioritize one factor over the others. The benefit/cost triangle, Figure 2.3c, is proposed as Hannay, Benestad, and Strand believes there should not be a distinct division between benefit and cost. Schedule is the remaining constraint as scope is argued to just be a part of benefit. Quality is referred to the technical quality, which include the architecture. The goal of the benefit/cost triangle is to maximize the benefit/cost ratio affected by the schedule and quality. (Hannay, Benestad, and Strand 2017b)

The theory of Benefit Points can be used as another way to estimate the priority of a specific Epic or Story. The use of Benefit Points provides a new dimension of prioritizing Epics and Stories, which provides the most value for the customer.

Page 6 of 55

Chapter 3

Research Methodology

This chapter presents our research approach and the choices made during the research approach.

3.1 Design Science Research

To address our research question, we have chosen to use Design Science as a research method. Design Science has its roots in engineering and other applied sciences. An important foundation is Herbert Simon's conceptualization in *The Sciences of the Artificial* (Simon 1996). March and Smith (1995) picked up Simon's idea. They noted that Design Science needs to undertake two main processes in a cycle: Build and Evaluate. Building is a process of constructing an artifact for a specific purpose whereas evaluation is the process of determining how well the artifact supports a solution.

As a guide for using a Design Science research method, we chose the paper *A Framework for Theory Development in Design Science Research: Multiple Perspectives* (Kuechler and Vaishnavi 2012). It was chosen, as Kuechler and Vaishnavi's paper is an acknowledged way of performing Design Science, and also provides us with a good structure. With the reasoning behind the choice of a Design Science research method, it it fruitful to document how to use it.

According to the guidelines outlined by (Peffers et al. 2007) there is a general procedure consisting of six steps to follow when conducting Design Science research. The six steps of the Design Science Research Method (DSRM) process model can be seen on Figure 3.1



Figure 3.1: *Illustration of steps in Design Science. Figure obtained from (Peffers et al. 2007, p. 54)*

3.1.1 Activity 1: Problem identification and motivation

The initial activity entails defining the specific research problem and justifying the value of a solution as seen on Figure 3.1. The problem definition is used to develop an artifact, which adequately provides a solution. However, it may be fruitful to atomize the conceptualization of a problem, so the solution can encapsulate its complexity.

The justification of the value of a solution accomplishes two things. Firstly, it provides the researcher and the readers of the research with motivation to apply the solution and to accept the results. Secondly, it assists to understand the rationale associated with the researcher's perception of the problem.

This activity requires knowledge of the state of the problem and the importance of its solution. The following activity is to define the Objectives for a solution.

Our Problem Identification and Motivation

The Product Owner has a central and difficult role in a project. He/She is responsible for a number of tasks and artifacts. The main artifact is the Product Backlog which is essential for development of a product during a project. Prioritization of the Product Backlog has been identified as a problem and described in Chapter 1. By creating a solution for this problem, with the theory of Benefit Points, can potentially increase value creation.

Page 8 of 55

3.1.2 Activity 2: Define the objectives for a solution

The objectives of a solution are derived from the definition of the problem and the knowledge of what is possible and feasible to accomplish. The objectives can be either quantitative or qualitative. Quantitative refers to elements, in which a solution would be an improvement compared to current solutions. Qualitative means a new artifact could be expected to assist solutions to yet to be addressed problems.

Our objectives for the solution

The objective is to create a solution which incorporates the theory of Benefit Points, so a Product Owner can use a combination of benefit and cost to prioritize the Product Backlog. There are several requirements a solution should fulfil, which describes the quality of the solution, and not the functionality. (Sommerville 2016).

Our non-functional requirements are as follows:

- **Correctness** The solution should perform all calculations from the theory of Benefit Points correctly. The calculations provide the required information to prioritize the Product Backlog correctly.
- **Robustness** The solution should be able to cope with errors. Since the solution revolves around a central artifact in a project, it needs to be reliable.
- **Usability** The solution should be usable. The Product Owner should with ease be able to interact and extract valuable information related to the prioritization process.

As discovered and documented in our previous semester (Haaber and Grønhøj 2017), we found a number of problems a Product Owner can encounter during a project. The solution needs to resolve the problem of prioritizing the Product Backlog (Baumgart, Hummel, and Holtenn 2015; Sverrisdottir, Ingason, and Jonasson 2014). Furthermore, as the solution incorporates the theory of Benefit Points, the benefit or more precise the value is more visual in a project, which means the problem of Value Creation (Sverrisdottir, Ingason, and Jonasson 2014; Kristinsdottir, Larusdottir, and Cajander 2016) potentially can be resolved.

Our solution can potentially resolve or partly resolve the problem of Organization Resources (Lehtinen et al. 2015; Judy and Krumins-Beens 2008), as it potentially can free up used time in the prioritization process and make the Product Owner more available to the Development Team. A further in-depth description of all the problems can be found in our previous semester report (Haaber and Grønhøj 2017).

3.1.3 Activity 3: Design and development

The creation of an artifact requires determining the desired functionality and necessary architecture, as well as the actual activity of creating the artifact. Artifacts can be widely different and potentially be constructs, models, methods, or instantiations, which each have a broad definition.

A design research artifact is, in concept, any object which is designed with a research contribution incorporated into the design. Traversing from objectives to design and development requires knowledge of theory, which can sustain in a solution.

How we designed and developed the solution

The solution to be designed and developed will have to adhere to design guidelines which will be described in Section 4.3. Sketches will be used to visualize early ideas, which are to be used in the design. A design research artifact will be created based on the theory of Benefit Points incorporated.

3.1.4 Activity 4: Demonstration

Activity 4 revolves around demonstrating the use of the artifact to resolve one or more instances of the problem. Appropriate activities which could be used are experimentation, simulation, case study, and/or proof. Depending on the nature of the artifact, the activity to use to demonstrate may vary, as the most appropriate activity should be used. Having extensive knowledge of how the artifact can be used to resolve the problem is advantageous.

How we demonstrate our solution

The theory of Benefit Points should be demonstrated using a proof of concept solution. The demonstration should involve experiences Scrum Team members. Required knowledge about the theory of Benefit Points should be provided for the demonstration depending on the evaluation strategy used.

3.1.5 Activity 5: Evaluation

Observation and mensuration of how well the artifact assists a solution to the problem, is an activity which involves the comparison of the objectives of a solution to the observed results from the demonstration. The nature of the problem and the artifact is pivotal to the form of the evaluation and the required knowledge of metrics and analysis techniques. A few examples are a comparison of the artifact's functionality with the specified solution objections, results of a satisfaction survey, or client feedback. Conceptually, an evaluation could potentially include one or more appropriate empirical evidence or logical proof.

The researchers can, at the end of the evaluation activity, decide to return to activity 3 to try improve the artifact based on the evaluation, or they can move on to the next activity and restrain any improvements for future projects.

Our evaluation

To be able to observe and measure how well the developed solution resolves the problem, an evaluation strategy must be chosen. We will use a comprehensive framework by Venable, Pries-Heje, and Baskerville for deciding a proper evaluation strategy and method to use (Venable, Pries-Heje, and Baskerville 2012).

We will perform a number of evaluations throughout our project. The evaluations will be in relation to the problem a Product Owner has with prioritization the Product Backlog (Haaber and Grønhøj 2017). Evaluation can also be completed on the non-functional requirements described in Section 3.1.2. Our evaluations are described in Chapter 6.

3.1.6 Activity 6. Communication

Communication is an essential part in sharing ones research with interested parties. Researchers must relay information regarding all aspects, such as the problem and its importance, and the artifact and reasoning behind it and the choices made. Typically researchers use a common structure of an empirical research process to present their findings in a research paper.

Page 11 of 55

How we communicate the findings

In this project, we document our Design Science research through a master thesis report, presenting our process, reasoning, and choices throughout the project. The report will be available through Aalborg University's project database. Our solution will be essential in sharing our take on the theory of Benefit Points. It can be shared through github, or the Atlassian marketplace if the solution meets a number of criteria set by Atlassian.

3.1.7 Our interpretations of the DSRM process model

To start off our approach for this project, we have adopted Figure 3.1 to represent the steps to take for the project. Figure 3.2 represents each activity we will iterate through. Previous research displayed a problematic situation for a Product Owner prioritizing the Product Backlog (Haaber and Grønhøj 2017). Items in the Product Backlog are normally prioritized based on cost, and there exists a need to include different information in the prioritization process. This triggered the development of a solution drawing in the theory of Benefit Points.



Figure 3.2: Our illustration of the six activities in Design Science. Modified version from (Peffers et al. 2007, p. 54)

Chapter 4

Design

This chapter presents our design of the solution to the problematic situation presented in Chapter 1. Section 4.1 describes the existing technologies in order to find the right project management tool.

Section 4.2 explains the project management tool we have chosen, and then in Section 4.3, a user interface is described and designed. Finally the architecture for the solution is explained in Section 4.4.

4.1 Existing Technology

Some applications for project management already exists; those deemed most relevant for this project have been analyzed, and in the end of the section a project management tool is chosen.

4.1.1 VersionOne

VersionOne (VersionOne 2018) is a web based agile management solution that is developed for teams and projects of various sizes. It is a compact platform that delivers in terms of managing and tracking of multiple teams, tasks, and projects. VersionOne can be customized to a business of any shape or size. The Product Backlog in VersionOne can be seen in Figure 4.1.

