# Visualizing Historical Project Data to Improve Technical Debt Management

## *A Design Science Research*

Vilde Rabbe Husby

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
60 credits

Institute for Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

# Visualizing Historical Project Data to Improve Technical Debt Management

*A Design Science Research*

Vilde Rabbe Husby

Visualizing Historical Project Data to Improve Technical Debt Management

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

# Abstract

**Background** Technical debt negatively affects the company's ability to deliver customer value. To identify and manage technical debt continuously, one benefits greatly from using a management tool. The design of current technical debt tools does not realize its full potential. Non-technical or technical stakeholders should be able to navigate and analyze their historical technical debt data to improve their management of technical debt. One should present historical data to be accessible and informative so that all stakeholders can correctly understand the company's technical debt. Such insight can be crucial to grasping the right future strategies for technical debt management. This research aims to determine how we can visualize historical project data to improve technical debt management.

**Methods** We have used design science research to design and develop an artifact containing six different visualizations of historical project data. As a part of this research design, we have conducted qualitative data collections with seven people from four different companies.

**Results** The results reveal how, where, and when the six different visualizations of historical data are useful for whom (non-technical or/and technical stakeholders). Additionally, it shows their level of understandability and areas for improvement for future research.

**Conclusion** The findings support that stakeholders can use the visualizations to improve the technical debt management activities: monitoring and communication. We proposed a new approach to technical debt monitoring that watched the amount of resolved technical debt over time compared to important metrics including unresolved technical debt, project goals, technical debt types, velocity, and risk. Additionally, we improved technical debt communication by presenting resolved technical debt items in useful and understandable graphical representations including bar charts and pie charts.

# Acknowledgements

First and foremost, I want to thank my supervisor, Antonio Martini, for his academic guidance, encouragement, and advice throughout the work and writing of this thesis.

I would also like to thank all of the people involved in the research by participating in the data collection. Their genuine interest in the topic and valuable input were decisive for the thesis.

Lastly, I want to thank my family and friends who encouraged and supported me in writing this thesis.

**Vilde Rabbe Husby**
University of Oslo
May 2022

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Software companies need to deliver customer value from both a short-term and long-term perspective. Unfortunately, customer value is often limited to the activity of delivering features to the customers. With too much focus on this goal, other quality aspects will receive less attention, such as good design, good programming practice, and complete coverage testing. Eventually, such a prioritization will negatively affect the company's ability to deliver customer value due to the occurrence of technical debt.

*Technical debt* (TD) is a financial metaphor that refers to the consequences of implementing sub-optimal solutions in order to meet short-term goals. For example, the goal of reaching time-to-market may outweigh the code quality. Whether these short-cuts are made consciously or unconsciously, the TD will eventually have to be repaid with compounding interest, which is the extra cost due to the presence of TD.

TD can occur in all phases of the software development life cycle phases, from the planning phase to the deploying and maintenance phase. To identify all types of TD and manage it continuously, one benefits greatly from using a management tool. There exists a variety of different management tools for handling TD. The most used are backlogs and static analyzers [33]. The disadvantage with static analyzers is that they will only partially measure TD, which will cause the rest of the TD to become invisible, as described in a quadrant by Kruchten et al. [25].

The TD metaphor was initially introduced to facilitate the communication between developers and business stakeholders. The decisions on how resources should be used are often made by non-technical stakeholders, and thus they also decide when and how much time is to be spent on TD. Therefore it is crucial that technical and non-technical stakeholders are aligned on their perception of TD in the system and that this perception is correct.

Martini et al. [33] expressed the challenge that the TD backlog itself did not make the TD more convincing for the management to be refactored. Still, it served the teams to remember to take care of TD, which would otherwise remain invisible and overlooked. This challenge indicates a need to improve the TD backlog to appeal more to the management part of the organization.

A few studies report the use of TD backlog management, e.g., [34, 44, 51]. Based on these studies, we can argue that current tools lack or have little support for the activities of navigating and analyzing the debt that has already been repaid [29]. By recording and analyzing historical data, we can provide useful insight into past events, for example, by identifying trends in data. The insights could form a better basis for making accurate business decisions in the future.

Additionally, we identified a few studies that argue how recording repaid TD can be advantageous. Power [36] emphasizes the value of recording repaid TD as it provides a sufficient overview of TD impact and investment. According to Guo et al. [20] historical (effort) data can be used to achieve a more accurate estimation when estimating principal, i.e., the amount of effort required to complete a TD item task [20]. Utilizing the historical data to do these estimations can be a step towards a fully automated TD management [34].

The repaid TD can be valuable for technical and non-technical stakeholders as long as it is easily accessible and informative. The challenge is that with the evolution of software, the complexity increases in innovation and size. As a result, information gathered from software development activities, such as repayment, will increase. A technique that can make this task easier is software visualization.

Visualization techniques allow the representation of information that is often difficult to analyze in textual or tabular form [1]. We want to figure out how we can visualize the repaid TD so that it can support stakeholders in their management of TD. Therefore, we want to address the following research question:

> *RQ: How can visualization of historical data improve the management of technical debt?*

To answer this research question, we have empirically designed and developed an artifact that is integrated with a TD-enhanced backlog. Our main contribution is six different visualizations represented as graphical user interfaces that, in each way, serve as a tool to improve the management of TD. The visualization's utility and understandability are demonstrated through rigorous research methods throughout the research process.

# Chapter 2

# Background

This chapter presents the results from the literature review on the main topics related to the research question. First, in Section 2.1, we present the main concepts of TD: the metaphor, trade-off, types of TD, and management of TD. Then, in Section 2.2, we present different relevant metrics for analyzing historical data: the principal and the interest. Additionally, we present relevant metrics from the analysis of the product backlog: velocity. Lastly, in Section 2.3, we present relevant topics on how we can visualize the historical data. A summary of the literature review is presented in Section 2.4. In addition to the literature review, we also conducted a tool survey. The results from this are presented in Section 2.5

Most literature that mentions this topic refers to recording repaid debt/completed items as historical data [20, 54, 28]. Moreover, a record can be seen as a collection of this data. Hence, we will refer to this feature as a historical record.

## 2.1 Technical Debt

### 2.1.1 The metaphor

The term technical debt was first introduced by Cunningham, which made a metaphor using financial terms about problems with the software:

> *"Shipping first-time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. Objects make the cost of this transaction tolerable. The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt"* [11].

This metaphor aimed to sell object-oriented programming by saying that it made debt tolerable. However, the metaphor extended beyond that and became helpful to the entire software engineering environment. Using this metaphor, we can better communicate this complex problem that constitutes ”TD” to the managers, thereby making more adequate business decisions.

In more recent literature concerning TD, the metaphor is simplified by phrasing ”making short-cuts” or ”sub-optimal” code. Avgeriou et al. [5] define the term TD as *”delayed tasks and immature artifacts that constitute a ”debt” because they incur extra costs in the future in the form of the increased cost of change during evolution and maintenance.”*

### 2.1.2 Trade-off

In contrast to the financial world where the debt is calculated with money, TD is diverse and complex and thereby not possible to calculate in the same way. For example, in software engineering, one needs to consider a trade-off between software quality and the cost of the software development process. Choosing the latter takes on debt by taking sub-optimal solutions to shorten time-to-market. The principal refers to the cost savings gained by taking on the debt or the cost of completely repairing the TD imperfection and eliminating the debt. The long-term negative impact is referred to as the interest paid on the debt [5].

### 2.1.3 Types of technical debt

When this metaphor was initially proposed and concerned, it originated in coding practices—for example, code duplication or violation of coding conventions. To date, the concept of TD applies to other phases in the software development life cycle, such as architecture-, design-, testing-, and documentation debt [8]. In a systematic mapping study done on TD and its management, they recorded a total of 10 different types of TD [29].

It may be beneficial to take on debt or not repay debt in some cases. Common for these types of TD is that the debt can stay in the system without any consequences in terms of interest. Such a case especially applies to small projects, like startups, since they are typically under extreme pressure to get to market quickly with limited resources and high uncertainty [6]. In this case, introducing TD can be seen as a strategic advantage.

### 2.1.4 Managing technical debt

TD needs to be managed to track visible and invisible debt and keep the accumulation of debt under control [29]. In a systematic mapping study on TD and its management, Li et al. [29] identified eight different TD management activities: TD identification, TD measurement, TD prioritization, TD prevention, TD monitoring, TD repayment, TD representation/documentation, and TD communication.

**TD identification** "detects debt caused by intentional or unintentional technical decisions in a software system through specific techniques" [29]. The main approach for TD identification is code analysis. This can be explained by fact that one can reuse existing analyzing tools for detecting TD related issues, e.g. violations in code, incomplete testing, complexity and code smells [29].

**TD measurement** "quantifies the benefit and cost of known debt in a software system through estimation techniques or estimates the level of the overall TD in a system [29]. The most studied method is to calculate TD through mathematical formulas or models, followed by code metrics and human estimation [29]. This activity is relevant for the historical record because by storing information about the estimates done on previous TD items or in total can potentially be used to improve the measures of unresolved TD.

**TD prioritization** "ranks identified TD according to specific predefined rules to support deciding which TD items should be repaid first and which TD items can be tolerated until later releases" [29]. One example is to use a cost/benefit analysis, where the TD items with a higher cost/benefit ratio are to be repaid first [53].

Another known technique is risk analysis. For example, TD items that pose a higher risk of generating TD interest would be more urgent for refactoring to prevent high maintenance costs [14]. The advantage of using the risk metric is that it is non-technical and easy to comprehend and, therefore, a very suitable metric to reach business leaders. When working with the risk, it is normal to specify a threshold for the risk in total. A risk threshold is the maximum amount of risk that the team can manage [14]. As the team approaches this value, it will indicate that action must be taken to reduce TD.

TD prioritization is relevant for the historical record because repaid debt data can provide valuable insights into how the team has prioritized TD in the past. Thereby, it can affect decisions on the prioritization of the unresolved TD.

**TD prevention** ″aims to prevent potential TD from being incurred″ [29]. This activity is mainly supported by software development process improvement, such as following a well-defined project planning and adopting good practices for software development. Identifying, managing, and estimating TD is also considered a preventive action [15].

**TD monitoring** ″watches the changes in the cost and benefit of unresolved TD over time″ [29]. This activity addresses the problem when there is ″too much″ debt in a system. Monitoring TD can be done in several ways, by including different metrics [29]. One example of how to address this is a threshold-based approach, where one defines thresholds for TD-related quality metrics and issues warnings if the thresholds are not met [29].

In the historical record, we want to monitor the changes over time by using graphic representations. For example, by comparing unresolved TD with repaid TD over time, we can identify correlations and insight that can affect future strategies.

**TD repayment** ″resolves or mitigates debt in a software system by techniques such as re-engineering and refactoring″ [29]. The refactoring-approach is the most used, which involves making changes to the code, design, or architecture of a software system without altering the external behaviors of the software system to improve the internal quality. While re-engineering involves evolving existing software to exhibit new behaviors, features, and operational quality [29].

**TD representation/documentation** ″provides a way to represent and codify TD in a uniform manner addressing the concerns of particular stakeholders″ [29]. Typical for studies covering this is to represent TD as TD items. The proposed formats differ, but all include the fields: ID, Location, Responsible/author, Type, and Description [29]. In our thesis, we wanted the historical record to support the representation of TD as TD items with the fields mentioned. The documentation and following properties are fundamental assumptions for our artifact. Further, we want to build on these items to represent the items graphically.

**TD communication** ″makes identified TD visible to stakeholders so that it can be discussed and further managed″ [29]. Li et al. [29] further discovered that the most studied approaches used to aid communication discussed in the literature are TD Dashboard, and TD Backlog. For example, the TD backlog can be communicated to the management, to show the risk associated to the system [34]. The main challenge related to TD communication activity is to fill the information gap between non-technical and technical stakeholders [24]. Research on TD management in practice [52] emphasize the importance of having a communication structure in place between the non-technical and technical stakeholders where there is room to talk about TD. This activity is usually facilitated by a product owner.

Our artifact supports the approach of using TD backlogs as means for communication. Further, the literature has emphasised that this approach in it self, is not enough to make the TD more convincing for the management to be refactored [33]. Therefore, we are interested in improving this approach. A benefit of the backlog strategy is that TD items are recorded, also in the long-run. This information should be informative and accessible to improve the backlog approach, and further aid communication.

## 2.2 Analysis of historical record

An advantage of recording repaid TD items is that they can be analyzed and give valuable information to technical and non-technical stakeholders. This information can also be used to improve principal and interest estimates.

### 2.2.1 Calculating the principal

As stated earlier, calculating the principal is challenging in terms of money, and a good way of doing so is not found yet. However, there is a common approach to estimating the principal using relative values. For example, Seaman and Guo [45] propose to initially estimate the principal on a rough ordinal scale from low to medium to high, which allows enough understanding to contribute to iteration planning.

Zazworka et al. [54] proposes that historical effort data can be used to make a more accurate and reliable estimation beyond the initial high/medium/low assessment. For example, suppose a TD item is a set of classes that must be refactored. In that case, the historical cost of modification of those classes can be used as the future modification cost (principal of the TD item) estimation.

Curtis et al. [12] propose a method for estimating the TD-Principal, where they agree with this by stating that the time to repay a TD item could be available from historical effort data. The number of hours to repay each TD item is one of three factors to calculate the principal of the complete application. The other two are the number of TD in an application and labor cost. For the latter, thy propose using an average burdened rate for the developers assigned to the activity.

### 2.2.2 Calculating the interest

Analysis of historical data can also be used to make assumptions about the interest. In the literature, the term interest is usually split into two conditions: the extra cost originating from the TD item and the extra cost from components dependent on the TD item. In addition to this, there is also an extra cost that arises from postponing the fixing of the issue. According to Lenarduzzi et al. [27] postponing fixing activities might also have a ripple effect, such as impacting other parts of the system. Zazworka et al. [54] proposes that historical usage can be used to improve the estimate of both of these interest conditions. The factor they list as necessary for this is TD principal, data on past defects, effort, and changes.

### 2.2.3 Analysis of the product backlog

The product backlog includes items with both statuses as completed/repaid and unresolved. Many TD management tools are using this same structure for recording TD items. The similarities provide opportunities to apply metrics used for analyzing backlog items to a historical record. As for the historical record, we are especially interested in the items categorized as completed/repaid and how these can be analyzed to contribute valuable information for developers/managers.

**Velocity** *Velocity* is a popular metric used in multiple management tools to measure how fast a team delivers business value. The term originates from agile software development and is a number representing total amount of work that is done during each sprint [42].

In Jira, [4] the velocity is calculated by taking the average of the total completed estimates over the last several sprints. The velocity can fluctuate but should begin to stabilize across multiple sprint iterations. Once a predictable velocity is established for a team, it can be used in sprint planning to determine how much work a team can take on for that sprint. Another management tool that uses Velocity is Forecast [41]

Additionally, velocity can also be used as a diagnostic metric. Velocity can be affected by the amount of TD. The higher the total TD gets, the more likely it is that the velocity decreases [36]. At worst, TD can rise to the level where the team cannot add new functionality to a product, which is equivalent to the velocity at zero. To avoid this situation, one needs to have control over the progress of this relationship.

An important point that should be noted is that historical data is non-existent at the start of the project. Therefore, comparing current data with historical data will be either impossible or not valuable initially. However, estimates related to these comparisons will increase utility by accumulating data. The more accurate and detailed the data an organization has, the more reliable the estimation will be [20].

## 2.3 Visualization of a historical record

According to Brown et al. [8] visibility is an important property of TD. Significant problems arise when debt is not visible to business stakeholders since they are responsible for the decisions to repay the debt. Therefore, visualization of TD should be used to improve stakeholder communication in the decision-making process.

The evolution of software in a project tends to produce a large amount of data. As a result, the number of recorded repaid TD items will grow to a large extent. To help visualize the TD items, they can be mapped to graphical representations. By transforming intangible software entities and their relationship into visual metaphors, they can easily be interpreted by human beings [16].

### 2.3.1 How to visualize TD

Power [36] studied the causes and impacts of TD on product development for teams and organizations. In the article, he discusses the challenge of visualizing TD. He argues that ″pie charts and bar charts that show how much capacity the team is investing in reducing and managing TD are very useful.″ The advantages are that ″they quickly show the relative effort in proportion to the other areas the team is spending their time.″

The article also presents how a large communication technology company uses pie charts to visualize planned investment in percentages, including features, performance, TD, and defects. The chart is used in retrospective meetings to compare planned versus actual investment for the current sprint [36]. The charts presented in this article are used in specific events in real-time. However, it could be interesting to see if such ideas are also applicable (and useful) to represent the long-term effects of past events.

Alves et al. [1] did a systematic mapping study on identification and management of TD, where 22 of the 100 primary studies proposed software visualization techniques in the context of TD management. The most proposed visualization techniques were dependency matrix, bar chart, and pie chart format. The study also showed that the most common platform proposed to display visualizations is the spreadsheet (with 15 citations). This kind of manual solution, a spreadsheet, is far from ideal, as it requires much effort to record the data extracted from the software project and keep it up-to-date [1]. Visualizing TD through software visualization techniques is still a challenging task due to the lack of support tools and research. Although software visualization has been shown to benefit the process of software understanding [1].

### 2.3.2 Visualization in Backlog Management

To gain more insight into ”visualizing TD,” we expanded the search to include visualization techniques used in backlog management. By investigating different types of software visualization techniques already proposed in other software maintenance and evolution contexts, we can adapt their aspects and use them to manage TD repayment. In this way, we don’t need to start from scratch when designing our artifact. The specific visualizations mentioned below will be included and discussed in the artifact’s design in Chapter 4.

**Burndown Chart** A known visualization technique used in agile backlog management is a *burndown chart* (see Figure 2.1). It is a graphic representation of how quickly a team works through a customer’s user stories. The chart shows the estimated effort of work (vertical axis) remaining over time (horizontal axis). The time frame is usually set to a sprint. For example, a team can start a sprint with 150 story points, and then the goal is to burn down to a point where there is no work left at the end of the sprint. The stippled lines, as one can see in the chart, can be used as a leading indicator to predict when the work will be finished [42].

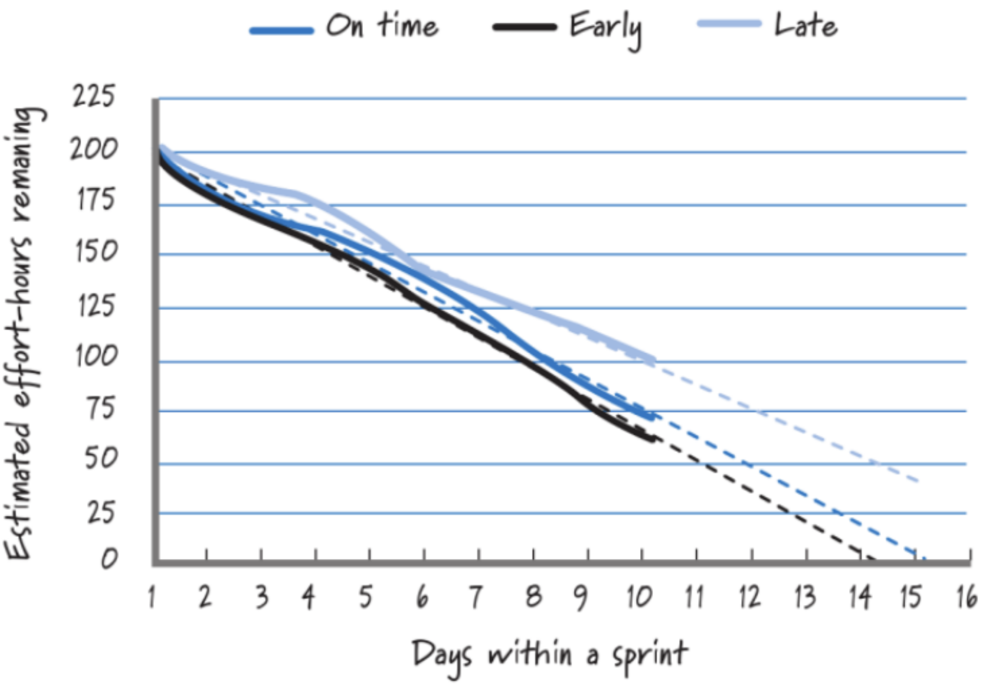


Figure 2.1: Burndown Chart [42]

**Burnup Chart** Another known visualization technique used in agile backlog management is the *burnup chart* (see Figure 2.2). The burnup chart is a visual representation of progress toward a target line. The target is usually set to a longer time frame, e.g., the number of user stories in a release. The vertical axis represents the amount of work, and the horizontal axis represents time. Also here, one can use stippled lines to predict when the target will be reached.