VersionOne's prioritization is based on a low to high scale, where the ranking of items is up to the Product Owner and could be based on cost. VersionOne do not support plugins so it is not possible to use VersionOne for this project.

Page 13 of 55

Y		My Hor	ne 🗸 sear	ch Q								9 \$ 8 0
٤	Pri No	oject ew Pro	oject	Section > Section ⇒ Section ⇒ Product → Planner	n Release 👌 🔔 Team							II Reports
+	E	Bacl	dog								Ad	d Backlog Item Inline 🗸 🗸
۲	6	1	lick or start typ	ing 🗸								★ My Views
**	Ν	love 1	To Project 🗸				4 4 1-7 of 7 🕨 🕅					\ ~
$^{\odot}$		•	Order †	Title	ID	Owner	Priority	Estimate Pts.	Project	Backlog	Backlog Group	r6
Q	ł		1	🚊 Design UI	B-01050	Administrator	High		New Project		Epic	Edit
ē			2	🚔 Database	B-01045		High		New Project		Epic	Edit 🗸
Ŵ			3	Language Implement totalContributionObjective	B-01047		High		New Project		Stories	Edit 🗸
			4	Benefit Points implementation	B-01046		Medium		New Project		Stories	Edit 🗸
			5	😫 Write Report	B-01048	Administrator	Medium		New Project		Other Stuff	Edit 🗸
			6	🚔 Design Architecture	B-01049	Administrator	Medium		New Project		Epic	Edit 🗸
			7	🚔 Implement UI	B-01051	Administrator	Low		New Project		Stories	Edit 🗸

Figure 4.1: VersionOne Backlog

4.1.2 Pivotal Tracker

Pivotal Tracker (Tracker 2018) is a web based project planning tool for agile software development. Pivotal Tracker can help with visualizing projects in the form of stories or virtual cards, and break down projects into manageable pieces. The Product Backlog can be seen in Figure 4.2.

😵 BenefitPoints		4	Q , Search project ?	WHAT'S NEW HEL	• MARTINHAABER1 •
STORIES ANALYTICS	SETTINGS MEMBERS A				
=	× Current Iteration/Backlog ×10				
~10 ±1 ÅA	▼ 1 + 19 - 25 Feb				0 of 13 points • 🚢
	2	now 1 accepted story			v
+ Add Story	> 👷 💼 As a user I want two step verification for the system (MH)				Finish 🗌
💼 My Work 6	> 🚖 🚪 As a user I want a way to see my data (MH)				Finish
i Current/Backlog	> 🚖 🔔 Design UI (MH)				Finish 🗌
🛞 Icebox	reate database (MH)				Finish
✓ Done	> 🔆 🛫 📮 As a user I want a way to store data (MH)				Finish
• Blocked 1	> 🚖 ፪ Design architecture (MH)				Finish 🗌
😲 Epiks					
🔖 Labels					
③ Project History					

Figure 4.2: Pivotal Tracker Backlog

The tracker part of Pivotal Tracker can show the team's performance with burn down charts. The tracker is intelligent in the way, that it can predict the estimations and the project's completion time, by looking at the teams previous projects and see if the tasks that are assigned to a Sprint can be completed. Pivotal Tracker encourages an agile software development process.

The problem presented in our introduction (Chapter 1) is about how prioritization can be geared more to value than cost. In Pivotal Tracker the prioritization of tasks are based on a scale of 0 to 3 where 0 is a low priority and 3 is high, The prioritization

Page 14 of 55

is cost based by using planning poker. The ranking of tasks on the backlog is based on what the Product Owner finds most relevant. Pivotal Tracker can incorporate both Epics and Stories. With the way the prioritization is accomplished in Pivotal Tracker, it could be possible to make a plugin that incorporates value based prioritization instead of the cost based prioritization.

4.1.3 Jira

Jira is a project management tool (Atlassian 2018b) developed for bug tracking, issue tracking, and project management of software projects. The Jira dashboard has a number of useful features, which are able to handle different issues. Some of the features and issues are: issue types and workflow screens. The Issue page and the backlog in Jira can be customized to match the business processes and they can be seen on Figures 4.3 and A.3

XIRA Dashboards - Projects -	Issues - Create		Search	० इ∜ @- ‡- ∏-
Benefitpoints	Open issues Switch filter -			View all issues and filters
 ✓r Summary DI Issues IB₂ Reports PROJECT SHORTCUTS Task management ideas Working in a project Add link 	Order by Priority 4 - 2 6 EN-5 As a user I want a two step verdification 2 6 EN-2 Design UI 2 6 EN-4 Design architecture 2 6 EN-4 Design architecture 2 6 EN-4 Design architecture 2 6 EN-3 Implement UI	Benefitipoints / BEN-5 As a user I want a two step verdification Image: Comment As a user I want a two step verdification Image: Comment As a user I want a two step verdification Image: Comment As a user I want a two step verdification Image: Comment Image: Comment As a user I want a two step verdification Image: Comment Image: Comment Image: Comment Image: Comment Attachments Image: Comments Image: Comment Image: Comme	People Assignee: Reporter: Votes: Watchers: Dates Created: Updated: HipChat discussion Do you want to disk Connect Dism	1 of 6 ▲ ♥ ℓ ³
		Comment		

Figure 4.3: Jira Issue page

Jira's prioritization of Epics and Stories on the backlog is also based on a lowest to highest scale, but is more granular with low, medium, and high. The prioritization is also based on cost. The ranking of tasks on the backlog is based on the tasks with the highest priority, but the Product Owner can decide the order of the items on the Product Backlog. Since Jira support plugins it could be possible to make a plugin to Jira that try and solve the problem we introduced in the Chapter 1.

4.1.4 Choice of Project Management Tool

It has been decided to create an extension for an existing project management tool, instead of developing a standalone solution. The reasoning behind this decision, is the feasibility of having to develop all the necessary functionalities for a well functioning tool. By using an existing tool as our foundation, we can focus on incorporating the theory of Benefit Points, and it will also provide a clear distinction between our extension and the normal use of a tool.

As we have decided to create an extension, it is not feasible to use VersionOne as our foundation for this report. The choice is between Pivotal Tracker and Jira, where the latter is the most used agile project management tool (versionone 2017). By choosing a tool which is used the most, could provide a closer connection with potential Product Owners to test our extension. Atlassian, the developers of Jira, also provides well described documents on how to develop a plugin for Jira.

4.2 Jira

Jira is a proprietary project management tool that is offered in three packages: Jira Core, Jira Software and Jira Service Desk. In this project we will use Jira Core and Jira Software. (Atlassian 2018b)

Jira Core is the generic project management package, which is the foundation of the use of Jira. It has project, task, and process management. Jira Software is an extension which brings the agile aspects to project management. Project features like Scrum, and Kanban exists in this extension. Since we work with Scrum, Jira Software is essential for our development. (Atlassian 2018b)

4.2.1 How to use Jira

There are several ways to use Jira depending on the type of project being developed. We use the Jira Core as it is the foundation and is required to run Jira as a whole. It also includes a number of libraries which eases development.

Figure A.1 shows the first screen the user is shown when starting Jira. In the top bar are four main menus, Dashboard will take you to an overview of the project and the information of activities pertaining to you as a user. The project menu will show the

Page 16 of 55

projects the user is associated with. The Issue menu shows the recent issues the user has worked on.

Figure A.2 shows the screen for creating a new project. It will shows all the available project types. In this case, it shows the Scrum, as Jira Software is installed. When a project is created, in our case a Scrum project, the user will be presented with the Backlog screen as seen on Figure A.3. When issues have been created for a project, they can be viewed on the issues page, as seen on Figure A.4, which provides a detailed overview of the issues.

4.3 User Interface

When designing a user interface there are a number of elements to be considered to make the correct choices. A user interface's main objective is to present information and functionality to the user. Depending on the context being designed for, there exists a number of guidelines which should be followed when designing the user interface. When designing the user interface for Jira, one should follow the design guidelines provided by Atlassian so it is possible to create a seamless integrated plugin.

4.3.1 Design Guidelines

Atlassian, developers of Jira, have created design guidelines for their products. Design guidelines exists to provide a shared understanding and foundation for implementing user interface elements. There exist guidelines for colors, iconography, typography, and writing style, which are the foundations of the design used in Jira. The various components of Jira also adhere to a number of guidelines which should be followed. The general advice is that new elements added to Jira should aim for a seamless integration in the overall application to create consistency and retain familiarity. (Atlassian 2018a)

Tables

A table is an essential component in Jira. Tables are used to structure large amounts of content in a presentable way to make it easier for users to read and comprehend the content. A table has a number of attributes which changes the look and behaviour of the table.

Page 17 of 55

By default, and in most cases, a table's width is the same as the page. However, at some cases, the width can make content less readable and comprehensible if the content is quite small. Table width should always be adjusted to fit the content it should present.

A table consists of a number of columns and rows based on the information requirements by the users. Information in columns should be presented by importance from left to right, so users can extract the most important information first.

The content in a table should by default be left aligned unless a different alignment assists with comprehension of the content. Numeric data is usually easier to read when it is aligned to the right. Column headers should always be aligned with column content.

Icons

An icon is a visual representation used for directories, files, common action etc. and provide a visual context as well as enhancing usability. An icon should be simple, but bold enough to catch the attention of the user.

When creating an icon, two versions should be designed, one on a 20x20px grid and another on a 16x16px grid. The two sizes are to accommodate both regular and small version of the icon throughout the application.

As a general guideline, it is best to build from base shapes like a square, circle, rectangle, or oval. To obtain a more smooth icon, corners should be curved using the circles available in the initial design grid provided by Atlassian.

4.3.2 Sketches

Sketching is a powerful design technique to illustrate a design concept. A sketch is fairly inexpensive to create compared to the benefit it will produce. We used sketching as a way of visualizing Benefit Points in a design context.

There are several aspects of Benefit Points we wish to present to the user, one of which is the information related to Figure 2.2a and Figure 2.2b. The two tables represent the presentation and input area of Benefit Points to the associated Objective and Epic, and the Contribution associated to a Return and Objective.

There exists a few ways to present a large amount of data, but using a table, as both Figure 2.2a and Figure 2.2b does, is the best way when the user is required to input

Page 18 of 55

Returns Kalis

Edil Del

Rotu

to add Retur

and edit the data on display. The focus should be moved towards explaining Benefit Points in an informative way so a new user can grasp and understand the concept of Benefit Points.

The task at hand is to present Benefit Points in enough details together with the table displaying the data, but also clearly separated from the table of the data so an experienced user's work is not interfered with by the details. Figure 4.4 is a rough layout of where to present relevant information about Benefit Points and the table of Benefit Points. Information should be presented at the top and not take up more than 1/3 of the tab. Any excess information to further deepen the understanding of Benefit Points should be moved to a helpful dialog accessible through the Benefit Points tab. The Contribution and Balanced Benefit Points tab will work in a similar way as the Benefit Points tab to retain consistency.



Figure 4.4: Benefit Points Tab

Figure 4.5: Returns Tab

Returns tab

abo

After having explained and display information relating Benefit Points, Contribution, and Balanced Benefit Points, two central elements remain. Returns and Objectives are essential in the calculation process and a complete comprehension of the two similar, but different concepts is important. Figure 4.5 presents a similar way of separating information from the data. There are however two distinct differences. The first is what and how the data is displayed. All the data about a Return or Objective must be presented, and a clear option to edit or delete each entry. The second difference is how to add, edit, or delete a Return or Objective, which will be completed through dialogues to provide information for the process.

Figure 4.6 and Figure 4.7 represents the dialogues for adding and editing a Return. The two dialogues would be fairly similar as they both should provide enough information to add a Return in the start or to edit a Return at any given time if a Return would change.

Page 19 of 55





Figure 4.6: Dialog for adding a Return

Figure 4.7: Dialog for editing a Return

The dialogues would be similar for Objectives to retain consistency throughout extension. A delete dialog would simply provide a confirmation if one wishes to delete the desired Return or Objective, while notifying which Return or Objective is to be deleted.

With a comprehensive outline and idea about the design, the design process can proceed to the next phase. The next phase is about creating a more detailed design to be implemented.

4.3.3 Design

We used the sketches and the outlined design guidelines to create a more detailed design of our extension. Figure 4.8 presents the design which displays five tabs, each representing essential elements of the theory of Benefit Points. Figure 4.8 more specifically shows information about the Benefit Points and what they are, how to assign them, and the reason to use Benefit Points. It will then present a dynamically generated table based on the amount of Epics and Objectives created for the project. Assigning a Benefit Point is done in the individual cells of the table.

Page 20 of 55

Scrum SCRUM board -	Benefit	Points	Objectives	Contribution	Returns	Balance	ed Benefit Points]	
Backlog Active sprints Releases Mn Reports	Wł Lorem ips adipiscing ultrices au Donec sed dignissim r	What is a Benefit Point? How to assign Benefit Points n ipsum dolor it amet, consecteur Lorem ipsum dolor it amet, consecteur Lorem ipsum dolor it amet, consecteur adiptscing elit. Nulla a felis convallis nisi adiptscing elit. Nulla a felis convallis nisi adiptscing elit. Nulla a felis convallis nisi st autor id quis urna. ultrices autor id quis urna. ultrices autor id quis urna. ultrices autor id quis urna. sim massa. dignistim massa. dignistim massa. dignistim massa.				Why use Benefit Points? orem ipsum dolor sit amet, consectetur dipiscing elit. Nulla a felis convallis nisi ltrices auctor id quis urna. Yones ced vestibulum dolor. Integer ut ignissim massa.			
ව= issues දු3 Components	Epic	Objec	ctive 1	Objective 2	Object	ive 3	Objective 4	Objective 5	Total
Benefit Points	SCRUM-1	3	3						3
PROJECT SHORTCUTS	SCRUM-2	8	3						8
Add a link to useful information for	SCRUM-3			5					5
Add link	SCRUM-4						13		13
	Total	1	1	5	0		13	0	29

Figure 4.8: Benefit Points Tab

The tabs for Contribution and Balanced Benefit Points will follow the same layout to retain consistency. The essential part of the balanced Benefit Points tab will be providing direct link to each individual Epic for accessibility, which also can be seen on Figure 4.8, and the ability to sort the table to create an improved overview.

The last two tabs, Returns and Objectives, will also provide information regarding what a Return and an Objective is, and how to determine one. Figure 4.9 shows the Returns tab, which has a similar information area at the top, as the other tabs, for consistency. It however also provide a create button as it should be possible to create a new Return. It will open a dialog to enter information about the Return and update the list of Returns. A list of all Returns will be presented to the user with all relevant information regarding a Return. Options to edit and delete a Return exists and will prompt a dialog for each action.

SCRUM board -	Benefit Po	oints Objectives	Contribution	Returns	Balanced Benefit Poi	nts	
■ Backlog Active sprints ▲ Releases Image: Reports	What Lorem ipsum adipiscing elit. ultrices auctor Donec sed ves dignissim mass	tt is a Return? dolor sit amet, consecte Nulla a felis convallis n id quis urna. ibulum dolor. Integer u ta.	tur Lore isi adipi ultric at Donn digni	ow to deter m ipsum dolo scing elit. Null es auctor id q ec sed vestibul ssim massa.	mine a Return r sit amet, consectetur a a felis convallis nisi is urna. um dolor. Integer ut	C	freate Return
DE Issues Components	Return	Description C	Contribution Value				
Benefit Points	1	Effectiveness	80 Million			Ed	it Delete
PROJECT SHORTCUTS	2	Less Work	800 Thousand			Ed	åt Delete
Add a link to useful information for	3	TBD	10 Thousand			Ed	it Delete
+ Add link	4	TBD	28 Million			Ed	lit Delete

Figure 4.9: Returns Tab

The Objectives tab will use the same layout as the Returns tab as displayed on Figure 4.9 and make use of the same options for creating, editing, and deleting an Objective.

Lastly, we also decided to add the information of balanced Benefit Points on the individual Epics. It will show the balanced Benefit Points for each Objective, and it can be seen on Figure 4.10.

Details				People		
Туре:	 Episk 	Status:	TO DO (View Workflow)	Assignee:		Unassigned
Priority:	↑ Medium	Resolution:	Unresolved			Assign to me
Labels:	None			Reporter:		😡 admin
Navn på episk:	Exception			Votes:		0
				Watchers:		 Stop watching this issue
Description						
Click to add descrip	ntion			Dates		
				Created:		3 days ago
Attachments				Updated:		3 days ago
	(j) [)rop files to attach, or browse.				
				Agile		
				View on Boar	rd	
Issues in Epic				+ Balanced Be	enefit Po	int
There are no issues	s in this epic.			Objec	tive	Balanced BP
Activity				1	·	8,25
All Comme	nts Work Log History Ad	tivity		3	2	25.45
These are no comm	ente vet en this issue			5	3	13.37
mere are no comm	ients yet on this issue.					
					1	
Comment						

Figure 4.10: Balanced Benefit Points on the Issue Page

4.4 Architecture

4.4.1 3-Layered Architecture

In this project we used a 3-layered architecture for our solution. A 3-layered architecture separates the presentation, application, and data layer, which benefits scalability and performance. In our project, the 3 layers are named User Interface, Model Component, and Data Structure. Figure 4.11 displays an overview of the 3-layered architecture of this project.



Figure 4.11: 3-Layered Architecture (Adopted from (Jinfonet.net 2018))

The User Interface layer is where information is displayed, and all interactions between the solution and the user happens. These interactions could be the user entering information, which the solution requires to calculate the balanced Benefit Points. The Model Component layer handles all the business logic of the solution. The business logic involves the calculation of the balanced Benefit Points, information to send to the User Interface layer and to the Data Structure layer. The Data Structure layer is how data is stored, which involves Benefit Points, Contributions, Returns, and Objectives.

4.4.2 Model Component

To get an overview of the problem area, a model component was created. Figure 4.12 illustrates the model component and the classes which are a part of it. The Epic class is a child of the Issue class, and it inherits the properties of its parent. Benefit Point is an aggregation of Epic, as a Benefit Point can not exist without an Epic. Likewise, an Objective has the same kind of aggregation with a Benefit Point, as an Epic has. This means that a Benefit Point can not exist without both an Epic and an Objective.

Page 23 of 55



Figure 4.12: Model Component of the problem area.

An Objective also has a relation with a Contribution. A Contribution is an aggregation of an Objective and a Return. This means that a Contribution can not exist without both an Objective and a Return.

Page 24 of 55

4.4.3 Data Structure

Atlassian recommends using Active Objects as it is a new object relational mapping layer for Jira. Active Objects enables easier, faster, and more scalable data access and storage compared to the existing options of Bandana and PluginSettings APIs offered by Jira. Active Objects is recommended to use for plugin data storage and acts as a real database (Atlassian 2016). it was decided to follow that recommendation.