Similar to the burndown chart, this chart also reports the amount of work completed towards a goal; only the progress moves upwards (burn up), which better illustrates the work that has been accomplished. This chart aims to offer the team insights on project progress. By looking at the entire scope, the team can easily see changes in the development [42].

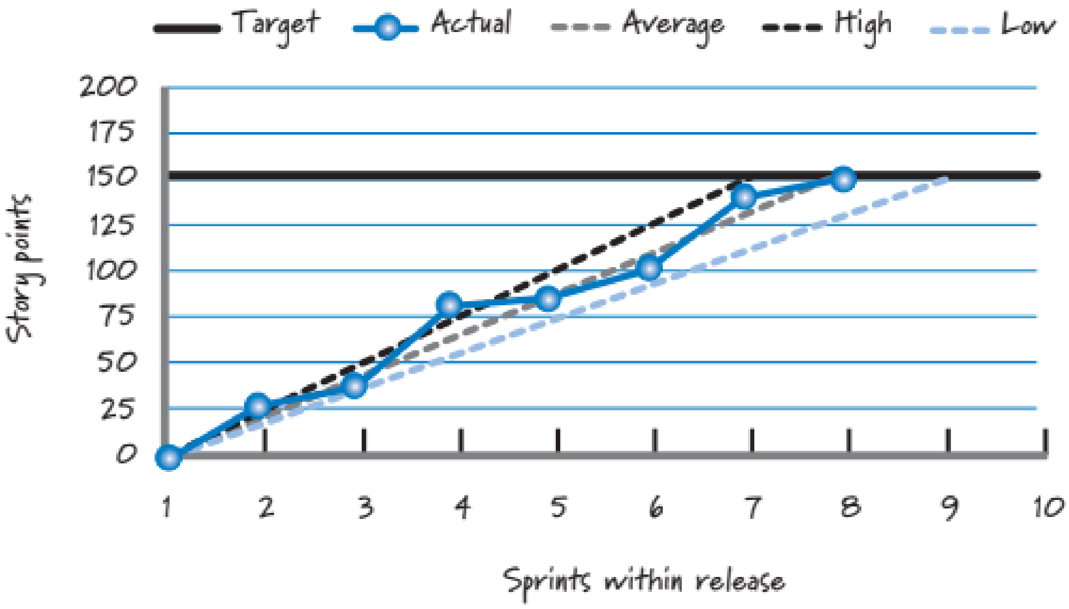


Figure 2.2: Burnup Chart [42]

### 2.3.3 Defining completed work

In backlog management and their visualization charts, like burndown and burnup charts described above, it is typical that the vertical axis represents total work remaining/completed. To define such values for the TD items, one usually estimates how much work they believe it requires to resolve it. The estimation can be the number of hours/days, story points, or another custom estimation the team has decided on. There is also possible to use the number of issues [3].

**Number of hours/days** To define completed work by estimating the number of hours or days is the most descriptive option. According to [42], it is one of the most common units of measure, after story points.

**Story Points** *Story points* is one of the most popular ways to measure product backlog items, especially in agile teams [42]. It is a estimate of the overall effort required to fully implement a product backlog item [3]. For example, a team can use an exercise called planning poker to give an estimate for each backlog item, where the estimated span will be 1-13. This exercise is commonly used in collaboration with agile management [42].

**Number of issues** The most straightforward way of representing the completed work is by counting the number of TD items in the backlog.

The discussion on the pros and cons of the different estimations concerning our artifact is included in the design and development Section 4.1.2.

## 2.4 Literature review summary

Cost savings in software development come at the expense of code quality, which is equivalent to taking on TD. Technical and non-technical stakeholders need to be aware of how much TD they take on by monitoring it continuously. If not, the costs they initially saved can lead to more costs in the long run. To help monitor TD, one can utilize TD management tools. It is desirable to have a tool that addresses all types of TD, such as backlogs. We identified a need for such tools to be improved by using all the information stored.

Having the opportunity to search for previous TD items can provide valuable insight. This information can be analyzed and used to calculate different metrics. e.g., TD velocity. Another advantage of a historical record is that the information can potentially be used to improve principal and interest calculations.

An essential aspect of TD management tools is their ability to visualize TD and thereby improve stakeholder communication. Even though software visualization has been shown to benefit the process of software understanding, there are few tools and research that discuss how it can be done. The most frequently suggested visualization techniques are dependency matrix, bar chart, and pie chart.

Further, since a historical record will contain backlog items, we can borrow visualization techniques used in project management, e.g., the burndown- or burnup chart. However, one needs to stress the visualization aspect when developing a historical record.

In Chapter 4, we will build a historical record that transforms repaid TD data into visual representations. Finally, in Chapter 6, we will discuss how the historical record can be included in the management of TD and how it can improve communication between non-technical stakeholders.

## 2.5 Tool survey

This section will outline some of the most used management tools used for TD. Martini et al. [33] did a study on the current state of practice of TD tracking, where one of the research questions was to find out what tools are used to track TD. We will use the results from this question as a starting point for this survey. We searched for the specific TD features each tool offered and (if anyone) historical recording of (TD) backlog items.

There exist many different types of tools for managing TD. They differ from each other by using different terms, metrics, features, and ways to identify and measure TD. The most used and effective tools are currently backlogs, and static analyzers [33].

Among the static analyzers, one has tools like SonarQube, SonarGraph, and Klockwork. The downside of using static code analyzers is the danger of leaving aside large amounts of potential TD. TD types like structural debt, architectural debt, and technology gaps are undetectable by such tools.

The other most used category of TD management is a backlog in project management tools, which includes either a dedicated backlog of TD issues or a usual feature backlog where TD items are mixed with features [33]. These tools allow one to track several types of TD, although not automatically. Examples of backlog management tools are Jira, Trello, Hansoft, Excel, and AnaConDebt.

There exist many different static code analyzers but not many tools to manage TD in the backlog. Due to this, we will not focus the survey on static analyzers.

### 2.5.1 Jira

Jira is a project management tool tailored to agile software development teams. In Jira, TD can be included items in the backlog by specifying the issue type as TD. In this way, they are visualizing and keeping track of TD. Completed items are recorded and enable some functionalities based on this. For example, it is possible to add a ″gadget″ called ″Created vs. Resolved″ to the project dashboard (See Figure 2.3). This gadget shows a chart of completed and created TD items, which helps the users see how things look in ″the bigger picture.″ The chart can also be filtered by variables like time created/resolved and issue type [38]. Additionally, Jira has something called a ″Velocity Chart″. This chart presents the number of items resolved vs. the total items assigned for each sprint. This data is used to calculate average velocity [4].

### 2.5.2 Trello

Trello is a lightweight project management tool that organizes projects into boards, lists, and cards [49]. These elements can be organized in any way that suits the team, for example, by visualizing a backlog as a list with TD cards. The tool is known as one of the most accessible platforms to understand and

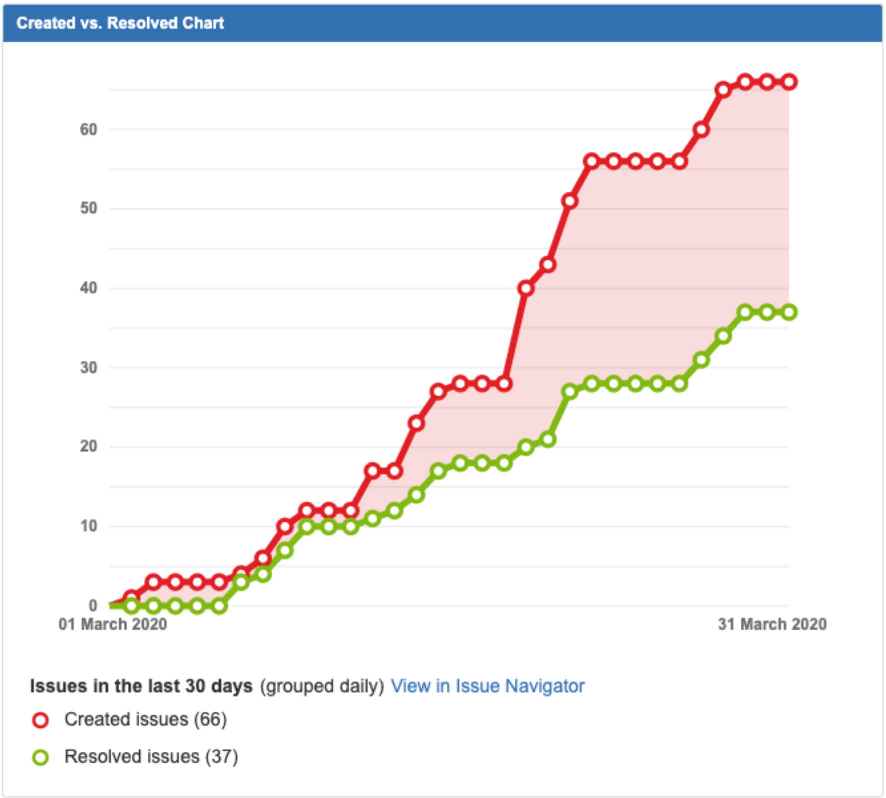


Figure 2.3: Created vs. Resolved [39]

use. In Trello, repaid TD item cards can be moved to their own ”done”-list. One can also archive the items from here. The archived items are removed from the board, and searchable in the archive.

### 2.5.3 AnaConDebt

There also exist project management tools dedicated to TD. One example of this is AnaConDebt which also tracks and assesses TD Items through a backlog. The backlog allows the creation of TD Items and performing TD-specific operations on the items, which are currently not available in other existing tools like, for example, Jira [32]. The tool also helps assess the cost-benefits of repaying the debt and calculating and communicating TD.

### 2.5.4 Choosing a tool

All the tools described above offer the basics we need to develop the artifact, namely a backlog with TD elements. Trello is a tool we are well acquainted with from previous project work. Jira and AnaConDebt, on the other hand, are tools we have not used before and will therefore require more time to get acquainted with. Trello also offers several more minor functionalities that can make it easier to get acquainted with their API. Unlike the other tools mentioned, Trello is also a free and publicly available offer. Given these benefits described, it became a natural choice for us to choose Trello as a tool to integrate with the artifact.

Besides these reasons, the scope of the thesis is to build a prototype artifact. Trello is a more accessible tool to use but complex enough to provide the features we need, and therefore it seems like the optimal choice.

# Chapter 3

# Methodology

This Chapter will describe the methodology used to guide our research. First, we will describe the methodology of design science research and how our process aligns with such a process. Then we will explain the data collection methods used in the research and how it was conducted. Next, we will explain the techniques used for data analysis. Lastly, we will reflect on the research limitations of the methodology chosen.

## 3.1 Design Science Research

The overarching methodology chosen for the thesis was design science research (DSR). DSR has received increasing interest in the last years and is now adopted by scholars and design practitioners from a variety of areas such as information systems, computer science, medical informatics, and software engineering [22].

DSR is rooted in the science of the artificial. In contrast to natural science, where the focus of the study is physical and abstract entities, the artificial science study is dedicated to entities (or artifacts) developed by humans to provide some functionality to solve limitations or problems [13].

In short, DSR is inherently a problem-solving process that seeks to enhance human knowledge via the creation of innovative IT artifacts [7]. The ultimate goal of DRS research is to prove the artifact's utility. The concept of 'IT artifacts' broadly refers to constructs, models, methods, and Instantiations [22].

The goal of our research was to find out how visualizations of historical data can improve the management of TD. To reach this goal, we needed to design and build a new and innovative artifact that could visualize the historical data. Therefore, we were confident that DSR was the right approach for our research. Furthermore, DSR offers the right tools to help guide our research so that we reach the goal of an artifact that fulfills its purpose of improving TD management.

The artifact we aim to build can be characterized as an instantiation. An instantiation is the realization of the artifact in its environment [31]. We can achieve this by designing and building a website that works with an existing project management tool (Trello). Instantiations are often used for specific, limited, and less mature knowledge [22]. This description is consistent with the field of our research.

### 3.1.1 Research process

Peffers et al. [35] propose a DSR process that consists of 6 steps: problem identification and motivation, the definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. In Figure 3.1 we present how the DSR process was applied in our research. Furthermore, we will explain each process in more detail below:

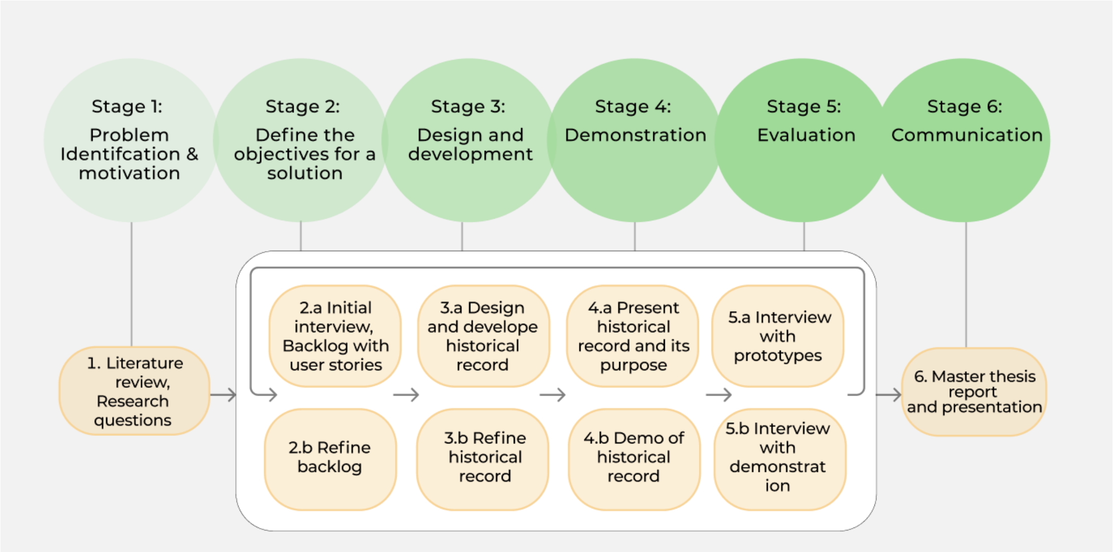

Stage 1: Problem Identifcation & motivation
Stage 2: Define the objectives for a solution
Stage 3: Design and development
Stage 4: Demonstration
Stage 5: Evaluation
Stage 6: Communication
1. Literature review, Research questions
2.a Initial interview, Backlog with user stories
2.b Refine backlog
3.a Design and develope historical record
3.b Refine historical record
4.a Present historical record and its purpose
4.b Demo of historical record
5.a Interview with prototypes
5.b Interview with demonstrat ion
6. Master thesis report and presentation

Figure 3.1: DSR Process

**Problem identification and motivation** The identification of the problem and the motivation to solve it is described in the introduction. The activity involved in this stage was to conduct a literature review of topics related to the research questions. The results from the literature review are described in the background, Chapter 2.

**Define the objective of a solution** In this stage (2.a), we conducted data collection in the form of interviews to explore the problem area where the phenomena of interest reside. The objectives were defined by specifying a backlog (see Table 4.1). Each objective is inferred rationally from the literature review and the results from the data collection. In step 2.b, we refined the backlog based on the feedback we received from the evaluation in step 5.a.

**Design and development** In this stage, we designed and developed an artifact. The stage includes all activity that involves determining the artifact's desired functionality and architecture and then creating the actual artifact. We started this process (3.a) by developing instances of the artifact (prototypes) based on the backlog of user stories. Then, in step 3.b, we refined the artifact by designing and developing all features based on the refined backlog.

**Demonstration** This stage involves demonstrating how the artifact solves one or more instances of the problem. The purpose is to establish that the artifact works. In the first iteration (4.a), we conducted an interview and presented different prototypes and their purpose. We made a presentation with a slide for each visualization to do this. This demonstration was a suitable approach early in the process, as it was quick and straightforward to organize. In the second demonstration (4.b), we did a live demonstration of the artifact. Instead of explaining the purposes of the charts, we just described their objectives to let the participant make up their own opinions on usefulness. Both of the demonstrations were conducted with relevant stakeholders.

**Evaluation** The evaluation stage is tightly coupled with the demonstration. In addition to showing how an artifact works, we also want to examine how well it works. The feedback from the interview is compared with the relevant objectives of the solution (stage 2). To improve the solution's usability, we decided to iterate back to stage 2 of the process. This iteration is represented with the letter b in Figure 3.1.

**Communication** The last stage after the iterating build-and-evaluate stages is communication. The resulting knowledge from the research is reported in this thesis.

### 3.1.2 DSR Evaluation types

The evaluation activity is a crucial component in DSR as it provides feedback for further development and assures the rigor of the research. When deciding which evaluation methods to use, one must consider the state of the designed artifact and the selected evaluation metrics. To help guide these decisions, Sonnenberg and Brocke [46] have identified four types of evaluation derived from the DSR activities. As illustrated in Figure 3.2, an evaluation should follow the four key activities: problem identification, design, construction, and use. Further, they categorize the activities into two distinct phases: ex ante and ex post evaluation.

Ex ante evaluation provides models for theoretically evaluating a design without implementing the material system or technology. From the perspective of DSR, we say that the evaluation takes place before the IT artifact is constructed. This phase can also be categorized as design research [37].

The activities related to this phase are the problem identification and design activities. The purpose of ex ante evaluation is to validate the prescriptive knowledge before it is applied in practice, to avoid the risk of devoting a significant amount of time to building insignificant solutions to practical problems [47].

On the other hand, ex post evaluation takes place when a chosen system or technology is acquired or implemented after the instantiation of the IT artifact [37]. The construction- and use activities are a part of this phase. The use-activity refers to the artifact being fully embedded within the organizational context. In the following section, we will describe where our evaluations can be placed according to these types.