To provide an overview of our data structure for the plugin, an Entity Relationship (ER) diagram was constructed, which can be seen on Figure 4.13. The ER diagram consists of four tables, or Active Objects, which are BenefitPoint, Contribution, Objective, and Return.



Figure 4.13: Entity Relationship Diagram

The relationships in the ER diagram has been specified according to the theory of Benefit Points (Hannay, Benestad, and Strand 2017a). The theory states that the Objectives has a zero to many relationship with BenefitPoint and Contribution. Return has a zero to many relationship with Contribution, as a Return can have zero Contributions or many Contributions depending on amount of Objectives. This structure is simple but effective and helps to simplify the overall architecture of the system.

Page 25 of 55

Chapter 5

Implementation

This chapter shows and explain how the system has been implemented. Section 5.1 describes the data structure and communication to the database. Section 5.2 explains the important calculations for balanced Benefit Points. In Section 5.3 the user interface is designed in Section 4.3 is implemented.

5.1 Data Structure

Based on the the Data Structure outlined in Section 4.4.3, we have implemented a way to save Active Objects for Benefit Points, Contributions, Returns, and Objectives. Listing 5.1 shows the method required to add a new Benefit Point.

```
public BenefitPoint add(int value, long epicID, long objectiveID,
     long projectID)
  {
2
3
      final BenefitPoint benefitPoint =
          ao.create(BenefitPoint.class);
4
      benefitPoint.setValue(value);
5
      benefitPoint.setEpicID(epicID);
6
      benefitPoint.setObjectiveID(objectiveID);
7
      benefitPoint.setProjectID(projectID);
8
      benefitPoint.save();
9
      return benefitPoint;
  }
10
```

Listing 5.1: BenefitPoint data

The method starts out on line 3, where a new object of the type BenefitPoint is instantiated as an Active Object based on the BenefitPoint class. Line 4, 5, 6, and 7 is simply invoking the set methods associated with the BenefitPoint class. Lastly, on line 8, the information has to be saved before it returns. The newly created Active Object is stored within Jira's internal database of Active Objects. The add method, as well as other relevant methods are invoked through a BenefitPointService class.

Page 26 of 55

The implementation for Contributions, Returns, and Objectives follow the same pattern, but with slight differences on the properties assigned and methods available to invoke.

5.1.1 Servlet

To retrieve input from the user, we use servlets to communicate with the server. When an action is performed, e.g. data is submitted through a form, the servlet will process the request through a method called doPost, which takes a HttpServletRequest and a HttpServletResponse as input. The doPost method processes the requests sent from the client and returns a response to the client. Listing 5.2 shows the doPost method for our BenefitPointServlet. When adding or editing a Benefit Point, it goes through the doPost method.

```
protected void doPost(HttpServletRequest req, HttpServletResponse
      res) throws ServletException, IOException{
2
3
    List<BenefitPoint> benefitPoints = benefitPointService.all();
4
       boolean added = false;
5
       if(!benefitPoints.isEmpty()){
6
           for (BenefitPoint benefitPoint: benefitPoints) {
 7
               if(benefitPoint.getissueID() ==
                  Long.parseLong(req.getParameter(
8
                   "returnIDBPInput")) &&
                       benefitPoint.getObjectiveID() ==
                       Long.parseLong(req.getParameter(
                   "objectiveIDBPInput"))){
9
10
                   benefitPointService.update(benefitPoint.getID(),
11
                   Integer.parseInt(req.getParameter(
12
                   "benefitPointInput")));
13
                   added = true;
               }
14
           }
15
16
           if (added){
17
               res.sendRedirect(req.getHeader("Referer"));
           } else {
18
19
               addBenefitPoint(req, res);
               res.sendRedirect(req.getHeader("Referer"));
20
           }
21
22
       } else {
23
           addBenefitPoint(req, res);
24
           res.sendRedirect(req.getHeader("Referer"));
      }
25
26 }
```

Listing 5.2: Benefit Point Servlet

Page 27 of 55

As two actions, add or update, can happen when invoking the doPost method, we perform a check to see if a Benefit Point with the same EpicID and ObjectiveID already exists. To do this, we require a list of all the Benefit Points, and if the list is empty, we simply just add the new Benefit Point. The iteration spanning line 6 to line 15 checks if a Benefit Point already exists. If a Benefit Point already exists the BenefitPointService update method is invoked on line 10 to 12. On line 16, a statement checks if a Benefit Point has been updated, and if not then the Benefit Point will be added on line 19.

5.2 Benefit Points Calculations

As prescribed by the theory of Benefit Points (Hannay, Benestad, and Strand 2017a), there are a number of calculations to perform. Some calculations are simple summations to prepare for the essential calculations, e.g. for balanced Benefit Points. We presented Equation (2.1), which calculates the total Contribution per Objective, and the implemented calculation can be seen on Listing 5.3.

```
public static double contribution(List<Contribution> conlist,
 1
      List<Return> retlist, int objectiveID){
2
3
      double total = 0;
 4
      for (Contribution contribution: conlist) {
           if (contribution.getObjectiveID() == objectiveID){
5
               for (Return return1: retlist) {
 6
 7
                    if (contribution.getReturnID() == return1.getID()
                       ){
                        total += (contribution.getValue() *
8
                           return1.getContribution()) / 100;
9
                   }
10
               }
           }
11
       }
12
13
       return total;
14
  }
```

Listing 5.3: Total Contribution per Objective calculation

The method requires three arguments, which are a list of Contributions, a list of Returns, and an Objective ID. On line 4, it will iterate through all Contributions to find the Contributions which belongs to the specified Objective. On line 6, it will iterate through all Returns and perform a singular calculation, which the result will be added to a total, which is returned at the end.

Page 28 of 55

Listing 5.4: Single balaned Benefit Point calculation

The main calculation to be performed is shown on Equation (2.2), which is for calculating a balanced Benefit Point. It requires a Benefit Point, a weight, total amount of Benefit Points, and the total amount of Benefit Points associated to an Objective. As shown on Listing 5.4, the method takes the mentioned four variables as arguments and returns the balanced Benefit Point.

5.3 User Interface

In our implementation of the user interface, we followed the created design from Section 4.3.3, which follows the design guidelines presented in Section 4.3.1. Figure 5.1 shows how the Benefit Points tab has been implemented based on Figure 4.8.

Benefit Points	Objectives	Contributions	Returns	Balanced Benefit Points						
What is a Balanced Benefit Point?				How is a Baland	ced Benefit Point calcul	Why use Balanced Benefit Points?				
A Balanced Benefit Point is the calculated value of a Benefit Point, taking into various information into account to create balance.			A Balanced Weight(PW Points for t The formul	Benefit Point is calculate), the Total amount of Ber he specific Objective asso a is BP*PW*TotalBP/Total	d by taking a single Benefit P hefit Points(TotalBP), and the ciated with the single Benefit BP(Obj) = Balanced BP.	Balanced Benefit Points provides the balanced benefit for each Epic. They have taken into account the Contributions distribution and balanced the individual Epics value. Balanced Benefit Point exists to prioritize Epics to create the most value at all times.				
	Epic		Ob Spread ti Bene	jective 33 he knowledge of Re fit Points	Objective 34 educe the learning curve of Benefit Points	Objective 35 f Provide Improved Prioritization	Objective 36 Increase value creation	Total		
Benefit	Points Templa	te		5.61	2.72	3.73	2.14	14.2		
Visualizati	on of value cre	ated		9.36	1.63	1.24	3.42	15.65		
Colour Co	des implement	ation		1.87	2.72	9.93	1.28	15.81		
Learning C	Curve Optimiza	tions		14.97	7.07	6.21	1.28	29.53		
Back	log Integration			9.36	2.72	16.14	2.14	30.36		
Story Poir	nts Implementa	ation		9.36	0.54	26.08	3.42	39.4		
Clarifications	s of size schen	ne etc.		14.97	7.07	6.21	2.14	30.39		
Improvemen	ts to existing s	olution		14.97	11.42	16.14	2.14	44.67		
Total			80.47	35.89	85.68	17.95	220.0			

Figure 5.1: Implemented version of the Benefit Points Tab

The most noticeable change is the addition of a button in the bottom right corner. This button, when clicked, save all the changes made to the table of Benefit Points. Figure 5.2 shows the implemented version of the Returns tab. It displays three sections of text areas at the top, to provide useful information about Returns.

	Benefit Points	Objectives	Contributions	Returns	Balanced Benefit P	oints				
What is a Return? A Return is the representation of the planned return of a project. A Return consists of an estimated overall return. An example of a Return could be the optimization of a storage facility, which would have an estimated overall return of 4 million.				How to dete	rmine a	Create a Return				
				A Return is determined between the strategic management and the project's management.			To create a new Return, click the Create Return button and fill in the fields. Create Return			
Return Description						Contribution				
	33		Increased knowl	edge of Ber	efit Points 6.0		6.0	Edit	Delete	
	34		Reduction	of learning	urve 3.0		3.0	Edit	Delete	
	35		Improve	d Prioritizati	ion		10.0	Edit	Delete	

Figure 5.2: Implemented version of the Returns Tab

The main difference between the implemented version, Figure 5.2, and the designed version, Figure 4.9, is the text added above the button to create a new Return as seen at the top right of Figure 5.2.

We also made a slight addition to the individual issue page in Jira, specific for the issues of type Epic. Figure 5.3 presents the implementation of the design, which can be seen on Figure 4.10.