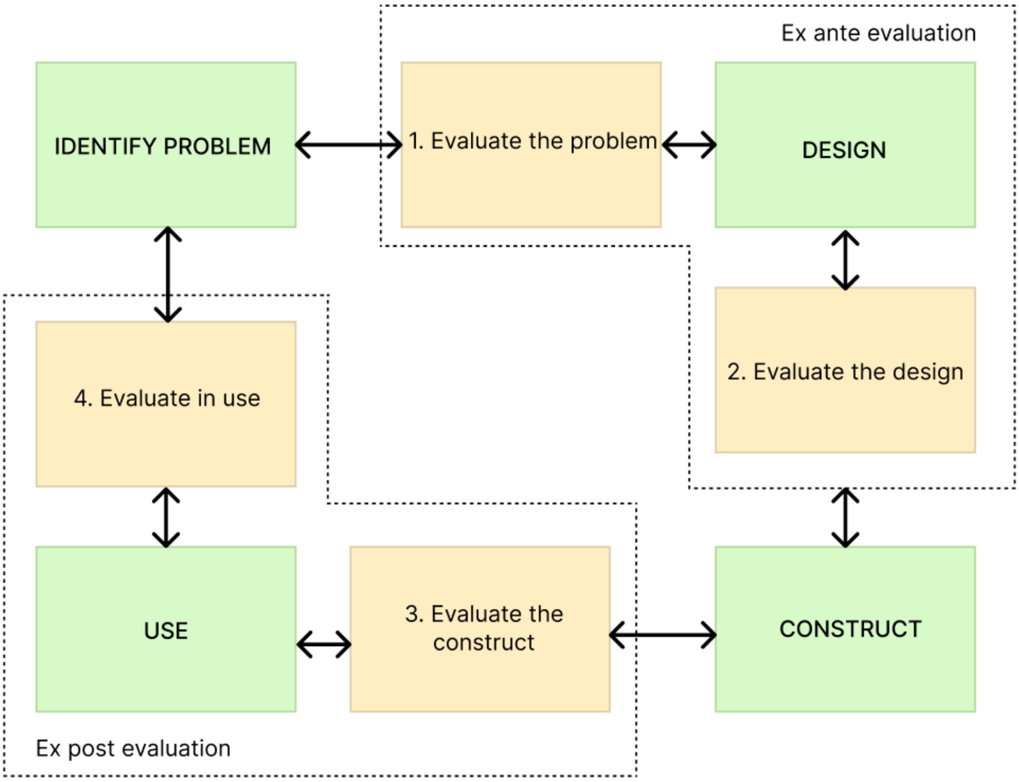


Figure 3.2: DSR Evaluation Activities adapted from Sonnenberg and Brocke [46]

## 3.2 Data collection methods

This section will describe the different data collection methods used in the research. There is a diverse set of research methods one can choose to apply in a DSR study. These include well-established methods in social science research, such as interviews, surveys, literature reviews, or focus groups [7]. We have primarily used interviews as method for the different data collections conducted in our research. Either as a methodology alone or in combination with other less formal methodologies. First, we present the reasoning's and the procedure conducted for the literature review. Then, we explain interviews in general in relation to our research. Lastly, we explain the specific interview data collections conducted.

### 3.2.1 Literature review

The first research method applied in the research process was the literature review. The literature review was conducted to familiarize ourselves with the problem domain and identify other research fields that could be relevant to the domain. The results from the literature review can be seen in Chapter 2, the background.

To conduct the literature review, we followed some of the same guidelines used in a systematic literature review [23]. One of the guidelines is to have a defined search strategy when planning the review [23]. This involved defining search terms and resources. A second guideline we followed was to determine the criteria for including in or excluding a study from the literature review [23].

The relevant keyword used in Section 2.1, ″Technical debt,″ was ″technical debt″ and ″systematic literature review.″ In this section, we only included primary studies mentioned in systematic literature reviews. In addition, we excluded studies where the scope went beyond the concept of TD.

For Section 2.2, ″Analysis of historical record,″ keywords used were ″Technical debt,″ ″historical data/record,″ ″analysis,″ and ″technical debt management.″ In this part, we extended the scope of the study to include studies that dealt with project management in general.

Lastly, the search string used for Section 2.3,″ Visualization of historical record,″ was″ technical debt,″ ″historical record,″ and″ visualization.″ We also used different synonyms for a historical record, e.g., historical data, record, archiving, completed items, and finished items. Also, here we included studies where the scope of the study was about project management.

The databases/resources used for data searching were mainly Google Scholar and The University of Oslo Library. Using these resources to retrieve data, we can be more confident that the sources are of good quality rather than through a simple internet search. Furthermore, in the parts where we talk about specific tools, we retrieved data from their respective documentation.

**Evaluation Type** The literature review can be defined as the first evaluation activity according to the DSR evaluation activities (see Figure 3.2). The DSR process started with a problem observed in practice. The literature review served as an activity to justify the problem statement and justify the engagement in the DSR project.

### 3.2.2 Interviews

Interviews are a qualitative data collection method that is a good source of in-depth knowledge. The ability to ″go deep″ is the strongest argument that makes interviews suiting for novel research fields such as ours. The interviewees are given space to reflect and evaluate questions, which can generate new ideas and insights that other methodologies, such as e.g. surveys, cannot. In addition, interviews are suited in many stages of the development process, such as initial explorations, requirements gathering, and evaluation of new artifacts [26].

Interviews can be structured in three ways: fully structured, semi-structured, and unstructured. All of the interviews conducted in our research were semi-structured. This structure allows for clarification, adding questions, or following the participant's comments wherever they lead us. The goal of such interviews is to dig through the interviewer's comments and look for possibilities to gain additional insight and understanding beyond the predefined questions [26].

To prepare for the interviews, we did pilot testing beforehand. The pilot interview helped us address the understandability of the questions and gave us an idea of how long the interviews would take. If the interview lasted more than an hour, we tried to cut some of the questions. Another preparation we did was to make interview guides. The preparation actions were essential for our research since there was only one researcher to conduct them.

Another preparation we did for the interviews was to make a consent form (see Appendix A) explaining what the purpose of the data collection was and what we will use the findings for. For the evaluations, we included this information in a presentation slide. The consent form also specifies participation is anonymous and that they can withdraw at any time. We included a check-box the participant's could tick, to see if it was okay for us to record the interview. In evaluations or digital meetings, we asked the participants directly.

### Initial interviews

The first data collection was conducted in the problem identification and motivation phase. The purpose of the interview was to map the business needs in the environment related to our research. The goal was to gain insight into how an organization manages TD. More specifically, we wanted to find out what information related to the repayment process of TD could be valuable for non-technical and technical stakeholders.

By following a semi-structured interview, we defined a few high-level topics that we planned to cover in the interview. The topics are included in the interview guide (see Appendix B). The topics we defined were derived from the literature review and the research question. Here we explain the reasoning behind each category:

1. *Work and role*. The goal here was to get to know the participant.

2. *TD in the work-space*. The goal here was to understand the subject's understanding of "technical debt" and how it is handled (if it is) in their organization.

3. *Cost-benefit analysis*. Here, the goal was to understand how they prioritize their debt (if they did) and explain how it could be done. Further, get some input on what they think of cost-benefit as a way to prioritize or other useful aspects of this process.

4. *Burndown chart and Created vs. resolved chart*. In these two categories, we presented two examples of visualizing TD. We wanted to map the participant's knowledge of them and explain if anything was unknown. Such external aids are known as "probes" and are included to promote engagement [26]. It helps the interviewee gain more understanding and insight into the topic. Furthermore, we asked what value the visualization could provide them and possibly what aspects could have been useful.

5. *Historical Record*. In this part, we wanted to explain the artifact we wanted to develop and the preliminary ideas. Then the interview objects could provide input on the ideas and come up with their suggestions if they had any.

**Evaluation Type** The results from this interview further validated the problem solution. Therefore, we can categorize this interview together with and literature review as the first ex ante evaluation "evaluate the problem," as seen in Figure 3.2.

**Participants** We conducted two interviews. The first interview subject was a senior developer with much experience with legacy systems. The second interview subject was a team leader in a team working with integration with cloud-based solutions. Cloud technology is fast-growing, exposing it to frequent changes, making its work prone to TD. Both were consultants working in the bank domain.

### Interview with prototypes

As part of the first iteration, we combined the methodologies interview and prototyping. We chose to make prototypes because we wanted to explore requirements and specific details before staring developing [9]. The prototypes were the objective of the evaluation.

The prototypes we made can be specified as ″rapid prototyping.″ Rapid prototyping refers to the creation of a model that will eventually be discarded rather than becoming part of the final delivered software [48]. In our case, the prototypes are rapid sketches of what the user stories may look like when they are implemented in the system. We describe each prototype in the following chapter. Evaluating the solution with prototypes is suitable at such an early stage of the research since it can be carried out fast with little resources.

The evaluation was conducted by first explaining the purpose of the prototypes, followed by asking the participants about their initial thoughts on usefulness. The participants were also encouraged to ask questions regarding the prototypes. Evaluation at this early stage tends to be more informal than later evaluations. Therefore, we did not provide a formal preparation. Instead, the presentation slides worked as an interview guide, and the criteria were general use.

**Evaluation Type** We specify this evaluation as an ex ante evaluation. We had not started to implement the artifact and wanted to decide whether or not to acquire or change the user stories related to the visualizations. In this case, we are evaluating the artifact's design, which refers to the second evaluation type seen in Figure 3.2. We are beyond the problem identification, and the purpose is to evaluate the solution design based on the design specification (backlog).

**Participants** We conducted one evaluation of the prototypes with two participants. They both had expertise in the field as they are working for ACDtek, a company delivering software solutions for TD management. Their respective roles in the company are CEO and CTO.

### Interview and demonstration

We combined the interview methodology with a live demonstration in the second and final evaluation. The objective of the evaluation was the implemented artifact. Further, the purpose of the evaluation was to gain insight into the artifact's usefulness from the perspective of technical and non-technical stakeholders. As we are still early in the development process, we refer to our artifact as a prototype. The artifact prototype mainly consists of visualizations with few interactive interfaces. Therefore, it was natural for us to get involved in the activity through a live demonstration.

The demonstration process started with an introduction where we explained the aim of the study. Further, we explained the assumptions for our solution and the purpose of the evaluation. The demonstration continued by showing how the artifact would work in its natural environment as an extension to a TD backlog. We demonstrated how the artifact was integrated and worked with a TD backlog made with the project management tool Trello. Then we explained the functionality that is available in the historical record. We focused on the visualization features, as this is most relevant for both the evaluation and the thesis. For the same reason, the interview questions were also concerning these.

To prepare for the evaluations, we made an interview evaluation guide (Appendix C). The guide includes the demonstration steps and the interview question goals. The goals concerned the information regarding their impressions and experience of the artifact related to the following evaluation criteria:

- **Usefulness** - ”The degree to which the artifact positively impacts the task performance of individuals [21, page 14-15].
- **Understandability** - ”The degree to which the artifact can be comprehended, both at a global level and at the detailed level of the elements and relationships inside the artifact” [21, page 14].

To achieve data related to usefulness, we asked the participants about related attributes such as where they would use it, who would use it, and when they would use it. The understandability measure concerns the participant's comments related to the perceived understandability of the artifact. To encourage these comments, we told the participant to ask questions, tell us their initial thoughts, and elaborate on their opinions on improvements that could be made.

**Evaluation Type** This evaluation can be categorized as an early ex post evaluation, according to Figure 3.2. We are, in this case evaluating a construction (artifact prototype), which refers to the third evaluation type in the same Figure. In contrast to the last evaluation, we have implemented the artifact. Therefore we are beyond the ex ante evaluation. However, we are still early in the development phase, as this is the first evaluation of the implemented artifact. The development process is yet not completed because the feedback can foster changes to the design.

**Participants** In total, we conducted three evaluations. The first evaluation included two participants, the same persons as the first evaluation. The second evaluation was done with a Senior Software Developer. Finally, the third evaluation included two participants with the roles of Chief Software Quality Engineers. The participants in the last two evaluations are working in large software companies.

## 3.3 Data analysis

All of the methods used in our research produced qualitative data in terms of notes and audio recordings. To make sense of this data, we conducted a *thematic analysis*. Thematic analysis is a method for identifying, analyzing, and interpreting patterns of meaning within qualitative data [10].

To use the method, we had first to transcribe the audio recordings. This step also made us familiar with the data set. Then, we read through the notes a couple of times for the cases where we had taken notes instead of recordings. When starting the coding exercise, we had a different approach for the different data sets:

### 3.3.1 Inductive approach

For the initial interviews, we used an inductive approach [10]. By this, we mean that the themes emerged from the data. We started the coding process by highlighting the parts that we perceived as interesting/relevant for the data collection goals. Further, we created different codes to describe the highlighted content. Then we grouped similar codes and made themes that described different problems/needs. The results from this analysis were used to define the initial backlog of user stories. The result are described as motivations in Appendix D. To further illustrate the process include one example. One of the themes from the initial interviews was "Lack of understanding." The following quotes were all coded and grouped under this theme:

- *"A developer would probably look at a library as "expired" after 2 years, while a manager would think 20 years."*
- *"..name of variables did not make sense. Unreadable. But, he made it sick fast. He received a lot of credit for working so fast from the project manager."*
- *"... it is very difficult. Been a theme for many years. To try to convince what needs to be done, but those who are not technical can not keep up."*

This theme was a part of the motivation behind the following user story: *"As a team member/business leader, I want to know the relation between team velocity and the number of repaid TD items so that I can monitor how development speed is affected by the amount of TD."*

### 3.3.2 Deductive approach

We followed the same process described but with a deductive approach for the data that emerged from the evaluations [10]. The established evaluation goals served as the predefined themes for the coding. We chose the deductive approach since the evaluation goals are already structured as we want our final findings to be. An example from the last evaluation of a goal and thereby a theme was "improvements." The following quotes were all coded and grouped under this theme:

- *"Mean value concept ... Don't do that, I don't think that helps. Instead of mean value, why don't you go for a net value? So then this would tell you the difference between TD, either you are adding TD or you could see graphically if you are adding or subtracting to a project."*
- *"If I could get this from let us say last five sprints, it would help to discuss. The more sprints, the better. At least five sprints or more."*

## 3.4 Methodological limitations

This section will go through the methodological limitations identified in our research design.

### 3.4.1 Few participants

Preparation, conducting, and analyzing interviews require, in general, much effort, which limits the number of participants to a relatively small number. One drawback with this is that we have built the visualization with ideas from a few specific persons, while other people could have different needs/ideas. However, we recruited persons with different roles and from four different organizations to mitigate this fact. These different perspectives will then still foster different insights/feedback.

### 3.4.2 Alternative methodology

An alternative methodology for the research could be to conduct a case study. A case study is an in-depth study of a specific instance within a specific real-life context [26]. What mainly differentiates our research from a case study is that we have included different companies. To tailor our study to a case study, we would have had to look at one company and investigate it continuously with multiple data sources (e.g., observation). Instead, we found it more suitable to investigate multiple cases to explore the field and get a broader perspective. However, a case study may be very appropriate for further research.

An alternative to the last evaluation activity could be to conduct a survey. The advantage of surveys is their ability to get many respondents at a relatively low cost. If it is done right, the results can be generalized to the user population [26]. Another benefit of this survey is that the researcher would not be present during the evaluation, making the context more true to its natural environment. The fact that we have two roles, as both the developer and the interviewer, can introduce researcher bias in the evaluation.

However, the benefits described also introduce drawbacks. The data produced by surveys are limited to ”shallow” data. In this research stage, we felt more need to investigate the ”deep” detailed data. Furthermore, we wanted to be present to ask follow-up questions if any interesting phenomena were mentioned. In addition to this, the subject of the thesis, TD management, requires competencies (e.g., from software architects) that a limited amount of employees have. This could make it challenging to recruit the right people for the survey.

### 3.4.3 Partial DSR Process

If we had designed and developed fewer features, we would possibly manage to do more refinement by more feedback loops. We could also have designed and developed more features, but this would maybe result in only one feedback loop. Our research field is novel and requires exploratory work. Therefore it makes sense for us to go ”in breadth” rather than focusing on a few aspects. However, we found a good combination of breadth and depth, allowing us to implement and evaluate the most significant aspects of our solution.

DSR is a comprehensive approach that involves many different phases and activities the researcher should involve in. The ideal outcome of a DSR study is to prove the artifact’s utility in a natural environment, i.e., within the organization. Such activity refers to the last evaluation, ”use,” seen in Figure 3.2. Such activity is critical to embracing all the complexities of human practice in real organizations. Examples of such evaluations are case studies and field studies. The evaluations we conducted involved real users and a real system (for the last one), but whit an unreal setting, which places both of our evaluations in an artificial setting.

A DSR process can have different entry points, depending on the input triggering the interest in the DSR project [47]. Our research process was initiated from a problem identified in practice. This implies that we started the process from scratch, as the problem needed to be explored initially. Considering this and the fact that we had limited time and resources to conduct our research, it would be an unrealistic goal to conduct the last evaluation activity. However, there is no requirement to complete the DSR cycle to conduct a DSR. As long as the process and results are well documented, the contribution can still be valuable in future research.

# Chapter 4

# Iterations

This chapter will describe how the artifact evolved from the initial backlog of user stories to a working website through two iterations. The phases consist of steps 2-5 in our DSR process (see Figure 3.1). We will follow the same structure in this chapter.

## 4.1 Iteration 1: Prototypes of the historical record

This section will describe the first iterations where we defined a backlog of user stories and designed the first sketches of the historical record. In the end, we will evaluate our artifact. These activities are represented in the DSR process with the letter ″a″.

### 4.1.1 Define the objectives for a solution

Before developing the artifact, we made a backlog of requirements formulated as user stories. We have included a simplified version of the backlog in this section (see Table 4.1). The detailed version of the backlog can be found in the Appendix (see Appendix D). The details includes an explanation of the motivations behind each user story, the assumptions for implementing them, and their prioritization (high, medium, low).

These user stories were rooted in the literature review and the initial interview. The roles included in the user stories are team members and business leaders. These are examples of roles we envision will apply to the artifact directly.

We had some additional ideas on features for the historical record that were not yet studied or mentioned in the data collection. So we decided to create some user stories based on these ideas. Which user stories this applies to is evident in the detailed backlog. The nature of DSR makes sure that the features will be evaluated to ensure their research relevance.

### Prioritization and categorization

The user stories were prioritized in the backlog based on their dependencies and the importance of the user stories. The ones rooted in the data collection and the literature review were prioritized over those created without any specific source.

To give more structure to the backlog, we categorized the items in different colors. First, the blue user stories are functions typically for an archive. The process of archiving is to transfer data to a less frequently used storage medium [2]. Functionality related to this is moving, searching, storing, and restoring items from the archive.

The purple user stories are related to each repaid TD item and what information they should hold. In addition to the rationale included in the backlog, we have also included the properties that are included in Guo et al. [20] example of a TD item, which are date, responsible person, type, location, and description.

Lastly, the green user stories are related to information about resolved TD items in total and other relevant metrics. The relationships give the user a more complex picture of the items in visualization and analysis.

The categorizations are used as a rough prioritization of the items since the general archiving functionality needs to be in place to incorporate the items. Furthermore, information related to each item must be in place to discuss their relationship.

|  | Backlog of user stories |
|---|---|
| 1. | As a team member, I want my debt items marked as done to be available in an archive, so that the repaid data is accessible for later use. |
| 2. | As a team member, I want to look up and search for repaid debt items in an archive so that I can find specific repaid debt items at any time, for example for reporting reasons. |
| 3. | As a team member, I want to restore a repaid debt item from the archive so that I can modify/redo the missing functionality. |
| 4. | As a user, I want to know at what time a debt item was created and marked as done so that it can be sorted chronologically. |
| 5. | As a team member/business leader I want to know the time used to repay an item, for each debt item, so I am aware of how resources are used. |
| 6. | As a team member/business leader I want to know what was the estimated time to solve the task, for each debt item, so that I can compare it to the actual time used. |
| 7. | As a team member/business leader I want to know the expected and actual cost-benefit of solving a debt item for each debt item so that I know if we solved the right item. |
| 8. | As a team member, I want to know who was responsible for repaying an item, for each debt item, so that questions or information related to it can be asked to the right people. |
| 9. | As a team member, I want the items in the archive to be categorized by type, module, and size of the debt item so that I can differentiate them and compare them. |
| 10. | As a team member/business leader I want to see a visual overview of repaid debt items in the archive so that I get a holistic overview of payment progress. |
| 11. | As a team member/business leader I want to know the total number of TD items solved for each TD size, for a given period, so that I know how much resources are used on TD. |
| 12. | As a team member, I want to have a predefined/optimal goal for number solved debt items for a given period, so that I get motivated to achieve this goal together with my team. |
| 13. | As a team member/business leader I want to see a visual overview of the average total number of TD items solved, for each TD size, for a predefined interval (for example a sprint). |
| 14. | As a team member, I want to know the anticipated principal for a TD category (type/location), so I can improve the accuracy of the cost-benefit analysis for debt items in the backlog. |
| 15. | As a team member, I want to know the anticipated interest of a TD category (type/location), so that I can improve the accuracy of the cost-benefit analysis for current debt items. |
| 16. | As a team member/business leader I want to know the relation between team velocity and the number of repaid debt items so that I can monitor how development speed is affected by the amount of TD. |
| 17. | As a team member/business leader I want to see the relationship between created vs. resolved debt items, so I know if there is positive or negative progress. |
| 18. | As a team member/business leader I want to see the total risk reduced for each sprint compared with the total risk left in the system, so I get an overview of risk progress. |

Table 4.1: Initial Backlog

### 4.1.2 Design and development

When deciding how to visualize the historical data, we got inspiration from the techniques used in agile development, especially the burnup chart (see Figure 2.2). This chart is explained in the background (Chapter 2). The burnup chart covers many aspects of what we want to achieve with our historical record. The main difference is that we are only interested in the TD items instead of looking at a whole project and its backlog items.

#### Defining completed work

Further, we need to agree on a precise way to define ″work completed″ as representative of our context. First, visualizing how many TD items have been resolved will not indicate how much TD has been resolved in total since the size of the debt is such an essential factor. To illustrate, replacing a non-optimal library can be more time consuming than documenting a class thoroughly. The users will not be able to separate a sprint where there are only repaying small TD items from a sprint where there are only repaying large TD items. However, the latter is much more time-consuming. Therefore, we must find other ways to visualize this by considering the size.

**Story points** One of the developers we interviewed initially used story points as estimating in the team. One disadvantage with using story points is that an estimate often can be quite different from the actual effort, giving the users a wrong perception of complete work. In the agile charts, the estimated effort remaining is also used to capture the actual effort expended. However, more precise measures should be considered in our context since it would improve the quality of the historical record and its features.

One solution to this problem is that the story point estimate is corrected after completion. However, the downside with this is that it requires more manual work by regular follow-up of each repaid item to ensure an accurate value.

**Calculating the time used** Therefore, we have looked at another solution to define ″work completed″ by calculating the time used to solve each item. The calculation of ″time used″ can be done by using the property ″time created″ and ″time-resolved.″ The assumption here is that the team continuously manages the backlog. So that when an item is moved from ″in progress″ to ″done,″ it will be an accurate measure of time used to fix the item.

The automatic calculation is a more fragile solution since users can forget to specify an item as ″in progress″ and ″done,″ requiring much manual work. Furthermore, we need to have the same values to compare this metric with the unresolved TD. Either the team would need to use the number of hours to estimate TD items, or we would need to translate the hours used into the estimate that the team prefers. The advantage of this solution is that it will be done automatically, with the precaution that the team handles TD items in a time-consistent way. This discussion on the choice of metric is included in the evaluation to figure out what the users prefer.

Additionally, we also want to include time in the x-axis. One idea here is to visualize the different sprint intervals with stippled lines. By taking these considerations into account, we made five different chart prototypes which are each described in the following sections:

### Chart 1: Technical Debt Progress

The user story related to this chart is the following:

> ”As a team member/business leader, I want to see a visual overview of the total effort on TD items repaid, for a given period, so that I know how much resources are used on TD. ”

The chart in Figure 4.1 is very much aligned with the burnup chart. The only difference is that the line that is supposed to be showing the ”target” as in the burnup chart is omitted. We did this because, in backlog management, the goal is completion, which means that all items are solved. Completion is not naturally the case for TD management. We want to limit TD to a manageable level but not necessarily pay back all debt.

It can be valuable to include more detail in the visualization for teams or smaller projects where the total number of TD items is low. Therefore, we also made a prototype including TD types, as seen in Figure 4.2.

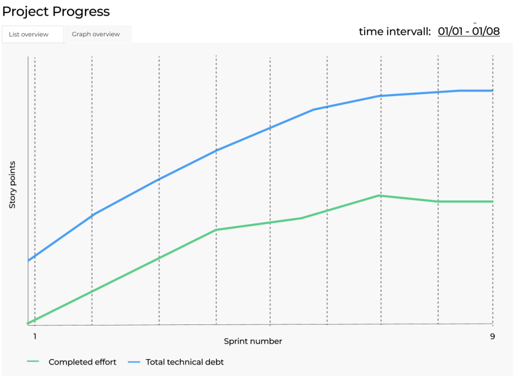


Figure 4.1: Prototype: Technical Debt Progress

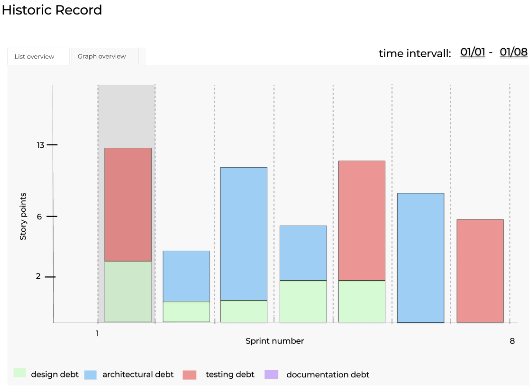


Figure 4.2: Prototype: Technical Debt Progress including Type

**Chart 2: Identified- vs repaid items**

The user story related to this chart is the following:

> ”As a team member, I want to see the relationship between created vs. resolved TD items, so I know if there is a positive or negative progress.”

This chart (see Figure 4.3) builds on the same idea as the user story mentioned above. The difference is that we want to simultaneously include the progress of new debt in this chart. The purpose is to see if the team keeps track of the repayment compared to what is identified—the inspiration from Jira’s created vs. resolved gadget. The chart is mentioned in the tool survey part of the background (Chapter 2).

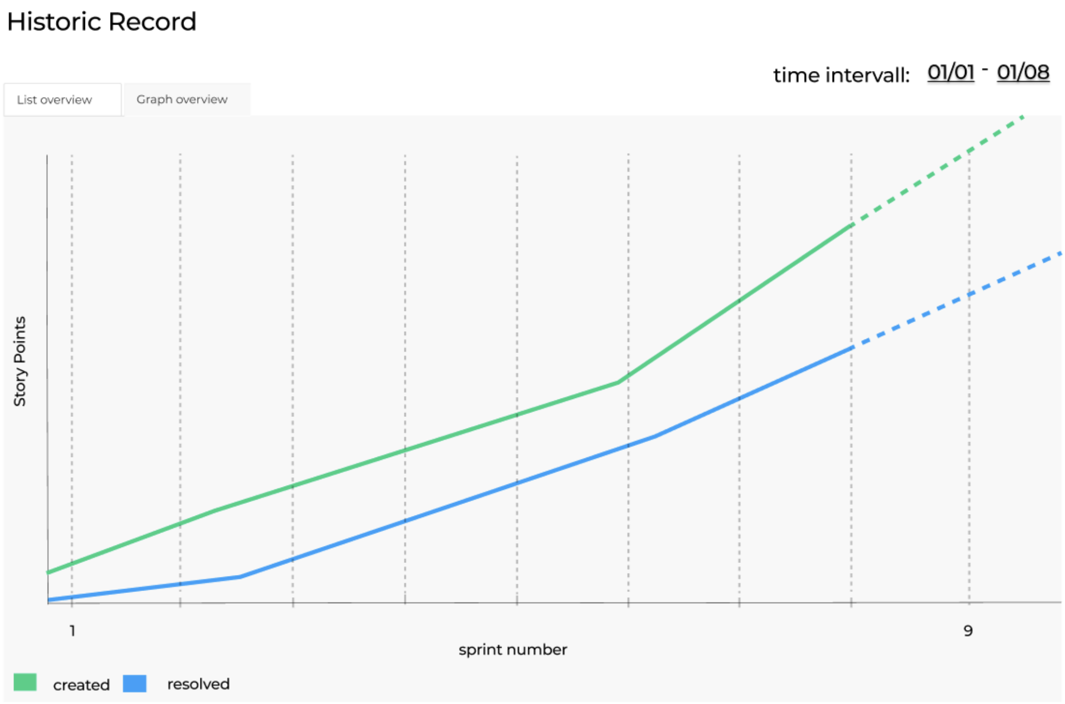


Figure 4.3: Prototype: Identified- vs repaid items

### Chart 3: Velocity vs. Technical Debt

The user story that is related to this visualization is:

> ”As a team member/business leader, I want to know the relation between team velocity vs. the number of repaid TD items to be more aware of how TD affects development speed.”

We want to show how team velocity is affected by the amount of TD. As mentioned in the background chapter, the short-term result of solving TD is fewer features and a lower velocity number. By visualizing the velocity - and TD relationship over a long period, we can provide the user insight into the long-term benefits. That is a reduced total of TD and positive effects on velocity.

This user story assumes that team velocity is reported for a specified interval in the backlog (for example, for each sprint). One adds up the total estimate of all completed items when the sprint ends, divided by the number of sprints. The total TD needs to use the same measure (story points) so that we can compare them to each other.

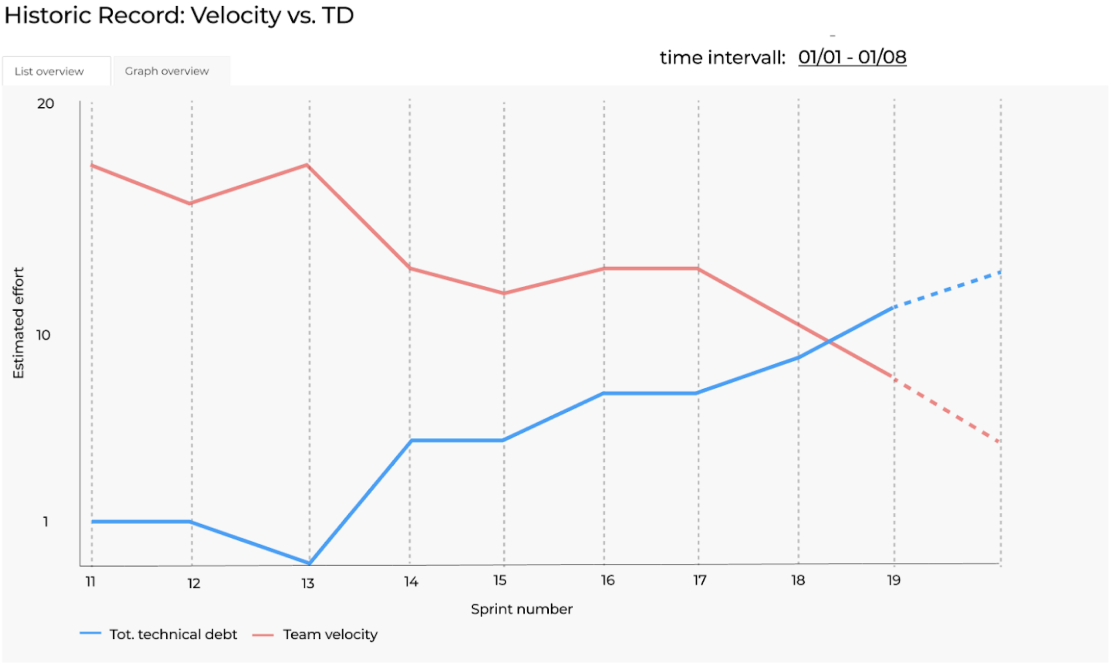


Figure 4.4: Prototype: Velocity vs Technical Debt

### Chart 4: Project Risk

The user story that is related to this visualization is the following:

> ”As a team member/business leader, I want to see the total of risk reduced for each sprint compared with total risk left in the system, so I get an overview over risk progress.”

We could visualize project risk by summing up the risk for all items completed in each sprint. Then the team gets an overview of the total reduced risk for a given time. Additionally, we could add the total unresolved risk to enhance the visualization to form a proportional impression.

We also want to include an acceptance threshold related to how much debt a team can manage. Visualizing the relationship between risk and this threshold value over time can be useful for technical and non-technical stakeholders. On the one side, to make them more aware of the risk they have produced, and on the other side, what trend they are currently moving towards. The team can aim to stay below this threshold and look at this as a common goal. Therefore, we could also incorporate the following user story here:

> ”As a team member, I want to have a predefined/optimal goal

> related to the amount of effort needed to repay TD items for a given period to get motivated to achieve this goal with my team."

We can include this as a "top line," visualizing how close/far the team is to its threshold. Additionally, each team should specify the value of the threshold themself since all teams work differently.

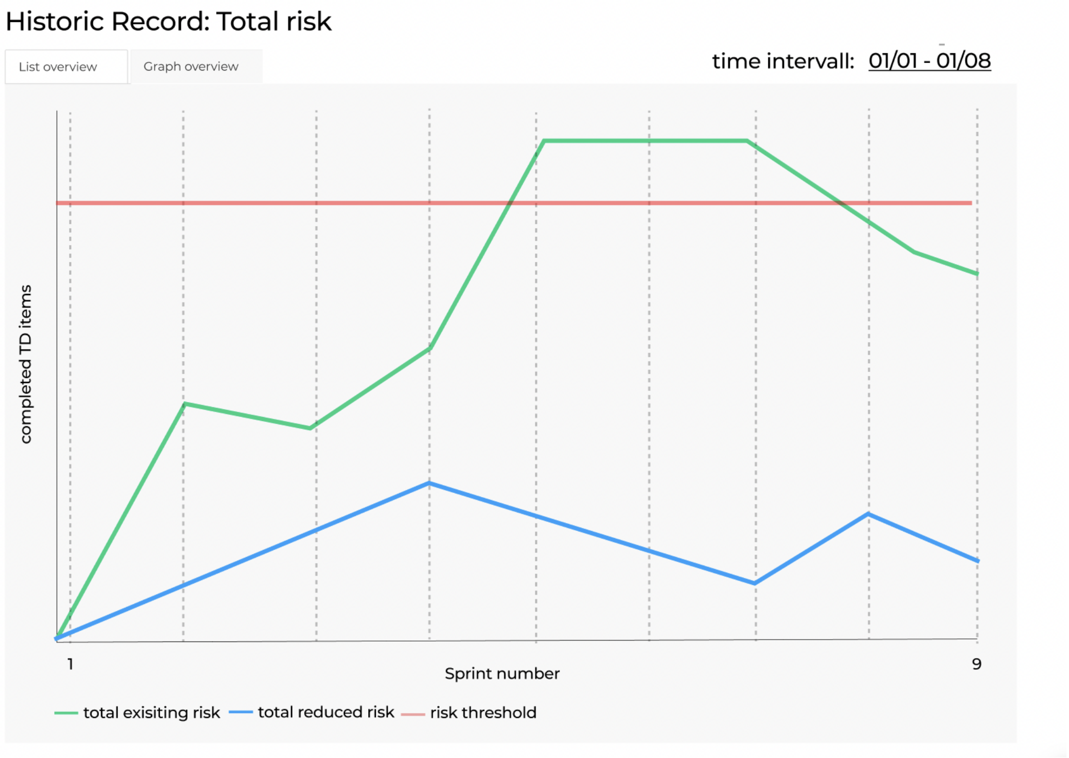


Figure 4.5: Prototype: Project Risk

### 4.1.3 Evaluation

For each chart, we will present the feedback from the interview. Some of the feedback addressed areas beyond the prototypes we had proposed. As a result, this led to new user histories, which we will discuss in the next iteration when we refine the backlog.

**Defining completed work**

The two metrics were explained together with the advantages and the disadvantages. Afterward, we asked about their thoughts and which ones they would prefer. We discovered some new disadvantages related to the metric ″time used.″ One of the interviewees commented the following:

> ″I would go for the estimated effort (story points) - since it is normal to start on a task but not complete it all if you already have put it in progress, the time would be wrong. As a developer, it would be difficult to make it work in practice.″.

Another disadvantage the participants addressed was that the recording of time is in one sense ″tracking the developer,″ which some developers can perceive as annoying.

The other metric, ″story points,″ was not perceived as perfect. They further explained that it is better to set the time/effort oneself instead of ″adjusting″ a value that's already defined. However, depending on manual values is a complicated matter since the artifact is very much dependent on these values to work. A concluding note from the participants was that the more alternatives the better.

Further, we followed the subsequent structure in the interview for the prototypes: First, we explained the purpose of the charts and presented the prototypes, as seen in Figures 4.1-4.5. We would discuss this further if this explanation led to some questions/unclarity. Lastly, we asked if they thought the solution was useful.

**Chart 1: Technical Debt Progress**

The interviewees were questioning that in the chart excluding types (see Figure 4.1), the line representing total TD will continue to grow. Therefore, the aggregating aspect of the chart was not perceived as useful. In this case, the new TD is added to the existing TD over time. The participants agreed that it would make more sense that the line represented the total TD left in the system at each sprint. Then the line of total TD would decrease and increase depending on how much debt is solved. The same applies to the line representing repaid effort. Instead of adding the value to the total for each sprint, the line should represent repaid debt solved at each sprint.

The interviewers expressed that including TD type distribution could be useful (see Figure 4.2). They further elaborated this by expressing that it is good to see where the effort has been put in the earlier phases.

One of the interviewers also proposed an alternative chart to give more information about the progression—a stacked bar chart as earlier, but with negative and positive values. The positive values represent repaid debt, and the negative values represent new debt. We can also visualize the mean between these two opposite poles in an informative way as a separate line. This idea is included in the next iteration.

### Chart 2: Identified- vs repaid items

One of the interviewers asked how this chart differs from the Technical Debt Progress chart. The answer is that we only show the new TD added to the system for each sprint instead of all the unresolved TD in the system. However, this indicated that we maybe could try to visualize it better.

### Chart 3: Velocity vs. Technical Debt

The interviewees agreed that this visualization is useful. Regardless, there were some questions regarding the velocity calculation if this included TD items. We understand that this must be clear to the user whether TD is included or not when we have calculated the velocity. When asked which one they would prefer, they expressed that both variants can be desirable.

### Chart 4: Project Risk

For this visualization, one of the interviewers problematized that ten items with risk 1 are not the same as one item with the risk of 10. Moreover, it would be confusing to sum up, the risk as if it were the same. Therefore, a solution was also proposed, which involved making the total risk biased to higher numbers. Apart from this, the other aspects concerning reduced risk and a threshold were understandable and useful.

## 4.2 Iteration 2: Implementing the historical record

In this phase, we wanted to refine the artifact based on the feedback from the evaluation. So, instead of making new prototypes, we implemented a website of the historical record. The technology used is described in Chapter 5. In addition to the user stories for visualization, we tried to implement all the other user stories in the backlog to end up with a complete artifact. These activities are represented in the DSR process with the letter ”b”.

### 4.2.1 Refine the objectives for a solution

**Defining completed work**

Based on the discussion from the evaluation on how to define completed work, we believed that using story points as measurement was the best solution. We should also include functionality that lets the user change the story point if the estimation is wrong. We decided to keep the ”time used” calculation as a TD item property, so it can be used as a possible reference point when changing the estimate. We changed the user stories in the backlog by replacing the phrasings including ”time used” to include story points.

**New user stories**

While discussing the Technical Debt Progress chart (including type), the interviewees emphasized the benefits of showing how the team's effort has been distributed over time, and that visualization on this is very useful, especially from a business perspective. However, we were left with the impression that additional information about the distribution of efforts could be included.

**Distribution of Completed Work** As mentioned in the background (Section 2), we found a paper that discusses the visualization of the TD effort: Power [36] discuss the use of a pie chart to represent the capacity invested in reducing and managing TD relative to the other areas the team is spending their time. The idea originates from a study of how TD is part of planning and retrospective activities. Reusing the pie chart in our study would be a useful contribution to finding out how useful the chart is from a long-term perspective and as a part of a tool. Therefore, we added a new user story to the backlog:

> ”As a team member/business leader, I want to see a visual overview of the average effort on repaid TD items compared to the team's total capacity for a predefined interval so that I get more insight into how the resources/efforts were distributed” .

This user story assumes that the team specifies the effort used on other activities in addition to TD repayment.

**Distribution of TD effort for each module** We could also include a pie chart that visualizes how the effort is distributed among the different class modules.