D	ates		
(Created:	5 days ago	
	Updated:	5 days ago	
A	gile		
,	View on Board		
В	alaned Benefit Poin	ts	
	Objectiv	Balanced Benefit	
			Points
	Spread the knowled Points	dge of Benefit	Points 5.61
	Spread the knowled Points Reduce the learn Benefit Po	dge of Benefit ing curve of pints	Points 5.61 2.72
	Spread the knowler Points Reduce the learn Benefit Po Provide Improved	dge of Benefit ing curve of pints Prioritization	Points 5.61 2.72 3.73
	Spread the knowled Points Reduce the learn Benefit Po Provide Improved Increase value	dge of Benefit ing curve of pints Prioritization e creation	Points 5.61 2.72 3.73 2.14

Figure 5.3: Implemented version of the Balanced Benefit Points on the issue page

There is a minor addition of the total value, which can be seen on the bottom right of Figure 5.3. This minor change was made, as it makes sense to present the individual values, but also the summation of these values, as they are available and used in the prioritization process.

5.3.1 Improvements based on evaluation

In Chapter 6, we describe the evaluation of our solution. The evaluation provided feedback, which made us make improvements to our solution. Figure 5.1 and Figure 5.3 represents the final version of the solution, which do include the improvements. Figure 5.1 shows that an Epic uses its summary as identification instead of its key. It also shows that the description of an Objective is added to better identify the individual Objectives. These identification changes has been applied on the Contributions and Balanced Benefit Points tabs as well. Figure 5.3 shows that the table on the individual issues page also has had the description of the Objectives added.

Page 31 of 55

Chapter 6

Evaluation

When conducting Design Science research, evaluation is a central and essential activity. We used the Design Science Research Evaluation Strategy Framework (Venable, Pries-Heje, and Baskerville 2012) as a guide to find the relevant evaluation strategy to use for our project.

We identified, using the Four-Step method, that we should observe our solution in an artificial environment. This means we will be using example information with our solution during the two evaluations.

6.1 IT-House Evaluation

We set up an evaluation of Benefit Points and our plugin, with two representatives from IT-House (IT-House is a made up name to keep the company anonymous). IT-House is a software company located in Aalborg and they have approximately 60 employees. The two representatives were a junior consultant and a software developer with a few years of experience, which both has been a part of the recently switch to using Scrum and Jira in IT-House.

We asked if they wanted the theory our solution is based upon, before our evaluation began. When our evaluation started, we asked to what extent they knew the theory, and they had only briefly gone through the theory. This meant that our approach for the evaluation should focus on explaining the theory of Benefit Points while we presented our plugin. Through the entire evaluation, we noted down all the feedback to be processed after the evaluation.

To create a realistic example, that the evaluatees could relate to, we decided to use an example the evaluatees came up with. They suggested a project for storage optimization and accounting in IT-House. We created four Epics named Sale, Purchase, Tasklet Factory, and Document Capture. After this we also created two Objectives, one for storage optimization of 50%, and the second being reduction of manual pro-

Page 32 of 55

cesses by 50%. Two Returns were identified, calculated, and created. They were reduction in accounting, which had a contribution of 420.000 danish kroner, and storage optimization with a contribution of 2.100.000 danish kroner.

With the Epics, Objectives, and Returns defined, we could proceed to the next phase of the evaluation by assigning Benefit Points and Contributions. It was decided to use a size scheme of 1-100 for assigning Benefit Points, and we let the evaluatees assign the points with us as guidance. After assigning the Benefit Points, we assigned Contributions, more specifically how much an Objective would contribute to a Return.

As all necessary information had been entered, we could present the balanced Benefit Points, and the evaluatees could see which Epic would bring most value. We then transitioned to an open discussion about Benefit Points and our solution in relation to IT-House.

The first topic of our discussion was the plugin we had created. We discussed how the plugin was to use, how it looked, and improvements which could be made. The second topic we discussed, was if Benefit Points and our plugin could be useful in IT-House. Lastly, we asked how their experience had been with the evaluation of Benefit Points and our plugin and which steps they suggest we take next.

Findings

Our focus for the findings have been general feedback towards the theory of Benefit Points and especially our plugin. The feedback provided made it evident that overall, the idea of Benefit Points and an integration of Benefit Points into a plugin made it tangible as the developer mentioned.

"I really like the idea of using value as a way of thinking, and making it available through a plugin to Jira makes it tangible to understand and work with. We are practitioners after all." - Software Developer

To dive deeper into our findings, we discovered issues relating to the learning curve of Benefit Points, usability of our plugin, and organizational thinking of projects.

The learning curve of Benefit Points was a bit steep at first. The evaluatees had only brief read the theory of Benefit Points we sent them, which required us to explain the theory to them. After a short while and throughout the evaluation, the evaluatees gained a basic understanding of the theory. The theory may be written to practitioners and be fairly simple, but conveying it through a plugin, by condensing the theory is a difficult task. The software developer mentioned in relation to the layout and information provided.

"I really like that the information is separated into multiple tabs. It provides a decent overview, while still keeping the information together.

Page 33 of 55

The information seems to provide a basic introduction to how you use it as well." - Software Developer

The separation of Benefit Points, Objectives, Contributions, Returns, and Balanced Benefit Points made the overall feel of plugin well balanced to use. There were however a number of issues which arose during the evaluation. The main issue which arose, was the identification of Epics, Objectives, and Returns when the user is on the Benefit Points, Contributions, and Balanced Benefit Points tabs. We identified and listed Objectives and Returns by their ID values, which made it difficult to differentiate between these. Epics were also listed by their key value, which also made it difficult to differentiate between them. Epics also provided a link to their designated page, where information could be changed, e.g. priority of the Epic, and we asked the evaluatees about this functionality to which the developer answered.

"I think it is a really useful functionality to quickly access an individual epic at anytime." - Software Developer

Another issue was the need of changing page to change priority of an Epic, which requires a lot of going back and forth. Lastly, when the evaluatees had to assign Benefit Points, an issue arose with understanding which size scheme, that could be used. It was unclear that any size scheme was possible to use when assigning Benefit Points.

The evaluatees presented a few ideas to address the issues. The first issue of identification, could be resolved by providing the description of an Objective and Return, as well for Epics. The second issue of page changing for the priority, could be integrating the balanced Benefit Points more directly into the Product Backlog of Jira. This could be done with the actual balanced Benefit Points, or with a form of colour coding with green, yellow, and red. The junior consultant suggested:

"It would be really cool if you made a template like the Scrum template, but for value instead. It could help with changing the thinking of an organization and if we were to restructure our processes and mindset." -Junior Consultant

We addressed the question about the usefulness of the theory and our plugin for IT-House. They were positive that it could be beneficial to use Benefit Points to estimate value and prioritize what to work on next, and by using our plugin they believed it was easier to start using. However, they had a few issues as to the use of Benefit Points. It would first and foremost require IT-House to include value thinking, which requires more planning in the start of a project and a template for Benefit Points could potentially assist in resolving this issue. They also mentioned the mentality of their typical client, which desired to know the exact cost of the solution they were to make. However, they agreed that it would be useful to present the information, so a

Page 34 of 55

client could see what they paid for and how the money was spent, in turn showing where the value was created for the client.

The last step we covered in the evaluation was the next steps to take. They suggested to implement Benefit Points for stories as we already covered Benefit Points for Epics. We agreed upon this suggestion, as the authors of Benefit Points wrote a paper explaining Benefit Points for stories (Hannay, Benestad, and Strand 2017b). They asked if we could provide this theory to them, which we did. We also shared a spreadsheet with example calculations, as per their request, so they could work with the calculations themselves.

"We really like the theory and it sounds quite smart to use. It would take time to get used to, and if value for stories could be added, the plugin would be even more useful." - Junior Consultant

6.1.1 Summary

To provide an overview of the issues and ideas, that was found during the evaluation with IT-House, we have made an ordered list. The ordered list has been prioritized by using our own solution to determine which item would provide the most value. The balanced Benefit Points can be seen on Figure B.1. The issue about replacing ID/Key identifiers for Objectives, Returns, and Epics with Descriptions/Names will be addressed in this project, while the rest can be implemented/improved in the future.

- Improvements to existing solution (Replace ID/Key identifiers for Objectives etc.)
- Story Points Implementation
- Clarifications to Size Scheme etc.
- Backlog Integration
- Learning Curve Optimizations
- Colour Codes Implementation
- Visualization of value created
- Benefit Points Template

6.2 WindIT Evaluation

Before performing a second evaluation with WindIT, we made some improvements to our solution, based on the first evaluation, which can be seen in Section 5.3.1. By taking this approach, we iterated back to activity three of our Design Science approach.

We set up a second evaluation of our solution and Benefit Points, with one representative from WindIT. WindIT is a software engineering company located in southern Jutland, with approximately 20 employees in the IT department. The representative from WindIT, was a Project Manager with multiple years of experience in software development, and WindIt uses Scrum on a daily basis.

Before the evaluation, we suggested to send the theory of Benefit Points (Hannay, Benestad, and Strand 2017a) to the evaluatee, but he informed us, that he would not have available time to read it before our scheduled evaluation. Our approach for the evaluation, like our first evaluation, should focus on explaining the theory of Benefit Points, while we present our solution. Throughout the entirety of the evaluation, we noted down the feedback.

To make the evaluation more relevant for the evaluatee, we decided to create, with the evaluatee, a realistic example, which the evaluatee could relate to. He suggested a project for a wind farm. We created a couple of Epics, two Objectives, and two Returns were identified, calculated, and created as well.