The static management tool SonarQube has functionality that is similar to this. However, they present the system's file structure and show how many percentages of different types of code debt are included in each file. The code debt types are bugs, vulnerabilities, code smells, security hotspots, coverage, and code duplications. In SonarQube, the debt in code is automatically identified, making it possible to include detailed information about the location. Since we want to include all kinds of debt, also the ones that need to be mapped and described manually, we need a less extensive property for each item. Based on this, we added a new user story added to the backlog:

> ″As a business leader/team member, I want to see a visual overview of the distribution of TD effort for each module so that I get more insight on how the resources/effort were distributed.″

This user story assumes that the team specifies which module each TD item belongs to.

### User stories that were down-prioritized

We did not manage to implement the user stories that had to do with cost-benefit analysis. These were down-prioritized while working with the backlog. The reason for this was that they were more or less detached from the rest of the user stories, in the sense that they involved new TD item properties which were not related to any of the others.

In addition to this, we had implemented many new features at this point. We had a desire to keep the evaluations lasting less than 60 minutes. It would have been challenging to keep this time if we had added more features. Therefore, we wanted to prioritize evaluating these instead of implementing all of the items in the backlog.

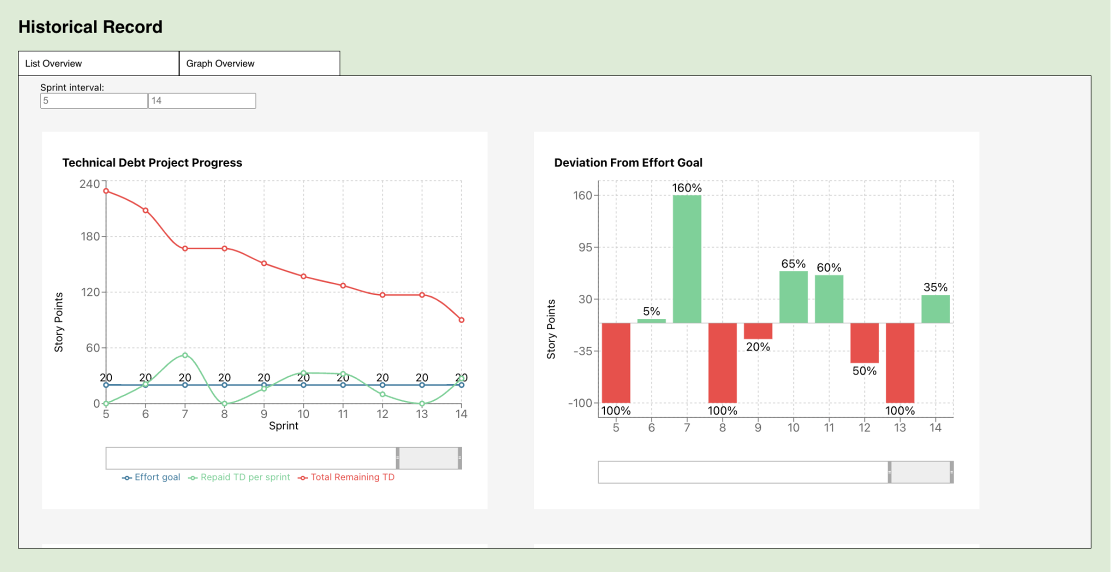
### 4.2.2 Design and development

**Chart 1: Technical Debt Progress**

We decided to include the following user story in the Technical Debt Progress chart:

> ”As a team member, I want to have a predefined/optimal goal for number solved TD items for a given period, so that I get motivated to achieve this goal together with my team.”

We did this by including a line for ”effort goal.” (see Figure 4.6). This user story is already included in the project risk chart (as the threshold), but we figured that not all teams deal with risk, and therefore we found out that it could be useful to include it here. We also wanted to extend this functionality by including a new chart which we called ”Deviation From Effort Goal (chart to the right in Figure 4.6). This chart shows how debt repayment differs from effort goals, which is a way to clarify whether the goals are achieved or not. Additionally, visualize the team’s overall goal trend.

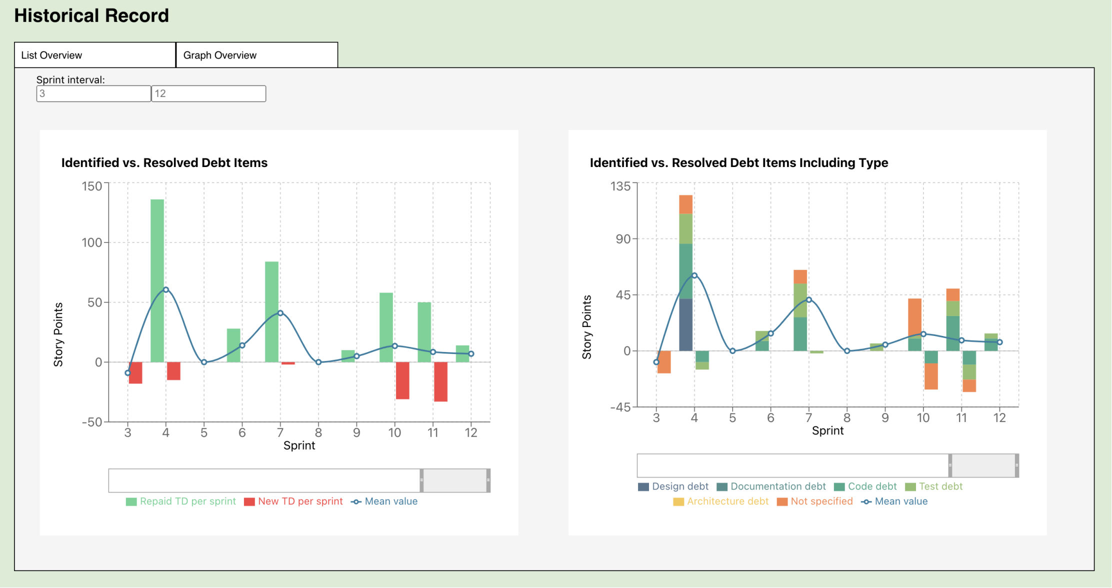


Sprint interval: 5 14

Figure 4.6: Screenshot of artifact: Technical Debt Progress

**Chart 2: Identified- vs. repaid items**

We liked the participant’s idea of using a stacked bar chart with negative and positive values to highlight positive and negative effects. But, instead of using it in Chart 1: TD progress, we figured that this could be used to enhance the identified-vs repaid chart (see leftmost chart in Figure 4.7). The chart can help clarify the difference between TD repayment as a positive effect, and TD identification as a negative effect. Additionally, we included details about TD type in a additional chart here (see rightmost chart in Figure 4.7).


Historical Record

Sprint interval: 3 12

Figure 4.7: Screenshot of artifact: Identified- vs. repaid items

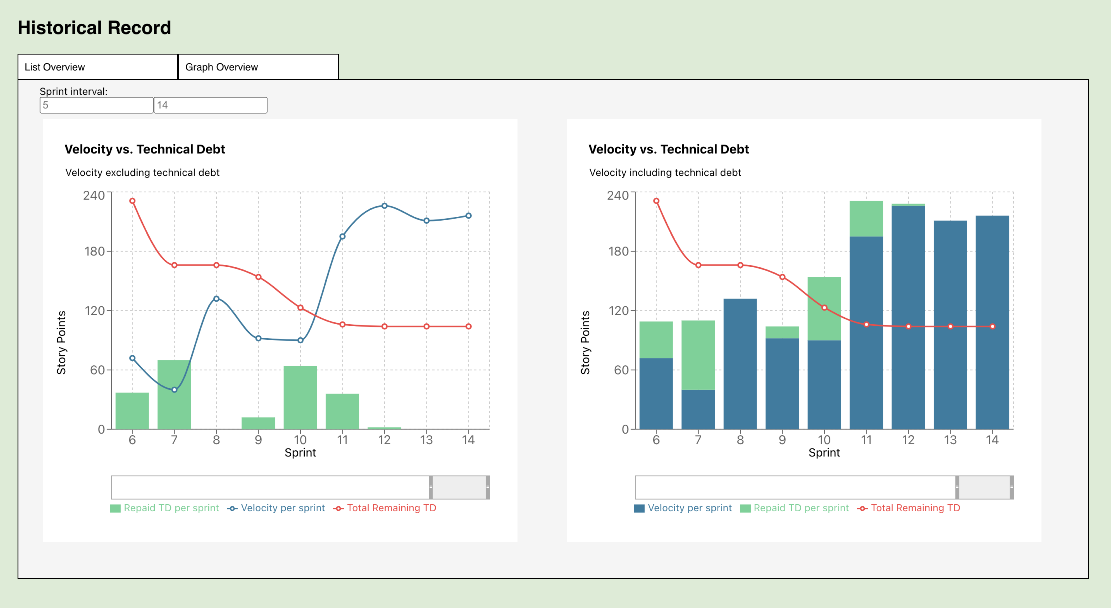
### Chart 3: Velocity vs. Technical Debt

We wanted to include two versions of this chart; one that visualized velocity excluding TD (see leftmost chart in Figure 4.8), and one that visualized velocity including TD (see rightmost chart in Figure 4.8)

If we include the TD in the velocity calculation, the velocity will remain within the average, regardless of TD repayment. Therefore, the visualization must reveal TD repayment inside the velocity, so the user gets insight into the work that has been done.

If we do not include TD in the velocity, it would immediately change the velocity average. Moreover, the decrease in velocity would imply that the team has distributed its resources elsewhere. Therefore, it is crucial to show the TD items to visualize why the velocity dropped here.

In a blog article, [19] tackling the question, ”should we estimate and count the TD in velocity?”, they demonstrate the differences by including TD or not in the calculations with charts. These were used as inspirations when we sketched our new design.

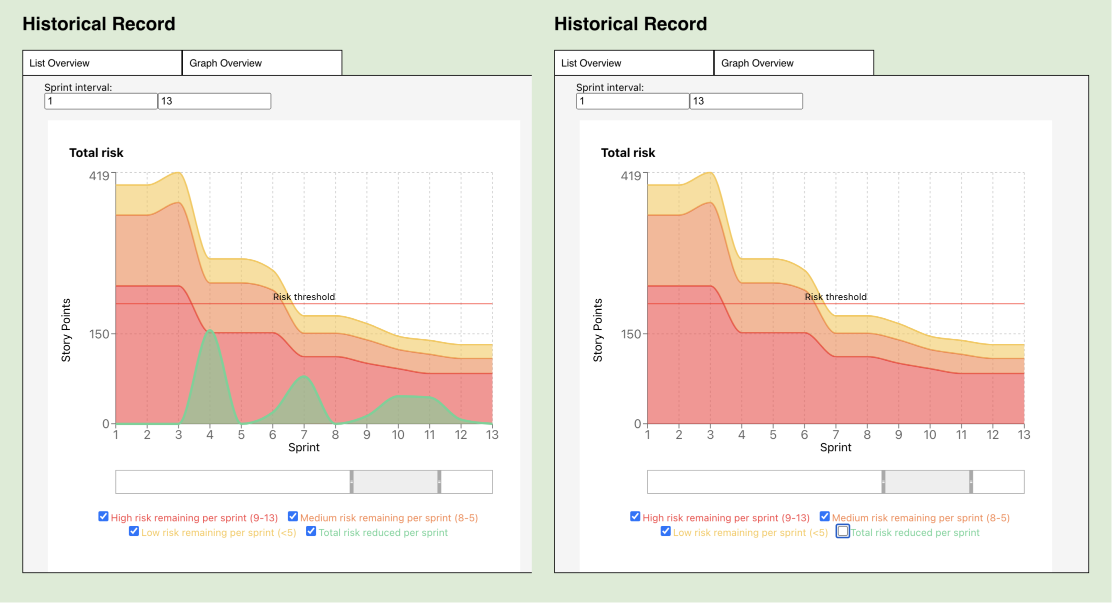


Sprint interval: 5 14

Figure 4.8: Screenshot of artifact: Velocity vs Technical Debt

### Chart 4: Project Risk

When it comes to defining risk values, it is natural to think that different stakeholders and organizations will have different perceptions of these values. Therefore, we wanted to solve the problem of summarizing risk in a way that could work for everyone. Our approach to this problem allows the user to filter out risk levels (see Figure 4.9). Then one can, for example, choose to only look at higher values by filtering out the low ones, which would give the user more control over the risk numbers. We also made it possible to filter out the total risk reduced per sprint, as illustrated in the same figure.



Sprint interval: 1 13

☑ High risk remaining per sprint (9-13) ☑ Medium risk remaining per sprint (8-5)
☑ Low risk remaining per sprint (<5) ☑ Total risk reduced per sprint

Sprint interval: 1 13

☑ High risk remaining per sprint (9-13) ☑ Medium risk remaining per sprint (8-5)
☑ Low risk remaining per sprint (<5) ☐ Total risk reduced per sprint

Figure 4.9: Screenshot of artifact: Project Risk

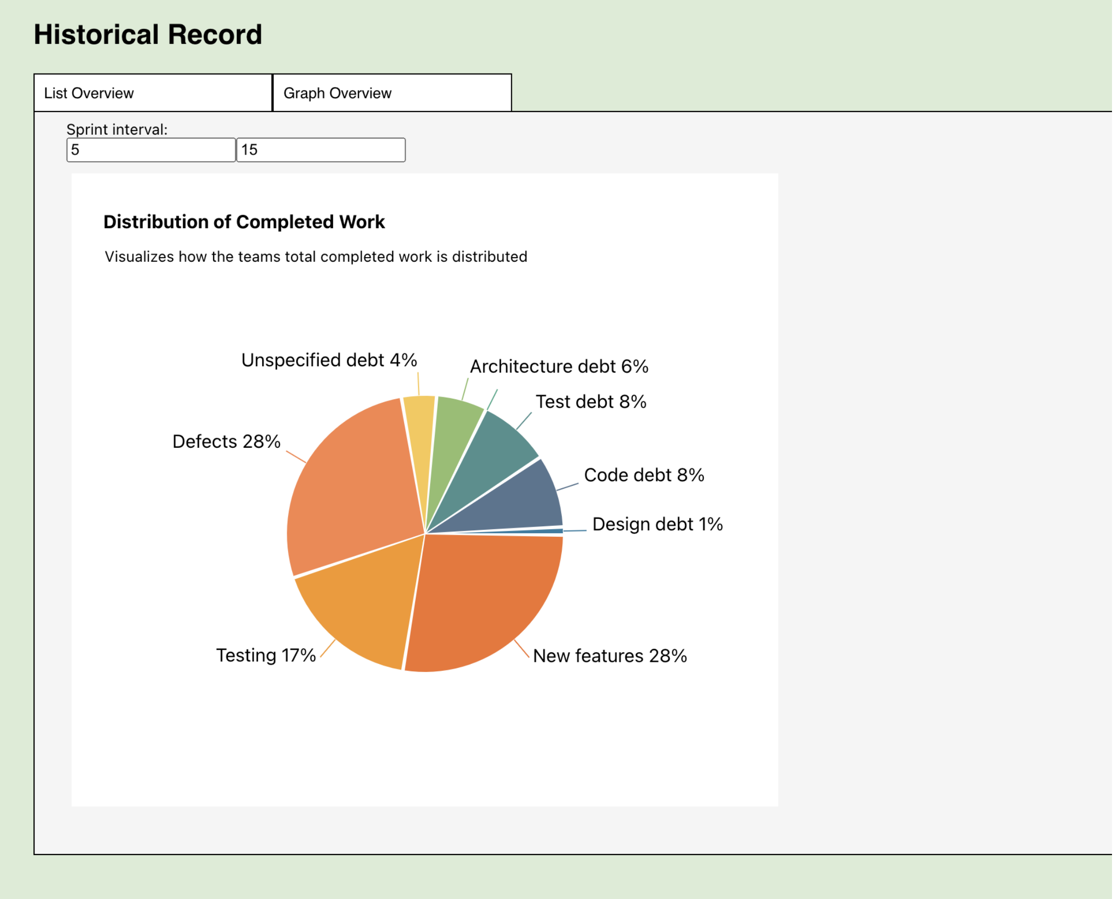
### Chart 5: Distribution of Completed Work

As described earlier, we added a new user story to the backlog:

> ”As a team member/business leader, I want to see a visual overview of the average effort on repaid TD items compared to the team’s total capacity, for a predefined interval, so that I get more insight on how the resources/effort were distributed” .

In addition to showing how effort for TD repayment is distributed among the different categories over time, we are here including all activities that make up the total effort for a sprint (see Figure 4.10). Typical activities include features, testing, TD, and defects. Using a pie chart gives us valuable insight into how much percentage of the total effort is used on TD.

Another benefit of using a pie chart is knowing the percentage of average effort for each debt type for a predefined interval. For example, if the average story point related to code debt is 12, we can plan that we will use the same effort in the next sprint.



Historical Record

Sprint interval: 5 15

Distribution of Completed Work

Visualizes how the teams total completed work is distributed

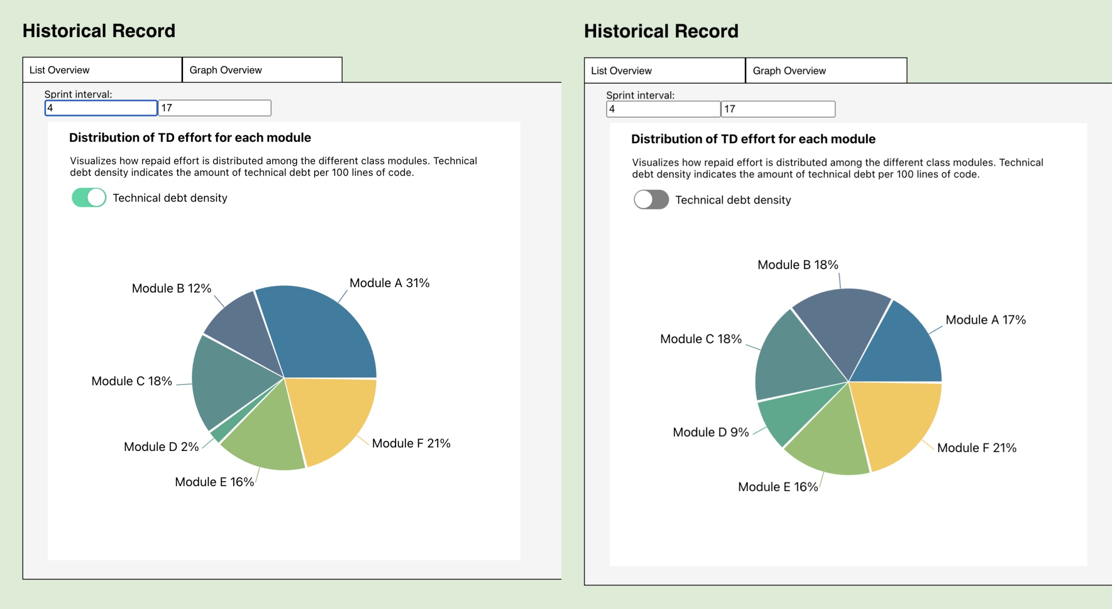
Figure 4.10: Screenshot of artifact: Distribution of Completed Work

### Chart 6: Distribution of TD effort for each module

This chart (see Figure 4.11) was also a result of the new user story added to the backlog:

> ”As a business leader/team member, I want to see a visual overview of the average distribution of TD effort for each module so that I get more insight on how the resources/effort were distributed.”

When we designed this feature, we wanted to find a way to make the pie chart tell us more about how large the TD is relative to the size of the module. An aspect that could give more detail about this is the density of TD. TD density is TD divided by lines of code [30]. We included a simple definition of TD density since we do not expect it to be a concept stakeholders are familiar with.



Historical Record

Sprint interval: 4 17

Distribution of TD effort for each module

Visualizes how repaid effort is distributed among the different class modules. Technical debt density indicates the amount of technical debt per 100 lines of code.

Technical debt density: on

Historical Record

Sprint interval: 4 17

Distribution of TD effort for each module

Visualizes how repaid effort is distributed among the different class modules. Technical debt density indicates the amount of technical debt per 100 lines of code.