After all the necessary information had been entered, the balanced Benefit Points were presented to the evaluatee, so he could see which Epic provided the most value. The evaluation transitioned into a discussion about Benefit Points and our solution.

Findings

The focus for this evaluation has not only been feedback on the theory of Benefit Points and our solution, but also for us to relate the feedback to our evaluation with IT-House. The feedback provided made it clear that the idea of Benefit Points, and the integration of Benefit Points into a plugin, made the theory more tangible for the evaluatee. We also identified a number of the issues the IT-House evaluation displayed.

The first issue, which was discovered, was the learning curve of Benefit Points. The same issue also occurred in the IT-House evaluation. The learning curve of Benefit Points was a bit steep, but the elements which was difficult to understand was not the same in the two evaluations. In the first evaluation, Epics, Objectives, Returns, and Contributions took time to comprehend, while in the second evaluation, it took the Project Manager some time to understand Benefit Points. The reason was, that

Page 36 of 55

the Project Manager is accustomed to working with Returns and Objectives on a daily basis.

The Project Manager also commented on the fact, that Benefit Points was condensed into a single page with tabs in Jira. This was a nice information separation, instead of spreading the information all over Jira.

"It is nice that you have separated the information into a single page in Jira instead of spreading it over the entire Project." - Project Manager

However, even though the evaluatee liked the condensed information on a single page, it was difficult to see the balanced Benefit Points on the Product Backlog. He suggested that we implement a place to view the balanced Benefit Points on the Product Backlog to make them more accessible.

In the first evaluation, we discovered an issue with understanding the size scheme to use. In this evaluation, we spent more time to explain, that any size scheme was possible to use, which lead to no issues in regards to understanding the size scheme.

The Project Manager mentioned, that it would be difficult to adopt the theory and solution internally in the organization. The leaders of the organization wants to know what a feature/project costs and what value it will create. This issue is similar to the issue of displaying value for the customer, which IT-House has. The discussion then moved on to talk about WindIT using the solution. The evaluatee said:

"If we should use the solution in our business, it would require a change of mindset in the company to think more value than cost." - Project Manager

The issue about changing the mindset of the organization, was also discovered in the first evaluation. This could relate to theories of change and resistance, as organizations and the individuals of said organizations are not necessarily good at change (Rose 2012, p. 17).

At the end of the evaluation, the evaluatee asked if he could receive a copy of the report, articles about the theory of Benefit Points, and a compiled version of our plugin, after handing in our project. The evaluatee said:

"I like the idea and I can see that it could be used to create more value for the Product Owner. but before we can use it I need to use some more time assessing it" - Project Manager

As a conclusion to the evaluation with WindIT, we did not identify any additional issues, compared to the first evaluation. It would be beneficial to address the identified issues before conducting further evaluations.

Chapter 7

Discussion

In this project, we used Kuechler and Vaishnavi's guide for using a Design Science research method (Kuechler and Vaishnavi 2012), and the guidelines by (Peffers et al. 2007) was used to provide us with a structure for the duration of the project. The structure meant that we could focus on the two main processes of Design Science research, Build and Evaluate (March and Smith 1995). Constructing a plugin for Jira, which incorporates the theory of Benefit Points, and evaluating the constructed plugin, as well as the theory.

In this chapter we discuss some of the choices made for this project. Section 7.1 discussed what this project has contributed with, following that, Section 7.2 covers the theory of Benefit Points. Section 7.3 introduces the choices made about the artifact, based on our research question presented in Chapter 1.

7.1 Contribution

In this project, we transition the theory of Benefit Points into a fully functional plugin for Jira. To our knowledge, no one has attempted to create a solution which incorporate the theory of Benefit Points. A reason could be that the theory of Benefit Points is relatively new, being published May - June 2017 (Hannay, Benestad, and Strand 2017a). Our contribution is twofold, one being supplementary knowledge, and the second being an implemented version of Benefit Points.

Hannay, Benestad, and Strand share their experiences with the approach for large enterprises and the public sector in Norway. According to the authors, their approach provided a better understanding of project objectives and a clearer perception of the value to be expected. (Hannay, Benestad, and Strand 2017a)

We evaluated the theory of Benefit Points along with our plugin for Jira. The evaluations, with three practitioners, resulted in valuable feedback. The evaluations showed that the theory of Benefit Points not only were useful for the public sector and large

Page 38 of 55

enterprises, but also for small to medium sized enterprises (Chapter 6). Our discoveries are supplementary to the initial conclusion presented by the authors of Benefit Points, as we show that Benefit Points can be used on a smaller scale (Hannay, Benestad, and Strand 2017a), and how this can be done with the aid of a Jira plugin.

We created a plugin for Jira, to take the theory of Benefit Points and make it more useful for practitioners. Our plugin tries to convey the theory in a simplistic manner. In Chapter 1, we introduced the problematic situation of prioritization the Product Backlog for the Product Owner. We have created a proof of concept, which potentially can resolve the problematic situation.

7.2 The Theory of Benefit Points

We introduced the theory of Benefit Points in Section 2.2 as the theory used for our solution to the problem presented in Chapter 1. Benefit Points is a simplistic and straightforward approach to assist in estimating value in a project. The intuitiveness and simplicity of the approach made it understandable in a short amount of time.

The theory of Benefit Points is relatively new, and more information about this approach is being published as time pass. At the moment of this report, only two articles exists that cover Benefit Points. The first article presents the approach at a high level and with a focus on Epics (Hannay, Benestad, and Strand 2017a). The second article dives into the relation of Benefit Points to Stories and cost (Hannay, Benestad, and Strand 2017b). While the available information regarding Benefit Points is sufficient, it is still changing and information regarding other aspects of prioritizing the Product Backlog, with Benefit Points, are being explored by the authors of Benefit Points (Hannay, Benestad, and Strand 2017a).

Benefit Points is not the only way of estimating value for a project. Scaled Agile Framework (SAFe), which is a framework, has an approach for prioritizing the Product Backlog, which is named Weighted Shortest Job First (WSJF). WSJF is calculated by Cost of Delay (CoD) with Job size. CoD is a summation of User-business value, Time criticality, and Risk reduction-opportunity enablement value. Each of these values are based on a number of other parameters. (Inc. 2018)

Both Benefit Points and WSJF have value at the center for prioritizing the Product Backlog. WSJF is though far more complex, as it involves a multitude of parameters compared to Benefit Points. Complexity and details may provide a more accurate estimation, but it also requires more resources to understand and use in the correct way, which may conflict with the problem about available Organization Resources (Lehtinen et al. 2015; Judy and Krumins-Beens 2008). Hence it would be wise to look

Page 39 of 55

at the value each approach would provide for a specific organization and its area of business. Based on our evaluations (Chapter 6), we also discovered that Benefit Points could be used to represent a customer, by presenting the strategic returns and project objectives. Customer representation is a problem about how to represent ones customers (Kristinsdottir, Larusdottir, and Cajander 2016; Sverrisdottir, Ingason, and Jonasson 2014; Baumgart, Hummel, and Holtenn 2015), and by introducing value as a representation factor, it could provide clarity to both the customer and the Scrum Team.

7.3 The Artifact

The main objective of this project was to create a solution, which addressed the problem on how to prioritize the Product Backlog, as the Product Backlog is a living artifact, which requires continuous refinement and re-prioritization during a project (Haaber and Grønhøj 2017). In this endeavour we had several requirements our solution should fulfill. We listed three non-functional requirements in Section 3.1.2, which are Correctness, Robustness, and Usability.

Correctness is pivotal to our solution. If the calculations of the balanced Benefit Points is calculated incorrectly, the information would be obsolete to be used by the Product Owner, and potentially lead to poor prioritization. Throughout the project period, we have performed calculations on paper and on spreadsheets to compare with our solution to ensure the calculations were correct. We have also, as a final approval, used the presented example in the theory of Benefit Points to verify the calculations (Hannay, Benestad, and Strand 2017a).

Robustness is important, as the solution should be reliable to use and handle errors in the proper way. Jira handle general errors, but our plugin also present information to the user if an action or input is not valid. Behind the scenes, we also have several safety checks, e.g. on division by zero, type check, out of bounce, and null check. All these measures are to ensure a reliable experience with our solution.

Usability is a difficult parameter to measure. We have evaluated our solution with three practitioners (Chapter 6). The evaluations showed us that our solution is, overall, usable and that it was possible to perform the necessary interactions. It was also clear that some areas of our solution was not as usable as we had hoped. One of the issues was the naming of Epics, Objectives, and Returns on the Benefit Points, Contributions, and balanced Benefit Points tabs, which made it difficult to recall the exact relation of each when assigning Benefit Points and Contributions. This was resolved before the second evaluation. Another main aspect was the ability to view the

Page 40 of 55

balanced Benefit Points, at the same time as your prioritize an item in the Product Backlog. This lead to decreased usability of our solution.

We also have two minor objectives, which our solution could potentially affect. These two objectives are Value Creation (Sverrisdottir, Ingason, and Jonasson 2014; Kristins-dottir, Larusdottir, and Cajander 2016), and Organization Resources (Lehtinen et al. 2015; Judy and Krumins-Beens 2008). These two have not been a main focus for our project, but we are hopeful that they can be resolved in the future. Benefit Points is about using value to prioritize the Product Backlog, but since it is concerned with creating value, and the fact that we visualize value through our solution, we are adamant that our solution resolve, or partly resolve the problem of Value Creation. Organization Resources is a far wider problem to handle. It involves a number of parameters, and we can not confirm, nor reject the possibility that our solution could have a partially impact on resolving the problem of available resources for a Product Owner in an organization.

7.3.1 Jira as Artifact Platform

The artifact we have developed, is a plugin for Jira (Section 4.1). By choosing an existing tool, we did not have to think about the implementation of existing project management functionality for our implementation of Benefit Points. However, developing a plugin, did bring some limitations as to what was feasible to accomplish. We argued that Atlassian provided well documented knowledge of how to develop a plugin for Jira, but this was only partially true, as we discovered as development progressed. Documentation was clear as to creating a few simple plugins, but complex examples were absent. Some resources was also scattered, outdated, or unclear as to which actions was required. This slowed development down and made the learning curve steeper than anticipated.

Though the documentation left much to be desired, the vital aspect was that Jira is a commonly used project management tool, which proved useful for the evaluation process. However, Jira uses different project templates, such as Scrum and Kanban, which lead to a decreased target audience for the evaluation, as our focus was on Scrum.

Overall, Jira as the platform for our artifact, has been a good choice, even given the negative aspects encountered throughout the project.

7.3.2 Using our Artifact

In this project, we implemented the theory of Benefit Points in a solution to prioritize the Product Backlog. We decided it would be fruitful to use Benefit Points to prioritize our own project. In the early stages of our project, we listed a number of elements which was required to be made. This list included report and artifact related items. Overall, we are judged by our report, so report related items are naturally assigned higher Benefit Points than artifact related items. However, due to using Design Science research as our methodology, our artifact required a higher amount of Benefit Points.

This lead us to create two estimation lists. The first list was an overall list of all elements to visualize the value generation for the complete project. The second list was created to prioritize and re-adjust Benefit Points as the project proceeded. This meant that the fundamental elements of the report was highly beneficial at the beginning of the project compared to the evaluation part, even though evaluation is a highly beneficial element of the report. The overall experience was that dependencies existed before specific elements could be proceeded with, but Benefit Points provided a clear visualization of the benefit of the elements in our project.

After the evaluation of our solution, we used our solution to estimate the value of the feedback received. By using our solution we could see the items which would provide most value to our project, and prioritize the next steps to take. These balanced Benefit Points can be seen on Figure B.1.

Chapter 8

Conclusion

This project has taken a Design Science research approach to create a solution to the problem the Product Owner in Scrum has of prioritizing the Product Backlog. Our research question was as follows:

• RQ: How can the Product Owner's value based prioritization of the Product Backlog be improved?

To address our RQ, we created an extension to the project management tool, named Jira, and incorporated the theory of Benefit Points (Hannay, Benestad, and Strand 2017a). The solution takes a value based perspective to resolve our research question, and provides assistance to use Benefit Points in practice.

We performed two evaluations of our solution, with three practitioners, and we discovered a number of issues, e.g. that value based thinking is a difficult discipline to enact. It would require an organizational change in the processes at the start of a project. The practitioners believed the transition would be smoother with our solution. However, we can only conclude that the solution potentially resolves our RQ.

To summarize, the solution is based on relevant theory chosen to resolve a justified problem (Haaber and Grønhøj 2017). We have evaluated our solution with three practitioners with relevant experience with Scrum and Jira, which justifies the potential of our solution and the theory of Benefit Points.

Chapter 9

Limitations & Future Work

In this chapter, the limitations and future work for this project is presented. In this project, we have had a few limitations, especially in regards to the evaluations. Since we only performed two evaluations of our solution and the theory of Benefit Points, our empirical foundation is limited. However, before resolving this limitation, improvements should be made to our solution (Section 6.1.1).

9.1 Benefit Points for Stories

A significant limitation is the fact that Benefit Points is only implemented for Epics. An Epic is a high level collection, which contains a number of stories, and getting closer to the technical/story levels as seen on Figure 2.1, would be beneficial. The natural step to take next, to resolve the limitation, is to add the theory of Benefit Points for stories. Hannay, Benestad, and Strand (2017) have written an article about the addition of Benefit Points for Stories. *Earned Business Value: See That You Deliver Value to Your Customer* (Hannay, Benestad, and Strand 2017b), presents how to use story points to find the Benefit/Cost ratio, as well as presenting how to use Benefit Points with stories. By using Benefit Points for stories, we can get a more complete experience for the entirety of a project. It would also mean that prioritization of stories would be more value based.

9.2 Benefit Points for Bugs

The currently available theory of Benefit Points only works with Epics and Stories. In Jira, there exists an issue type, named Bug. The current theory is not concerned with Bugs. To address this limitation, we could outline a theoretical foundation based upon the theory of Benefit Points and it would most likely be similar to the theory for Stories. Our reasoning, is that finding out what value can be created by fixing a

Page 44 of 55

bug can be beneficial. In a perfect world, bugs do not exist or at least the bugs will be resolved, but this is not always the case. Some bugs may be deemed too minor or unfeasible to resolve. By bringing in the aspect of Benefit Points to determine if a bug should be resolved or not can be helpful with bringing clarity and potentially more value to be created.

9.3 Dependency Inclusion

In Jira, issues can be linked together. This means that a story can be blocked by another story or even an Epic. Our solution does not account for these dependencies, which can affect how a Product Owner prioritize the Product Backlog. By including the dependencies of issues into the calculation of Benefit Points, a new level of prioritization can be achieved. An example could be a story, which on itself add low value to the overall project. However, this story blocks an Epic, which brings significantly more value to the project. The story which blocks the Epic is required to be resolved before the valuable Epic can be resolved.

It would be beneficial for the prioritization of the Product Backlog, to create an option which takes dependencies into account. It is important to enable a user to create the most value at any given time, but also enabling value creation for the entirety of the project.

9.4 Backlog Integration

The Product Backlog is a central artifact, and the prioritization process is done in the Backlog. Currently, Balanced Benefit Points is available on a separate page, which limits the usability of our solution. Integrating the balanced Benefit Points into the Product Backlog page would be beneficial and make it less tedious to switch between the pages, as discovered in our evaluations (Chapter 6). Various ways could be used to represent the balanced Benefit Points, including the actual values or color codes.

Color codes can be presented on each item in the backlog, and when an item is green the value is higher than a threshold. If the value is red, the value created would be below the threshold and thus not wise to develop next. The threshold could be defined based on a number of factors like average value, benefit/cost, or even just a user setting.

Page 45 of 55

Bibliography

[Atl16]	Atlassian. Active Objects. Dec. 8, 2016. URL: https://developer.atlassian.
	com/server/framework/atlassian-sdk/active-objects/(visited
	on 02/10/2018).
[Atl18a]	Atlassian. Design. June 5, 2018. URL: https://atlassian.design/
	server/ (visited on 06/05/2018).
[Atl18b]	Atlassian. <i>Jira</i> . June 1, 2018. URL: https://www.atlassian.com/software/jira (visited on 06/01/2018).
[BHH15]	Ruth Baumgart, Markus Hummel, and Roland Holtenn. "Personality Traits of Scrum Roles in Agile Software Development Teams - A Qualitative Analysis". In: <i>ECIS 2015 Completed Research Papers</i> (2015).
[Con09]	Kieran Conboy. "Agility from First Principles: Reconstructing the Con- cept of Agility in Information Systems Development". In: <i>Information</i> <i>Systems Research</i> 20.3 (2009), pp. 329–354.
[HBS17a]	Jo Erskine Hannay, Hans Christian Benestad, and Kjetil Strand. "Benefit Points: The Best Part of the Story". In: <i>IEEE Software</i> 34.3 (2017), pp. 73– 85
[HRS17b]	Io Frskine Hannay, Hans Christian Benestad, and Kietil Strand, "Farned
[1100110]	Business Value: See That You Deliver Value to Your Customer" In: IFFF
	Software 34.4 (2017), pp. 58–70.
[HG17]	Martin Haaber and Patrick Grønhøi. "A Product Owner's Problems In
	Scrum - A Literature Review". In: Semester project department of Com- puter Science, Aalborg University (2017).
[Inc18]	Scaled Agile Inc. <i>Weighted Shortest Job First</i> . Apr. 20, 2018. URL: https: //www.scaledagileframework.com/wsif/ (visited on 05/22/2018).
[Jin18]	Jinfonet.net. 3-Tier Architecture: A Complete Overview. May 24, 2018. URL:
	https://www.jinfonet.com/resources/bi-defined/3-tier- architecture-complete-overview/(visited on 05/24/2018).
[JK08]	Ken H. Judy and Ilio Krumins-Beens. "Great Scrums Need Great Product
	Owners: Unbounded Collaboration and Collective Product Ownership".
	In: Proceedings of the 41st Hawaii International Conference on System Sciences (2008).
[KLC16]	Sigurhanna Kristinsdottir, Marta Larusdottir, and Åsa Cajander. "Respon-
	sibilities and Challenges of Product Owners at Spotify - An Exploratory Case Study". In: <i>Human-Centered and Error-Resilient Systems Develop-</i> <i>ment</i> (2016), pp. 3–16.