Technical debt density: off

Figure 4.11: Screenshot of artifact: Distribution of TD Effort For Each Module

### 4.2.3 Evaluation

In this section, we will describe the results from the last evaluation. First, we go through each chart and present the results according to the interview goals: usefulness, who, when, understandability, and improvements. In addition to this, we also added a paragraph about the artifact in general. Lastly we summarize all the findings (see Table 4.2), which follow the same structure as the detailed description.

To separate each participant, we reference them by using different letters:

- Participant A is the CEO and CTO of ACDtek.
- Participant B is the Senior Software Developer.
- Participant C represents the two Chief Software Quality Engineers.

**Chart 1: Technical Debt Progress**

**Usefulness** All of the participants expressed that they perceived these charts as useful. For example, participant C described this by saying: *"It looks good, and it could be relevant to use visualization like this because one of the things we focus on when we talk to the team about TD management is to make sure you pay enough attention to it over time."*.

Participant B added, *"I would use it for sure since it is clear to see the number of story points."*

Participant A said that *"it is nice that you can draw relations between all these three things [repaid TD, total TD, Effort goal]"*, and further stated that they found it useful and informative. They also pointed out that it would be valuable to see it compared to the usual backlog (with features) and how much they are getting rid of there.

With Participant A, we also discussed the implications by including data from one team or multiple teams. They expressed that it would tell us different things since one team's data would not say much about TD in the overall system, just the user stories they are working on.

**Who would use it** Many roles were mentioned when we discussed which stakeholders would use these charts. Participant C also expressed that most people would have needed this in a way.

All participants mentioned the management and specified roles like product owner and product manager. Participant A elaborated by saying, *"product owner really benefits from this as well because they will want to really transparently see what* *the actual benefit is of repaying TD."* Participant C said, *"They [PO] may want to follow up that they spend the time they need on TD repayment."*

Participants C also mentioned roles such as developers and architects because they could use the charts like this as hard facts to get enough time to prioritize TD repayment.

Participant C further elaborated that it can also be used by people higher up in the organization(e.g., business stakeholders) who want to dive in and see if the teams continuously work with TD. Participant B also said something along these lines when mentioning the management: *"you could use it to see the progress. It is quite useful to see the TD decreasing or increasing. It is important because they don't see it, so they don't want to think about it."*

Participant A mentioned that if the data represents multiple teams, it would be more suitable for someone higher up in the management. They also mentioned roles such as CTO and VIP engineering.

**When would it be used** All participants thought that these charts could be used in planning meetings. For example, participant B said they could use it to plan what they should pay or define the goal for the next sprint. Participant C said that if the chart is included in planning sessions and backlog grooming (prioritization of the backlog), it can signal to the team when they are starting to incur much debt. The signal will then make them realize that they must use more time on TD repayment in the next sprints.

Participant C also thought it could be included in the daily work log, for example, in a dashboard grouped with other important metrics. The users would then establish a continuous relationship with the current status of TD progress. Participant A commented something along these lines that a chart like this should be included in the core activity. How often depends on the company, but it is important to make it a consistent process. For example, if the team uses scrum, it should be integrated into sprint planning. Participant B also mentioned that it is project dependent but could be used every sprint.

Participants A and B also explained that it could be used in the retrospective to see if we have had progress or not.

**Understandability** All participants managed to decode the chart to the left, "technical debt project progress," and understood what it was meant to visualize. One of the participants asked questions about the chart to the right, visualizing the "deviation from effort goal." We needed to explain the meaning behind the values further: The effort goal was not as easy to see in this chart. The participant commented, *"When I see 200%, I don't know the percentage of what exactly. It depends on my effort."* The idea behind it was understandable, but the participant found it hard to read because of the percentages, and the vertical label only says "story points."

**Improvements** Participant B proposed an improvement to the chart to change the vertical axis of the goal chart to include percentages or change the percentages to story points.

Participant C wished that they could get an aggregation of the entire period you look at so that you can assess in total whether you are above or below the goal. The aggregation would also allow the users to assert long-period goals, not just per sprint/week. *"... it is nice to be able to measure it of course from sprint to sprint, but also over longer time horizons."*

### Chart 2: Identified- vs. repaid items

**Usefulness** The overall impression was that the charts were perceived as useful. Participant C was already making visualizations that looked similar to this chart. The only difference is that they use risk instead of story points. However, they serve the same purpose: to get an overview of whether the relation between repayment and identifying new TD goes up or down.

**Who would use it** There was an agreement among the participants that different stakeholders would use the two charts. The chart without debt type specification was more suited for the management. Participants A and C explained this by saying that the management only wants to see total and aggregated data, which tells them something about the performance in simple terms.

However, the chart with debt type specification is more suited for people closer to the development—for example, architects, chief architects, developers, and product owners. Participant A argued this as they are repaying and identifying TD and therefore need to know what happens in detail. The chart tells them the incurred debt and which areas they need to improve.

Participant C also argued for the same by saying that *"This could be really quite interesting for the team I think - to maybe be able to pin-point a bit if you introduce a lot of code debt or architect debt, test debt, etc ... Then you could perhaps see that only test debt increases and that you perhaps could take a closer look at the processes around the tests one does in the team."* The participants did not them-self operate with the specification of debt type but saw the benefits it might provide overtime to make more strategic decisions.

**When would it be used** The retrospective phase was the most frequent phase mentioned where these charts could be useful. Participant A mentioned retrospective and sprint review because these are places where you discuss progress. Participant B also mentioned retrospective but believed it would not be used every sprint because you don't see that many TD items are identified/repaid; every month would be a more suitable interval. Finally, participant C explained that it could explain why some processes went slower for a while, or if the team has had a problem with the production, then maybe it can be correlated with the identified TD.

**Understandability** All of the participants found the first chart as easy to understand. There were only two comments related to the understanding of the chart. Participant B emphasized that the one including types would only make sense if you first look at the one without types. The latter one tells us what the values above and below the zero line are meant to represent, but this is not apparent in the other chart.

Participant C was questioning the use of story points at the y-axis. This comment is also relevant for the other charts, but it was mentioned here. They found it misleading to use it in this context since it usually is (or in their case) used for all other possible tasks/issues in a team.

**Improvements** To fix the labeling issue of the chart, including types, participant B proposed to add a label on the y-axis that tells us what the values above the zero line and below the zero line are meant to represent.

One comment related to the charts that both participants A and B mentioned was related to the mean value. They believed it was not that informative/valuable to use the mean in this context. Participant A proposed that maybe using a net value would be more useful because you could see graphically if you are adding or subtracting to a project.

Participant A also commented that they found it a bit confusing that the bars for the same sprints did not match each other (vertically) but were split into two bars beside each other. They expressed that it was a minor issue, but the use of only one bar would emphasize that they belong to the same sprint.

#### Chart 3: Velocity vs. Technical Debt

**Usefulness** All of the participants found these charts useful. Participant C expressed that representation is something they are missing since velocity and TD is an important relationships to be aware of. They found the chart as a valuable tool to possibly validate the hypothesis: an increasing amount of TD leads to lower velocity. Participant B said that the charts would hold valuable information we could discuss with the management.

**Who would use it** When we discussed who would use these charts, all participants mentioned the management. Participant A proposed roles such as CEO, and CTO, which would use them for high-level strategies. Participants B and C proposed the product owner or the manager since they are responsible for the prioritization. This person can then use this chart to discuss with the management and use it to validate for adding couple more TD items. Participant C also mentioned chief architects who can use this chart for the same reason. The participant mentioned that the charts could provide valuable insight for the team but that it was more relevant to the management.

**When would it be used** All participants expressed that it would not make sense to look at these every sprint. Instead, these charts can be included later in the development, for example, every quarter. And then, it would be included as a part of the refactoring strategy discussion. Participant C also mentioned that the data could be compared to previous quarters, and one could then draw conclusions about the amount of TD related to repayment and velocity values.

**Understandability** Since we included two charts showing somewhat the same data, it became natural to compare them in terms of understandability. Participant A preferred the chart where TD was included in the velocity because it was easier to read. However, they also liked the representation of the chart without TD included because you have two lines going against each other. Participant C also preferred that the TD was included because it gives you information about the total effort and the distribution. However, participant B preferred the chart without TD included and thought that this was easier to read and see that the velocity increased (got better). There was, therefore, no agreement on which one was best, but with that, we got confirmation that both were understandable and that it could be valuable to keep both variants.

**Improvements** The titles of the charts were a bit confusing, and participant A proposed an improvement would be to revise them to be "Velocity vs. TD" and "Velocity ratio with repaying TD" or something along these lines. Participants B and C wanted the charts to reach over a longer time interval as default since it would provide more relevant data for discussion; as participant C said, *"One should see it over a relatively long time horizon maybe then ... because there can be quite a few things that affect a velocity from a sprint to a sprint ... One should perhaps be careful to draw conclusions from too few data points"*.

To reduce such bias in the velocity, they further elaborated that one could use average data points instead of real numbers to highlight the trends. Average values could be most relevant for the chart with velocity excluded TD since it uses lines instead of bars.

### Chart 4: Project Risk

**Usefulness** Participant B mentioned that risk in a project would be very useful for those who know and use the metric risk. They also mentioned that including the repaid TD in the chart could be useful because you can see when you stopped repayment, and the risk is increasing.

Participant C expressed the usefulness of this chart by saying *"I think it gives such a nice overview of the risk distribution, on the total TD you have. Also, you can in a way give such an insight into whether the team only pays the simple debt - or maybe just focuses on the high-risk debt"*.

Participant C already used something similar to this chart in their company, but they made a chart for each team and not the total system. They highlighted that risk- thresholds and values are very team-dependent, and how one evaluates and determines risk values can fluctuate from team to team. Additionally, old legacy products with high TD need a higher risk threshold since the repayment needs to take a longer time. Moreover, the new products with lower total TD would need a lower TD threshold.

**Who would use it** Participant B thought this would apply more to the management because they are more invested in risk values. Further, the participant believed that teams should care about risk, but in practice, they do not. Also, their perception of risk would be different from the management.

Participant C mentioned upper management, *"It could be interesting to see maybe the total debt for all teams, or how this risk is distributed per team - maybe one could shuffle a bit on resources within these ten teams, for example. It can be interesting to at least show to stakeholders."*

**When would it be used** Participant B thought that the use of the chart would depend on the direction of the risk trend. For example, if it is an increasing trend towards the threshold, they would care about it, but if it is decreasing and below the threshold, they would not care about repayment and think they have more important things to do. Participant C, who already uses a similar chart, has had a good experience using the message application Slack to notify when the risk is getting close to the threshold.

**Understandability** Participant A found it confusing that the repaid TD is included in the chart since one already can tell what is happening in the risk-related areas. In addition, participants A and C found it confusing that we mixed different value points (risk points and story points). Finally, participant B pointed out that using areas in the chart can be misleading since it seems that the risk is behind the others.

**Improvements** One improvement proposed by participant B would be to change the risk area to something else, maybe bars or lines so that the values would not be misunderstood. Both participants B and C thought that the repaid TD should be switched off by default since one already can see the reduction of TD risk without it, but keep the opportunity to include it in the chart as a checkbox like it is now. Another alternative proposed to improve this chart based on the discussion of usefulness would be to have one chart for the total risk and one chart for each team.

#### Chart 5: Distribution of Completed work

**Usefulness** In terms of usefulness, participant A thought this was really great. Participant C has experienced that similar charts are already used to evaluate the team's total capacity in a retrospective context, emphasizing its usefulness. Further, they argued that such charts could help the stakeholders track where the time is distributed. However, including details about TD type is not something they have seen before, and they thought it was a positive contribution. Finally, participant B explained that the pie chart alone is not that valuable to determine if the distribution is good or bad and would be more valuable with another chart. However, the effort distribution alone is a useful insight.

**Who would use it** When we discussed the usefulness, participant C mentioned that the team would value this level of detail to see if we spent enough time on each debt category or other activities. Time distribution is not necessarily what the team thinks about as they continuously work with different things. However, the project leaders or similar roles would be more interested in seeing the total TD alone, not as relevant with the level of detail.

**Understandability** Concerning the understandability of the pie chart, the participants thought it was a bit hard to separate the different groups of TD and other activities. Participant B expressed this by saying *"It is a bit difficult to* *separate what is TD and what is not, I have to read the legends."*. He further stressed that features and defects have the same color almost, which is a bit confusing.

Even if the participant was not a fan of pie charts, he agreed that it is a good way to distribute the total amount of work, and for example, a bar chart would not do this.

**Improvements** Participant A would like the chart better if we would highlight the time interval for the pie chart by including it under the chart like the other charts. Participant B mentioned that the colors should be improved so that different colors are used to highlight the different activities. Further, he proposed that at least features, defects, and TD should have contrasting colors. Finally, participant C mentioned that a more simplistic chart could be suitable in some cases. Therefore they would like to have a feature that lets them merge everything that had to do with TD as one slice in the pie chart. We could do this by including a checkbox or something similar so that one can split it back to the detailed version.

### Chart 6: Distribution of TD effort for each module

**Usefulness** It was not as clear to all of the participants in what ways these charts could be useful. The reason behind this is as participant B formulated *"If it's something that we have done already... maybe its not so useful because we paid already. We are more worried about the new things. It would be nice to see... Would be important to see when if I failed... but the graphics don't tell me that"*.

However, participant C believed that it could be useful if we saw it together with the number of bugs or incidents for each of the modules/systems:

*"Now we have spent 30% effort and paid off the TD that is in module A, and then I might have naturally expected that, for example, the number of incidents will decrease."*

Then they would see if the amount of effort used on a module was time well spent or if the effort should have been elsewhere.

So both participants B and C did not find the chart useful alone but saw the potential that it could be useful if we could compare it to other metrics or situations to analyze past events.

**Understandability** A critical aspect of this pie chart is that all participants believed that the chart showed the unresolved TD in the system and not the repaid. Participant B believed that the story points made it confusing since the term is usually associated with features. Participant C thought that it was because of the unclear title, as it did not say anything about repaid TD.

As anticipated, no one of the participants was familiar with the term TD density. Despite the definition included in the chart, the participant did not fully understand the concept, which made us need to explain it more.

**Improvements** Participant A thought filtering out the density and adding it to a new chart would be an improvement. They believed that it was an important aspect, but as it was now made it confusing. Another suggestion would be to describe the concept of TD density in more detail.

To avoid misunderstanding which type of TD data the chart is visualizing, we could revise the title to include ″repaid,″ such as ″Distribution of repaid TD effort for each module.″

Another suggestion from participant B for improvements to the chart would be to include the story points inside each slice in the pie chart without needing to hover it. So that both the percentages and story points are visible. The last suggestion was to include an aspect that highlight's the different prioritizes/risks for each debt, for example, by adding checkboxes for high, medium, and low risk. This division would perhaps provide new valuable insight.

**General feedback** Participant C discussed how this could work in their context. They mentioned that their teams have moved away from the framework Scrum and were using other frameworks such as Kanban. Therefore values such as sprint would not apply to them. Therefore they believed that an improvement would be to use more general terms for the artifact. For example, instead of using ″sprints″ as a value for the y axis, one could use time as default and let the teams tailor the values, so it suits their context.

Additionally, participant C did neither use story points in a TD context. Therefore another improvement they proposed was to include the possibility of changing the y-axis to include other values than story points. Finally, as they did operate with risk severity, it would be nice to tailor this chart so it would be useful for them too.

|  | Usefulness |  |  | Understandable | Improvements |
|---|---|---|---|---|---|
|  | General | Who | When |  |  |
| Technical Debt Project Progress | • Informative and possible<br>• Show that you are paying enough attention to TD management over time<br>• Nice to draw relations between repaid TD, total TD, and Effort goal<br>• It's clear to see the amount of story points | • Top management, e.g business stakeholders<br>• Product owner/product manager<br>• The team, e.g developers and architects | • Planning-meetings (+ backlog grooming)<br>• Daily worklog (e.g. dashboard)<br>• Retrospective-meetings | • The Project Progress graph was understandable<br>• "Deviation from effort goal" was a bit hard to read because of label and value mismatch | • Goal-graph: Include % in the vertical axis or change the percentages to story points<br>• Show a aggregated goal value for the entire time interval so one could assess long-term goals |
| Identified vs repaid debt | • Useful to see if the total TD is increasing or decreasing | • (including types) People close to the development such as (chief) architects, developers, and product owners<br>• (excluding types) The management | • Retrospective-meetings and sprint reviews | • Both were perceived as understandable<br>• The graph including types doesn't tell what the negative and positive values represent | • Add a label on the y-axis that explain what the negative and positive values represent<br>• Replace the mean value with a Netto value that tells the team whether they are adding or subtracting to the total<br>• Merge the negative- and positive values into one bar |
| Velocity vs. repaid debt items | • The relationship between velocity and TD and how they affect each other is an important to be aware of<br>• Valuable information we could discuss with the management | • The management: CEO, CTO, Product owner and manager<br>• Possibly the team as well | • Refactoring strategy discussions<br>• Later in the development with long intervals (e.g every quarter) | • Both graphs were easy to read<br>• Velocity incl. TD: easier to read the total effort, and the effort distribution<br>• Velocity excl. TD: easier to see that velocity gets better with the two lines going against each other | • Improve the titles<br>• Longer time interval as the default (e.g quarter)<br>• Use average data points instead of real numbers to highlight the trends |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Project Risk | • Useful for those who have knowledge about risk<br>• Gives an good overview of the risk distribution and the total TD<br>• Insight into whether the team pays only the low-risk or the high-risk debt | • The management and stakeholders<br>• Possible the team | • When the risk trend is increasing towards or crosses the risk threshold<br>• Notify when the risk is getting close to the threshold | • Found it confusing that the repaid TD is included since the risk areas alone tells you what is happening<br>• Confusing with a graph with both risk points and story points<br>• It seems that the risk areas are behind each other instead of on top | • Change the risk areas to emphasize that the values are on top of each other<br>• Repaid TD should be switched off by default since you can see the reduction of TD risk without it<br>• Include a risk graph for each teams |
| Distribution of Completed Work | • Useful to help stakeholders get an overview of how the total capacity have been distributed over time<br>• Does not tell whether the distribution is good or bad | • Team (detailed TD)<br>• The project leaders or similar roles (total TD) | - | • Hard to separate the different slices of TD types with the other activity-slices<br>• Confusing with the use of similar colors | • Include the time interval as a part of the graph<br>• Features, defects and TD should have contrasting colors<br>• Make it possible to merge the TD slices into one slice |
| Distribution of TD effort for each module | • Not so useful on its own<br>• Could be useful in comparison with metrics/events to analyze past events | - | - | • All of the participants thought the graph visualized current TD and not repaid TD<br>• The participant did not fully understand the TD density concept | • Remove the TD density, make a separate chart for it or describe it in more detail<br>• Improve the title<br>• Include the story points inside each slice in the pie chart<br>• Show the different prioritizes/risk for each debt to give more insight |

Table 4.2: Summary of findings

# Chapter 5

# Artifact

## 5.1 Technology used

This section will describe the technology used in developing the historical record. Considering the research goal and choice of technology, we focused on creating a technical prototype. The final product was developed to be used in a live demonstration with relevant stakeholders and not as a robust technical solution ready for production. The artifact is a perfect example of a project where taking on TD is used as an advantage.

Hence, the most important factor for us when choosing which technologies to use to build our artifact was that they were easy to learn and apply to our project. Therefore, we used technology that we were already familiar with or widespread with well-documented technology.

**React** To implement our artifact, we used the front-end library JavaScript with React. React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets one compose complex UIs from small and isolated pieces of code called ”components” [40].

It is also known as one of the most popular libraries for creating web applications. We have also experience with this library from previous projects.

React has multiple component libraries filled with valuable design elements, making it easy to find suitable libraries for our project.

**AdGrid** Our solution needed a grid component that could store the TD items. To do this, we used the data grid library AdGrid. AdGrid is the leading Enterprise JavaScript data grid and offers complete React support. In addition, it offers many features like sorting, filtering, and grouping, which allow for quick value with minimal coding [17].

**ReCharts** Another crucial component of our solution was the different graphical representations. To help with this, we used the library ReCharts. ReCharts is a composable charting library built on React components [18]. It is one of the most popular charting libraries for React, with many users and a large community.

**Trello Api** As described in the Tool survey (Section 2.5), we decided to use the project management tool Trello as a part of our artifact. Therefore, we made a Trello board that simulated a TD backlog (see Figure 5.1). Then we made a script that added a large number of TD items into it. To use the data from the example TD backlog made in Trello, we needed to gather data from the Trello API. The TD items placed in the ”Done” list are the ones we need to collect for the historical record. Trello provides a simple RESTful web API where each type of resource (e.g., a card, a board, or a member) has a URI that you can interact with [50].

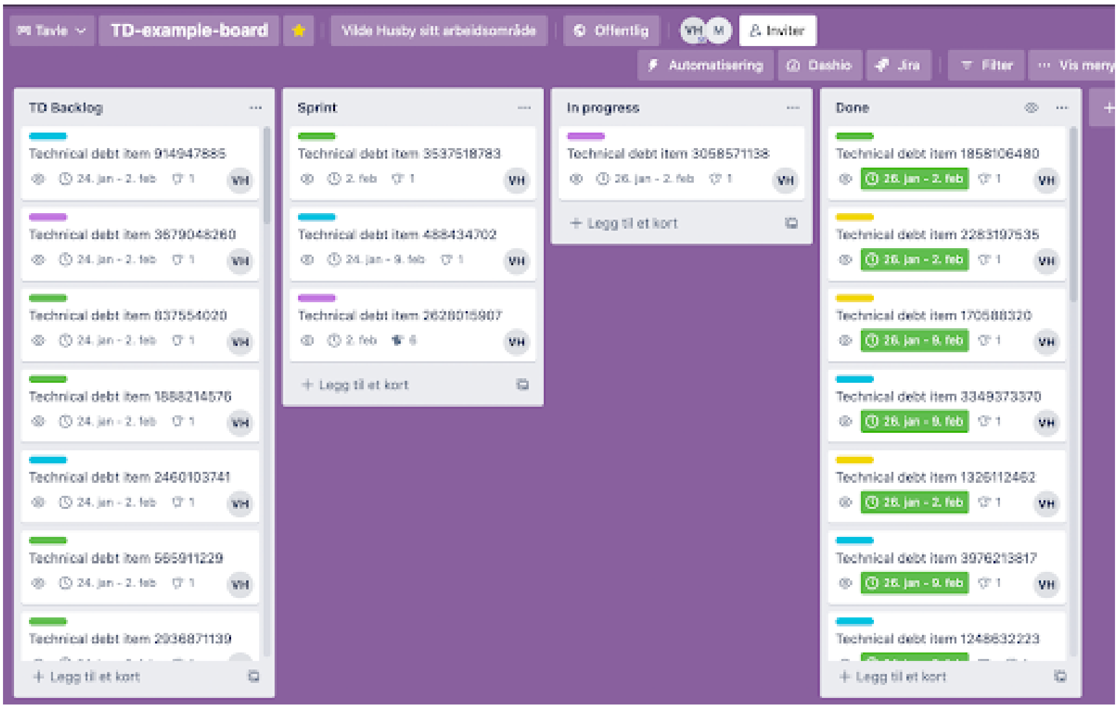

Tavle
TD-example-board
Vilde Husby sitt arbeidsområde
Offentlig
VH
M
Inviter
Automatisering
Dashio
Jira
Filter
Vis meny
TD Backlog
Technical debt item 914947885
24. jan - 2. feb
1
VH
Technical debt item 3879048260
24. jan - 2. feb
1
VH
Technical debt item 837554020
24. jan - 2. feb
1
VH
Technical debt item 1888214576
24. jan - 2. feb
1
VH
Technical debt item 2460103741
24. jan - 2. feb
1
VH
Technical debt item 565911229
24. jan - 2. feb
1
VH
Technical debt item 2036871139
Legg til et kort
Sprint
Technical debt item 3537518783
2. feb
1
VH
Technical debt item 488434702
24. jan - 9. feb
1
VH
Technical debt item 2628015907
2. feb
6
VH
Legg til et kort
In progress
Technical debt item 3058571138
26. jan - 2. feb
1
VH
Legg til et kort
Done
Technical debt item 1858106480
26. jan - 2. feb
1
VH
Technical debt item 2283197535
26. jan - 2. feb
1
VH
Technical debt item 170588320
26. jan - 9. feb
1
VH
Technical debt item 3349373370
26. jan - 9. feb
1
VH
Technical debt item 1326112462
26. jan - 2. feb
1
VH
Technical debt item 3976213817
26. jan - 9. feb
1
VH
Technical debt item 1248632223
Legg til et kort

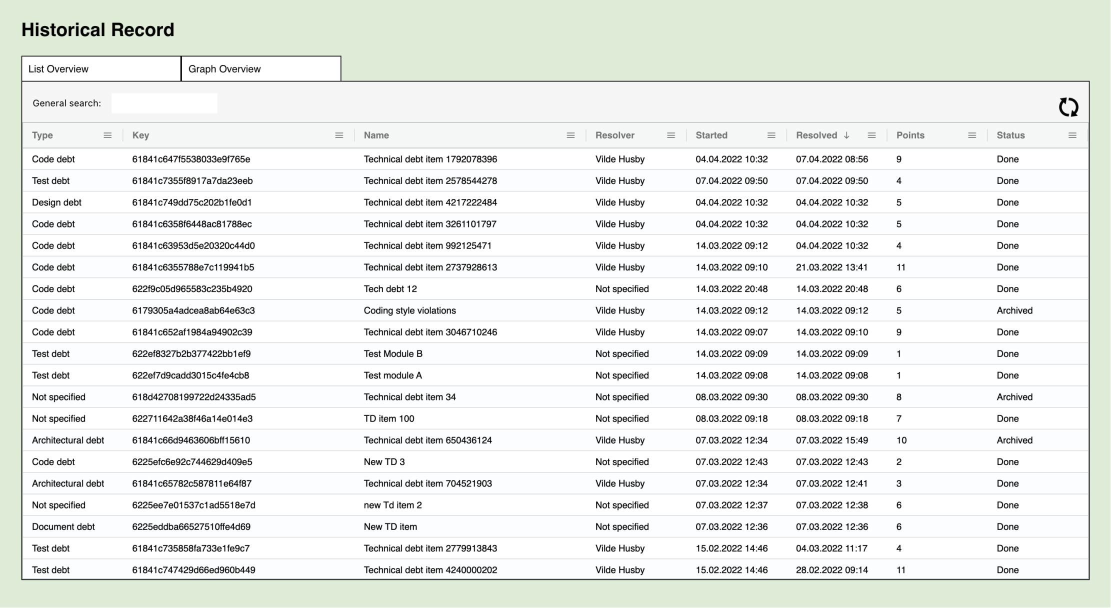
Figure 5.1: TD backlog in Trello

## 5.2 Main functionalities

This section will describe the main functionalities of the implemented artifact. Due to the scope of this research, we have not specifically evaluated all parts of the historical record. Therefore, we will here describe the complete artifact, including all of the user stories in the final backlog (see Appendix D).

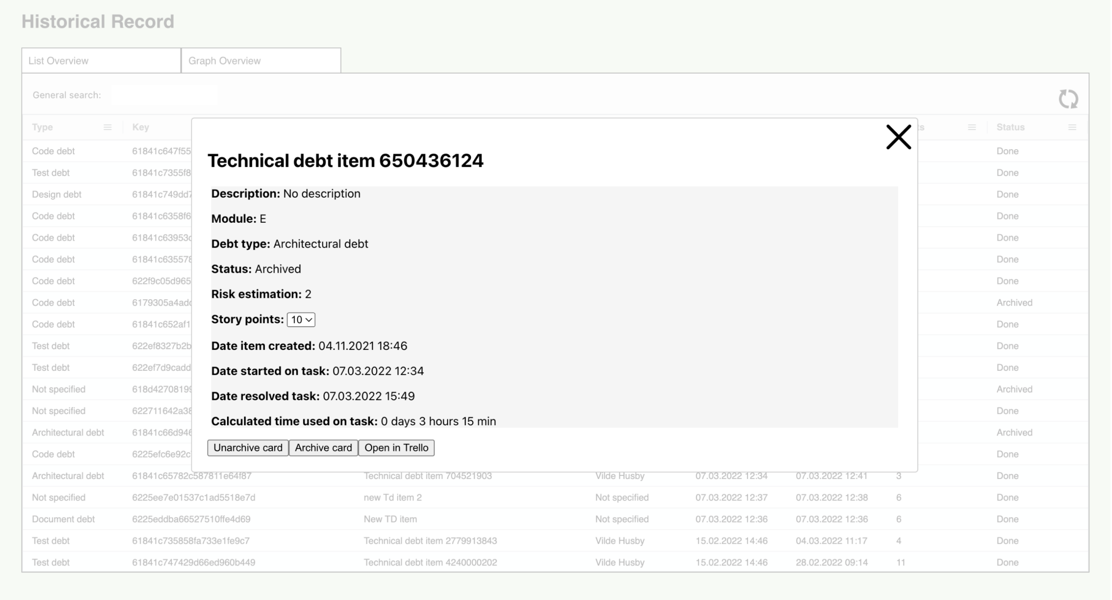
The first overview the user sees when opening the chart is the ”List Overview.” (see Figure 5.1). Here we have included a table that contains TD items gathered from the TD backlog in Trello. The table offer functionality that lets one search in and sort on each column. Additionally, the user can also do a general search for all columns. By default, the items are sorted after resolved date.

If one clicks on a row in the table, they are prompted with a ”modal.” (see Figure 5.2). The modal includes more details about the chosen TD item: description, module, debt type, status, risk estimation, story points, creation-, started- and resolved date, and calculated time used on the task. The story points value is possible to change. Further, the modal view offers three buttons with the following functionality: archiving-unarchiving- and opening a card in Trello.



| Type | Key | Name | Resolver | Started | Resolved ↓ | Points | Status |
|---|---|---|---|---|---|---|---|
| Code debt | 61841c647f5538033e9f765e | Technical debt item 1792078396 | Vilde Husby | 04.04.2022 10:32 | 07.04.2022 08:56 | 9 | Done |
| Test debt | 61841c7355f8917a7da23eeb | Technical debt item 2578544278 | Vilde Husby | 07.04.2022 09:50 | 07.04.2022 09:50 | 4 | Done |
| Design debt | 61841c749dd75c202b1fe0d1 | Technical debt item 4217222484 | Vilde Husby | 04.04.2022 10:32 | 04.04.2022 10:32 | 5 | Done |
| Code debt | 61841c6358f6448ac81788ec | Technical debt item 3261101797 | Vilde Husby | 04.04.2022 10:32 | 04.04.2022 10:32 | 5 | Done |
| Code debt | 61841c63953d5e20320c44d0 | Technical debt item 992125471 | Vilde Husby | 14.03.2022 09:12 | 04.04.2022 10:32 | 4 | Done |
| Code debt | 61841c6355788e7c119941b5 | Technical debt item 2737928613 | Vilde Husby | 14.03.2022 09:10 | 21.03.2022 13:41 | 11 | Done |
| Code debt | 622f9c05d965583c235b4920 | Tech debt 12 | Not specified | 14.03.2022 20:48 | 14.03.2022 20:48 | 6 | Done |
| Code debt | 6179305a4adcea8ab64e63c3 | Coding style violations | Vilde Husby | 14.03.2022 09:12 | 14.03.2022 09:12 | 5 | Archived |
| Code debt | 61841c652af1984a94902c39 | Technical debt item 3046710246 | Vilde Husby | 14.03.2022 09:07 | 14.03.2022 09:10 | 9 | Done |
| Test debt | 622ef8327b2b377422bb1ef9 | Test Module B | Not specified | 14.03.2022 09:09 | 14.03.2022 09:09 | 1 | Done |
| Test debt | 622ef7d9cadd3015c4fe4cb8 | Test module A | Not specified | 14.03.2022 09:08 | 14.03.2022 09:08 | 1 | Done |
| Not specified | 618d42708199722d24335ad5 | Technical debt item 34 | Not specified | 08.03.2022 09:30 | 08.03.2022 09:30 | 8 | Archived |
| Not specified | 622711642a38f46a14e014e3 | TD item 100 | Not specified | 08.03.2022 09:18 | 08.03.2022 09:18 | 7 | Done |
| Architectural debt | 61841c66d9463606bff15610 | Technical debt item 650436124 | Vilde Husby | 07.03.2022 12:34 | 07.03.2022 15:49 | 10 | Archived |
| Code debt | 6225efc6e92c744629d409e5 | New TD 3 | Not specified | 07.03.2022 12:43 | 07.03.2022 12:43 | 2 | Done |
| Architectural debt | 61841c65782c587811e64f87 | Technical debt item 704521903 | Vilde Husby | 07.03.2022 12:34 | 07.03.2022 12:41 | 3 | Done |
| Not specified | 6225ee7e01537c1ad5518e7d | new Td item 2 | Not specified | 07.03.2022 12:37 | 07.03.2022 12:38 | 6 | Done |
| Document debt | 6225eddba66527510ffe4d69 | New TD item | Not specified | 07.03.2022 12:36 | 07.03.2022 12:36 | 6 | Done |
| Test debt | 61841c735858fa733e1fe9c7 | Technical debt item 2779913843 | Vilde Husby | 15.02.2022 14:46 | 04.03.2022 11:17 | 4 | Done |
| Test debt | 61841c747429d66ed960b449 | Technical debt item 4240000202 | Vilde Husby | 15.02.2022 14:46 | 28.02.2022 09:14 | 11 | Done |

Figure 5.2: Screenshot of artifact: Table



**Technical debt item 650436124**

**Description:** No description

**Module:** E

**Debt type:** Architectural debt

**Status:** Archived

**Risk estimation:** 2

**Story points:** 10

**Date item created:** 04.11.2021 18:46

**Date started on task:** 07.03.2022 12:34

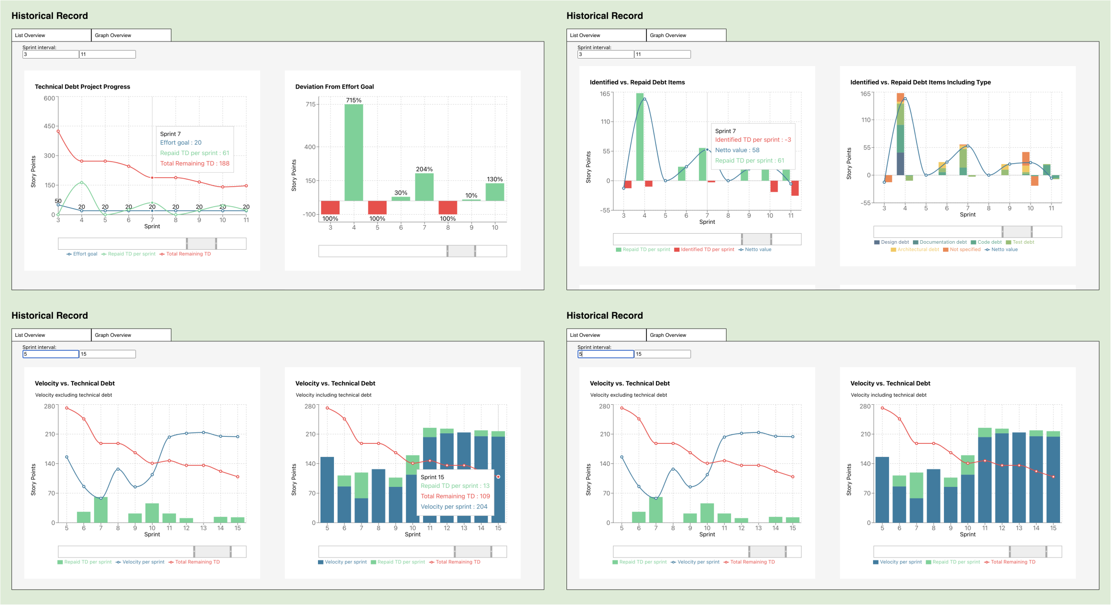
**Date resolved task:** 07.03.2022 15:49

**Calculated time used on task:** 0 days 3 hours 15 min

| Type | Key | | | | | | Status |
|---|---|---|---|---|---|---|---|
| Code debt | 61841c647f55 | | | | | | Done |
| Test debt | 61841c7355f8 | | | | | | Done |
| Design debt | 61841c749dd7 | | | | | | Done |
| Code debt | 61841c6358f6 | | | | | | Done |
| Code debt | 61841c63953c | | | | | | Done |
| Code debt | 61841c635578 | | | | | | Done |
| Code debt | 622f9c05d965 | | | | | | Done |
| Code debt | 6179305a4adc | | | | | | Archived |
| Code debt | 61841c652af1 | | | | | | Done |
| Test debt | 622ef8327b2b | | | | | | Done |
| Test debt | 622ef7d9cadd | | | | | | Done |
| Not specified | 618d42708199 | | | | | | Archived |
| Not specified | 622711642a38 | | | | | | Done |
| Architectural debt | 61841c66d946 | | | | | | Archived |
| Code debt | 6225efc6e92c | | | | | | Done |
| Architectural debt | 61841c65782c587811e64f87 | Technical debt item 704521903 | Vilde Husby | 07.03.2022 12:34 | 07.03.2022 12:41 | 3 | Done |
| Not specified | 6225ee7e01537c1ad5518e7d | new Td item 2 | Not specified | 07.03.2022 12:37 | 07.03.2022 12:38 | 6 | Done |
| Document debt | 6225eddba66527510ffe4d69 | New TD item | Not specified | 07.03.2022 12:36 | 07.03.2022 12:36 | 6 | Done |
| Test debt | 61841c735858fa733e1fe9c7 | Technical debt item 2779913843 | Vilde Husby | 15.02.2022 14:46 | 04.03.2022 11:17 | 4 | Done |
| Test debt | 61841c747429d66ed960b449 | Technical debt item 4240000202 | Vilde Husby | 15.02.2022 14:46 | 28.02.2022 09:14 | 11 | Done |

Figure 5.3: Screenshot of artifact: Modal

The following view is the "Chart Overview," with the six different graphical representations (see Figure 5.3 and 5.4). The data used in these charts are the same as for the table. In the upper left corner, we have included a sprint interval that lets the user change the start- and end sprint numbers for all charts. For each bar chart, we have also included a "brush" that lets the user change the interval for that chart individually. All charts display legends for each sprint, so the user is prompted with all of the specific values.