Page 46 of 55

- [KV12] William Kuechler and Vijay Vaishnavi. "A Framework for Theory Development in Design Science Research: Multiple Perspectives". In: *Journal of the Association for Information Systems* 13.6 (2012), pp. 395–423.
- [Leh+15] Timo O.A. Lehtinen et al. "Why the Development Outcome Does Not Meet the Product Owners' Expectations?" In: International Conference on Agile Software Development XP 2015 (2015), pp. 93–104.
- [MDK09] Nils Brede Moe, Torgeir Dingsøyr, and Øyvind Kvangardsnes. "Understanding Shared Leadership in Agile Development: A Case Study". In: *Proceedings of the 42nd Hawaii International Conference on System Sciences* 42 (2009).
- [MS95] Salvatore T March and Gerald F Smith. "Design and natural science research on information technology". In: *Elsevier* 15.4 (1995), pp. 251–266.
- [Oom+17] Sandra Oomen et al. "HOW CAN SCRUM BE SUCCESFUL? COMPETENCES OF THE SCRUM PRODUCT OWNER". In: *Proceedings of the 25th European Conference on Information Systems* (2017).
- [PB17] Jan Pries-Heje and Richard Baskerville. "The translation and adaptation of agile methods: a discourse of fragmentation and articulation". In: *Information Technology & People* 30.2 (2017), pp. 396–423.
- [Pef+07] Ken Peffers et al. "A Design Science Research Methodology for Information Systems Research". In: *Journal of Management Information Systems* 24.3 (2007), pp. 45–77.
- [PHL12] Maria Paasivaara, Ville T. Heikkilä, and Casper Lassenius. "Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum". In: *IEEE Seventh International Conference on Global Software Engineering* (2012).
- [Ros12] Jeremy Rose. *Software Entrepreneurship: two paradigms for promoting new information technology ventures.* Vol. Beta. Department of Computer Science Selma Lagerløfs Vej 300 Aalborg Øst, 2012.
- [SIJ14] Hrafnhildur Sif Sverrisdottir, Helgi Thor Ingason, and Haukur Ingi Jonasson. "The role of the product owner in scrum - comparison between theory and practices". In: *Procedia - Social and Behavioral Sciences 119* 119 (2014), pp. 257–267.
- [Sim96] Herbert Alexander Simon. *The Sciences of the Artificial*. Vol. 3. MIT Press, 1996.
- [Som16] Ian Sommerville. *Software Engineering*. Vol. 10. Pearson Education, 2016.
- [SS16] Jeff Sutherland and Ken Schwaber. The Scrum Guide. July 1, 2016. URL: http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf (visited on 02/10/2018).
- [Tra18] Pivotal Tracker. *Pivotal Tracker*. June 1, 2018. URL: https://www.pivotaltracker. com/features/ (visited on 06/01/2018).

[ver17]	versionone. 11th Annual State of Agile Report. Apr. 6, 2017. URL: https:
	<pre>//explore.versionone.com/state-of-agile/versionone-11th-</pre>
	annual-state-of-agile-report-2 (visited on 02/13/2018).

- [Ver18] VersionOne. VersionOne. June 1, 2018. URL: https://www.versionone. com/ (visited on 06/01/2018).
- [VPB12] John Venable, Jan Pries-Heje, and Richard Baskerville. "A Comprehensive Framework for Evaluation in Design Science Research". In: *Design Science Research in Information Systems* (2012), pp. 423–438.

List of Figures

2.1	Blue rows are enterprise level and red are project, grey rows are techni- cal levels. Figure obtained from (Hannay, Benestad, and Strand 2017a,	
	p. 75)	3
2.2	Assignment of Benefit Points. (a) Epics' contribution to the Objectives.	
	(b) Objectives' contribution to the Returns. (c) The Balanced Benefit	
	Points. The cells in white are input cells, and the orange shaded cells	
	shows the automatic generated results of the inputs.	5
2.3	The Iron, Agile, and Benefit/Cost triangles (Hannay, Benestad, and Strand	
	2017b, p. 61)	6
3.1	Illustration of steps in Design Science. Figure obtained from (Peffers et	
	al. 2007, p. 54)	8
3.2	Our illustration of the six activities in Design Science. Modified version	
	from (Peffers et al. 2007, p. 54)	12
11	Version One Backley	14
4.1	Pivotal Trackor Backlog	14
4.2		14
4.5	Bonofit Doints Tab	10
4.4	Deficit Folitis Tab	10
4.5	Dialog for adding a Return	20
4.0	Dialog for editing a Return	20
4.7 1 Q	Banafit Doints Tab	20
4.0 1 9	Returns Tab	21
4.J	Returns fab	21
4 11	3-I avered Architecture (Adopted from (Jinfonet net 2018))	22
4 12	Model Component of the problem area	24
4 13	Fntity Relationshin Diagram	25
1.10		20
5.1	Implemented version of the Benefit Points Tab	29
5.2	Implemented version of the Returns Tab	30
5.3	Implemented version of the Balanced Benefit Points on the issue page	31
A.1	Jira Dashboard	52
A.2	Jira Create Project	52
A.3	Jira Backlog	53

Page 49 of 55

A.4	Jira Issue Page	53
B.1	Using our own solution, showing the Balanced Benefit Points	54

Page 50 of 55

Listings

5.1	BenefitPoint data	26
5.2	Benefit Point Servlet	27
5.3	Total Contribution per Objective calculation	28
5.4	Single balaned Benefit Point calculation	29

Page 51 of 55

Appendix A

Jira

ÄJIRA Da	shboards - Projects - Issues - Boards - Create		Search	٩	44	@-	۰.	•			
System E	Dashboard							•••			
Introduct	lon	Assigned to Me									
Ŵ	Welcome to Your Company JIRA Not sure where to start? Check out the JIRA 101 guide and Atlassian training course.	You currently have no issues assigned to you. Enjoy your dayl									
	You can customize this text in the Administration section.	Activity Stream									
		Your Company JIRA									
		Alarch 23 admin created BE-1 - Hej 2 23/Mar/16 10.43 AM Comment									



Create p	project		
📀 SOFTWA	ARE		^
	Scrum software development Agile development with a board, sprints and stories. Connects with source and build tools.		
	Kanban software development Optimise development flow with a board. Connects with source and build tools.		
	Basic software development Track development tasks and bugs. Connects with source and build tools.		
	SS		- 1
	Project management Plan, track and report on all of your work within a project.		
h	Task management		-
Import a pro	oject 🕀 Create with shared configuration Crea	te sample data Next	Cancel

Figure A.2: Jira Create Project

Page 52 of 55



Figure A.3: Jira Backlog



Figure A.4: Jira Issue Page

Appendix B

Balanced Benefit Points

	oints?	inced benefit int the e individual Epics to create	Total 🗸	44.67	39.4	30.39	30.36	29.53	15.81	15.65	14.2	220.0	
	Why use Balanced Benefit P	lanced Benefit Points provides the bala ballong they have taken into accound influding distinct data and a data accound its value: lanced Benefit Point exists to prioritize 1 anced Benefit primes.	Objective 36 Increase value creation	2.14	3.42	2.14	2.14	1.28	1.28	3.42	2.14	17.95	
		P), the Project Ba mount of Benefit for TotalBP(Obj)). Ep Ba Ba	Objective 35 Provide Improved Prioritization	16.14	26.08	6.21	16.14	6.21	9.93	1.24	3.73	85.68	
ints	anced Benefit Point calculated?	lated by taking a single Benefit Point(B) Benefit Points(TotaBP), and the total a ssociated with the single Benefit Point(otaBP(Ob)) = Galanced BP.	Objective 34 Reduce the learning curve of Benefit Points	11.42	0.54	7.07	2.72	7.07	2.72	1.63	2.72	35.89	
Returns Balanced Benefit Po	How is a Bal	A Balanced Benefit Point is calcu Veight(PW), the Total amount of veight(PW), the specific Objective a onits for the specific Objective a The formula is BP*PW*TotalBP/T	Objective 33 Spread the knowledge of Benefit Points	14.97	9.36	14.97	9.36	14.97	1.87	9.36	5.61	80.47	
Benefit Points Objectives Contributions	What is a Balanced Benefit Point?	A Balanced Benefit Point is the calculated value of a Benefit Point, taking into value of a Benefit Point, taking into virous information into account to create F balance.	Epic	Improvements to existing solution	Story Points Implementation	Clarifications of size scheme etc.	Backlog Integration	Learning Curve Optimizations	Colour Codes implementation	Visualization of value created	Benefit Points Template	Total	

Figure B.1: Using our own solution, showing the Balanced Benefit Points

Page 54 of 55

Appendix C

Testing Our Plugin

To test our plugin, one must have admin access to a version of Jira with Jira Software installed. Jira Software and Jira Core version 7.6.1 is the tested version. If one do not have access to Jira, it is possible to download the Atlassian SDK on Atlassian's homepage, more specifically follow their guide to setup a development environment. If Jira is available, a .jar file named "valuepoints-1.0.0-SNAPSHOT" has to be installed under the plugin management of Jira.

If the route for a development environment setup has been followed, one must choose a folder to execute Jira from. It will require between 1GB and 2GB of available space. Open an instance of cmd in the designated folder and choose either "atlas-run" or "atlas-debug". The first command is for running a normal distribution of Jira, and the second command will run a version, which is possible to debug while developing. If the designated folder also contains the source code for our plugin, it will automatically install the plugin. In our case, the source folder is named valuepoints.

When Jira has started, which can take 5-10 minutes, it can be accessed on port 2990 and the full address is localhost:2990/jira. Use a browser of your choosing, but Chrome seems to be operating at a reasonable speed. A local distribution has "admin" as both username and password. Create a new Scrum Software project and the plugin elements should be visible for that project.

Page 55 of 55