Historical Record

List Overview | Graph Overview

Sprint interval: 3 – 11

Technical Debt Project Progress

Sprint 7
Effort goal : 20
Repaid TD per sprint : 61
Total Remaining TD : 188

Deviation From Effort Goal

Identified vs. Repaid Debt Items

Sprint 7
Identified TD per sprint : -3
Netto value : 58
Repaid TD per sprint : 61

Identified vs. Repaid Debt Items Including Type

Sprint interval: 5 – 15

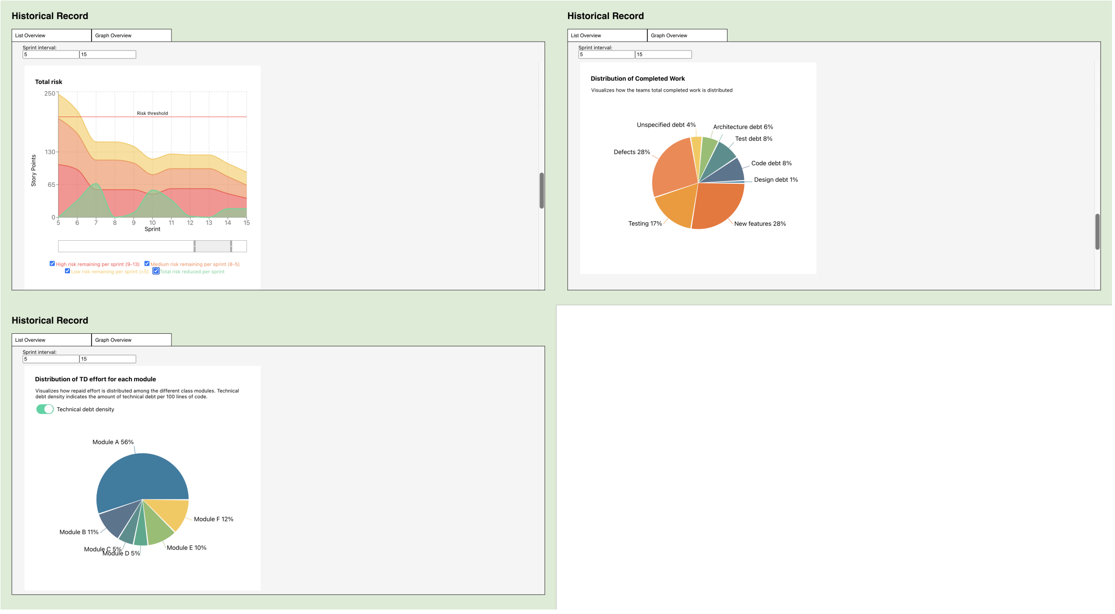
Velocity vs. Technical Debt

Velocity excluding technical debt

Velocity including technical debt

Sprint 15
Repaid TD per sprint : 13
Total Remaining TD : 109
Velocity per sprint : 204

Figure 5.4: Screenshot of artifact: Graph view


Historical Record

List Overview | Graph Overview

Sprint interval: 5 – 15

Total risk

☑ High risk remaining per sprint (9-12) ☑ Medium risk remaining per sprint (6-9)
☑ Low risk remaining per sprint (<5) ☑ Total risk reduced per sprint

Distribution of Completed Work

Visualizes how the teams total completed work is distributed

Distribution of TD effort for each module

Visualizes how repaid effort is distributed among the different class modules. Technical debt density indicates the amount of technical debt per 100 lines of code.

Technical debt density

Figure 5.5: Screenshot of artifact: Graph view 2

# Chapter 6

# Discussion

In this chapter we discuss and present the theoretical contributions of our research. The discussion is based on the research question proposed in the introduction. Next, we will discuss the related work in relation with our contribution. Further, we will discuss the lessons learned from the research, both how it was conducted, and the resulting artifact. Next, we discuss the quality of the research, in terms of validity and the DSR guidelines. Lastly, we will present suggestions for future work.

## 6.1 Theoretical contributions

The question we aimed to solve in the conducted research was the following:

> *RQ: How can visualization of historical data improve the management of technical debt?*

As an attempt to answer the research question, we designed and developed an artifact that consists of six different visualizations that, in each way, try to serve as a tool to improve TD management. First, we will go through the intended purpose/goal of the visualizations and compare them with our results. The purpose/goal of the visualizations is a part of the related user stories. Then, after concluding each chart's results, we will go through the relevant TD management activities and discuss whether or not the results can be used to support these activities.

### 6.1.1 Did we achieve the goals?

**Technical Debt Project Progress** The purpose of the chart was to give stakeholders a visual overview of repaid TD items to make them more aware of the payment progress. Additionally, we wanted to improve the chart by including their goal for the number of solved items to motivate the team to achieve this goal.

The evaluations proved that these purposes were met to some extent. All of the participants mentioned directly or indirectly that it would increase the attention of the TD over time. From a bottom-up perspective, the team can use the chart to argue the need for more resources/time for TD repayment. From a top-down perspective, one can use the chart to follow the TD project progress, tell if it is going as planned, and transparently see the benefits of repaying TD.

However, the evaluation did not provide results that supported the purpose of motivating the team to archive their goals. Additionally, in general, how the team could directly have used this chart. However, we did not ask about team motivation directly. Instead, we found that the goal was useful for the product owner and the management to assess the project process.

**Identified- vs. repaid items** This visualization aimed to make the stakeholders see if they are keeping up with the TD work that is coming in by visualizing the identified and repaid items. Our findings support this purpose. We found that the non-technical stakeholders could use the less detailed version to gain insight into the team's performance. The technical stakeholders can use the detailed version with TD types to see where they have incurred debt and which areas they need to improve.

**Velocity vs. Technical Debt** The purpose of this visualization was to let the stakeholders monitor how development speed is affected by the amount of TD. Our findings supported this purpose. However, as we also learned from the first chart, it applied more to the management than the teams. The developers are hands-on and will, therefore, in most cases, be aware of project slowness due to TD. However, the manager and the management are not in this position. There can be many reasons why a velocity will drop, and this chart visualizes the impact TD can have on project progress.

**Project Risk** This chart aimed to give the stakeholders an overview of risk progress. We also, in this chart, included a risk threshold as a goal aspect to motivate the team to repay the TD. The findings supported these purposes. However, we experienced that we need to differentiate the team's risk progress and the project's risk progress, as they can be interpreted differently and have different threshold values. Moreover, we are still unsure how efficient the threshold is as an incentive to repay debt, but the results indicated that it increased attention to the chart and the risk.

Additionally, through the DSR iterations, we included details about risk-level distribution in the chart. The distribution made the TD repayment transparent in showing which risk levels have been paid off. Offering more transparency improved the overall overview of risk progress.

**Distribution of Completed work & Distribution of TD effort for each module** The purpose of these visualizations was to provide insights into how resources/efforts were distributed. As recording in the findings, the purpose was met. However, these insights were not necessarily useful. They did not say anything about if the distribution was good or bad. Therefore, the purpose of the chart should also concern this matter.

The distribution of completed work could be useful since the team can have their perception of what is good or bad. For example, from the evaluation, we learned that one participant had a goal of dedicating 20% of their total capacity to TD items. However, the distribution of TD effort for each module chart was not as easy to evaluate in terms of good and bad.

To make these charts (more) useful, we need to present the data with values that tell us something about the health of the project/modules, such as velocity and incident/bugs.

### 6.1.2 Improve TD Monitoring

Li et al. [29] identified TD monitoring as a TD management activity that watches the changes in the cost and benefit of unresolved TD over time. This research introduces a new approach to TD monitoring concerning the changes in the amount of resolved TD over time. The monitoring of changes in resolved TD is compared with other important metrics including unresolved TD, project goals, TD types, velocity, and risk. Our findings described how monitoring these relationship could provide different useful insights.

In the project progress chart, we monitor the progress of TD repayment over time. Here we include the unresolved total TD to help the stakeholders understand how much the repaid TD makes up of the total. We also include the goal to help the stakeholders monitor goal progress over time.

Monitoring the relationship in the identified vs. repaid chart tells stakeholders whether the amount of outstanding TD is increasing or decreasing. Additionally, watching the changes in TD type efforts.

In the velocity vs. TD chart we monitor the TD velocity and total TD relationship over time. Here we include the changes in repayment effort to highlight the effect this has on their relationship. In that way, the stakeholders will gain useful insights into how TD and repayment activities can impact the velocity.

The project risk chart lets the user monitor changes in the total risk relative to a predefined risk threshold. Monitoring this can reveal healthy and unhealthy progress depending on how close the risk is to the threshold. By including TD repayment, it can possibly explain the trend to increased/decreased risk.

In the distribution of completed work chart we monitor how the distribution of repayment of TD have been, in relation to other activities that constitute the total capacity. Lastly, in the distribution of TD effort for each module chart we monitor the distribution of where in the system we have repaid debt. By changing the time-interval, one can compare how the distributions have been at different times.

As long as the TD backlog is continuously managed, the graphical representations will be updated automatically. This makes the monitoring accessible and understandable for all stakeholders, at all times through the tool. Additionally, the scope of the changes the stakeholders want to monitor can easily be changed by changing the interval in the artifact. For these reasons, our artifact serve as a tool to maintain control over the TD evolution, and communication the evolution effectively to stakeholders.

### 6.1.3 Improve TD Communication

Li et al. [29] also identified TD communication as an activity in TD management. The activity involves making TD visible to stakeholders so that it can be discussed and further managed. Our artifact builds upon the approach of communication TD in a backlog. We further improve it by visualizing the resolved TD items in graphical representations including bar charts and pie charts. Our findings have identified how our visualizations are useful, to which stakeholders they are useful, and in which parts of the existing processes the visualization can be used. The findings prove that our artifact serves as a tool to improve TD communication and, further, TD management.

**Non-technical stakeholders** Improvement of TD communication applies the most to the non-technical stakeholders. Klinger et al. [24] argue that the main challenge related to TD communication activity is to fill the information gap between non-technical and technical stakeholders. They are detached from the activities where TD occurs and could benefit from simple visualizations that provide enough insights to make the right decisions for future TD management.

The TD project progress chart can be used to improve non-technical stakeholders' perception of the total TD. Either by actively looking at the charts themselves to identify the trends or used as a tool by the team to explain the trends to the stakeholders. These actions would improve discussions, especially in planning meetings concerning how TD should be managed further. Furthermore, discuss how project progress has been in a retrospective meeting.

Further, the identified vs. resolved chart (excluding types) can make the management better understand how the team's performance has been in general. The velocity vs. repaid debt can make TD's impact on velocity visible and further influence discussion on refactoring strategies. The overview of project risk can help them assess if the team is paying off the correct risk. Additionally, by visualizing one risk for each team, the management can rearrange the resources between teams to avoid crossing the risk threshold. Lastly, the distribution of completed work (without details) can be used to assess if the teams are spending enough time on TD repayment relative to other activities.

**Technical stakeholders** The visualizations can also support the teams by giving them a tool to communicate their problems/needs more efficiently. Also, the identified vs. resolved chart (including types) can improve the teams' internal communication, e.g., in a retrospective meeting, on areas they can improve on and where the effort has been placed. The latter also applies to the distribution of completed work chart; only here do we visualize TD effort relative to non-TD activities. Finally, the project risk chart used with notifications can be an effective tool to involve the team in the risk and perhaps serve as an incentive for repayment.

### 6.1.4 Improve TD Management

By demonstrating how the historical record can improve TD monitoring and TD communication, we also prove that it can improve TD management. The research builds upon the approach of recording TD items in a backlog. One big drawback of this approach is that the first investment can be burdensome. The process of identifying, measuring, and prioritizing the initial TD requires much effort [33]. It can be challenging for the organization to foresee that the investment will yield a sufficient return. Research in the field and the development of tools for this approach can be important in clarifying the profit. We have identified how one can improve TD management by following the backlog approach through our research. Hence make, it more lucrative to invest in a TD backlog tool.

The following points summarize the main improvements our artifact provides:

- The artifact collects all information categorized as historical data from a TD backlog so that it can be easily accessed and analyzed by technical and non-technical stakeholders.
- It serves as a tool to translate large and complex historic record information into easily comprehensive visualizations, making it easier to identify patterns, trends, and anomalies.
- The visualizations help the stakeholders monitor the changes in repaid TD over time, compared to unresolved TD, goals, TD types, velocity, and risk.
- Otherwise, invisible insights are made visible to technical and non-technical stakeholders, which increases their knowledge of TD in their system(s).
- It is fundamental for a precise and common understanding of TD between non-technical and technical stakeholders to aid communication between them.
- It offers TD progress transparency to achieve accurate project measures for future strategies, such as more precise capacity planning.

## 6.2 Lessons learned

**Research design**

First and foremost, we experienced that DSR proves to be an efficient methodology for good communication between the fields of research and industry. By developing artifacts targeted to solve problems identified in the industry, we experienced genuine interest in our research. Moreover, our artifact inspired the practitioners to think differently about their TD management. For example, in the last evaluation, the participants mentioned that they would consider applying our ideas to their practice. At the same time, we received valuable feedback from them on changes in the design to make it more suitable for their practices.

Further, we experienced that finding the right participants for the evaluation is crucial for gaining valuable insight. As mentioned earlier, the subject of the thesis, TD management, requires competencies that a limited amount of employees have. However, we would highly recommend putting extra effort into finding these because it would yield valuable feedback. Due to this fact, it could be preferable to choose a methodology that studies a company in debt, such as a case study.

A challenge we experienced concerning data analysis was deciding if it only applied to a certain company or could be applied to improve all. We believe that this is a particular hard task in DSR and other methodologies that workers closely with few practitioners.

**Artifact**

The artifact should provide options that let the team tailor the visualizations/artifact to suit their processes. To do this, one has to implement a solution that considers that each team uses different terms/values to explain the same things, such as how to define work completed. For example, a solution is to offer different options for the horizontal- and vertical axis. Project teams also use different frameworks. To make the artifact more applicable for different teams, one should use generic terms. E.g., avoid framework-specific terms such as Sprint/Story Points.

Utilizing existing practices from agile management gave us a good starting point in designing our artifact. On reflection, we have realized that this has influenced some of our design choices. It will be beneficial to look at the problem from several perspectives, which may be beneficial in future work.

The visualizations that are mainly useful for the non-technical stakeholders or the management should be designed for less frequent use and minor detail. Additionally, aggregation of data in longer time intervals is preferred. On the other hand, the visualizations that are mainly useful for the team should be designed for more frequent use and include more details, e.g., TD types and specific numbers. These lessons lead us to believe that the artifact could benefit from splitting up the visualizations into two views. One view should appeal to the non-technical stakeholders, and one view should appeal to technical stakeholders.

The artifact should be incorporated into an artifact that visualizes the existing TD in a backlog. As we experienced both in the findings and relevant research, the unresolved TD is the biggest concern for the stakeholders. Therefore, these functions presented here should act as a supplement to existing solutions for managing TD.

## 6.3 Related work

We could only identify one empirical research done on the visualization of repaid debt. Power [36] describes a simple technique for visualizing, quantifying, and tracking a team's TD as a portion of their overall capacity investment. The study considers only how much capacity the team is investing in reducing and managing TD regarding visualization. The distributions are presented in pie-and bar charts. The work is not related to a tool but as a visualization technique in a retrospective context. We chose to include the pie-chart visualization technique in our artifact to investigate if the idea were also applicable for representing long-term effects and as a part of an artifact.

However, there exists empirical research done on visualization on TD in general. It relates to our research as it offers solutions that similarly want to visualize the TD. The difference is that such tools only cover TD that can be detected automatically in the source code. The TD is not documented as TD items, which makes our research innovative. There are currently no technologies based on TD backlogs that offer visualization of repaid TD.

Project Management Tools also offer different visualizations of a backlog, such as Jira. However, these are no dedicated solutions for TD.

There exist different tools for managing TD, and our artifact tries to improvement these. However, the improvement does not try to replace any existing solution but works in addition to current TD management. Our solution is innovative; no artifact offers dedicated TD visualizations on repaid debt. The related work may have similar visualizations, but the intention is to cover a smaller area of TD (e.g., code debt) or a larger area than just TD (project management). The sources are used as inspiration when building our artifact.

## 6.4 Research quality

### 6.4.1 Validity

The concept of validity measures the accuracy of the findings that are derived from the study [26]. To determine the different aspects of validity, we look at the ones listed for case study research[43]: construct-, internal-, external validity, and reliability. The internal validity aspect is not relevant to our study, as it concerns casual relations between the studied objects.

**Construct validity** Construct validity reflects *"to what extent the operational measures that are studied really represent what the researcher has in mind and what is investigated according to the research questions [43]."* A typical threat to construct validity is misunderstandings. We did multiple actions to avoid this threat. First, in each interview, we started by discussing the concept of TD so that we achieved a common understanding for the rest of the interviews. Common understanding was essential in the initial interviews when the study objective was less concrete. Secondly, we conducted pilot interviews to ensure the research questions were understandable. Next, in the last evaluation, the TD items used as input in the artifact were fake data produced by the researchers. Therefore, the data does not necessarily represent what it would look like in a real project. However, this was conveyed to the participants in the different data collections to make them aware of this drawback. Additionally, to avoid leaving out important perspectives on TD management, we included both technical and non-technical stakeholders from four different companies.

**External validity** The aspect of external validity is concerned with to what extent it is possible to generalize the findings and to what extent the findings are of interest to other people outside the investigated case [43]. The generalization aspect of our research study is analytical considerations and not statistical. The DSR evaluation proves that the artifact's visualizations are useful and understandable according to its current DSR-phase. We have also proven that the solution is efficient by implementing a working artifact prototype that works with the project management tool Trello. However, these results are based solely on the literature review and interviews with four different companies, making generalization difficult. Future work needs to investigate the artifact in use. However, the external validity is increased by including different companies with different TD management processes in all of the data collections. Additionally, the artifact is designed for TD management where TD backlogs are recorded; we can therefore not generalize to companies that do not record TD in a backlog.

**Reliability** The aspect of reliability is concerned with to what extent the data and the analysis are dependent on the specific researchers [43]. Reliability threats are especially prominent in qualitative data analysis. The subjective nature of the reading text makes the analysis vulnerable to researcher bias. A typical and essential way of mitigating this threat is through triangulation [43]. We have mitigated this threat through source triangulation by including different companies and roles in our study. However, we have not used triangulation of the methodology or observer. We only used qualitative methodology and the data is analyzed and interpreted by only one researcher, which decreases the reliability.

Besides this, we have been working closely with potential users and presenting the findings to them in terms of new design choices. We would be told indirectly through new evaluations if our interpretations were wrong. Additionally, by recording the interviews and using analytical methods for analysis, the reliability of the study is increased.

### 6.4.2 DSR guidelines

Hevner et al. [22] provide seven guidelines on how to foster effective DSR. In this section we will evaluate our research design by discussing how we have followed the DSR guidelines.

**Guideline 1: Design as an Artifact** *"Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation"* [22].

Our contribution is an IT artifact in the form of an instantiation (website) that serves as a tool to improve TD management. The evolution of the artifact is thoroughly described in Chapter 4 (Iterations), and an overall presentation of the artifact are presented in Chapter 5 (Artifact).

**Guideline 2: Problem Relevance** *"The objective of design-science research is to develop technology-based solutions to important and relevant business problems"* [22].

The research was initiated from a problem observed in practice. First, we explored the knowledge base by conducting a literature review on relevant literature. Next, we got to know the environment relevant to the research by conducting interviews. The interviews were conducted with people in the problem domain to ensure research relevance. The goal of these activities was to find out what the business needs were in that environment. Further, we ensured problem relevance by using the business problems as a basis for a new artifact.

**Guideline 3: Design Evaluation** *"The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods"* [22].

Through two iterations, the artifact was evaluated using the methodology interviews, including prototyping and live demonstration techniques. The methodology is described thoroughly in Chapter 3. The utility and understandability of the artifact are proven and presented in the last part of Chapter 4 (Iterations). However, the evidence of utility can be stronger by evaluating the artifact in use, for example, in a case study. Then, it will also be more relevant to study quality attributes.

Further, efficacy refers to the degree to which the artifact produces its desired effect considered narrowly, without addressing situational concerns [21]. Therefore, we can argue that the historical records efficacy was proven by designing and implementing the artifact (both as a prototype and implementation).

To limit the scope of the evaluations, we only evaluated the visualization features, as they were the most interesting for the thesis. The other features were included in the live demonstrations to give the user the initial idea of the artifact, but we did not ask questions about them.

**Guideline 4: Research Contributions** *"Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies"* [22].

The designed artifact constitutes a research contribution as an extension to the knowledge base. In this chapter, we discussed the theoretical contributions provided by the designed artifact. In addition, the contribution includes the lessons learned when developing and evaluating the artifact.

**Guideline 5: Research Rigor** *"Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact"* [22].

When we have produced a design for the artifact, we have continuously evaluated the result against the knowledge base for verification. Whether this verification proves correct or false, it serves as an addition to the knowledge base.

**Guideline 6: Design as a Search Process** *"The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment"* [22].

The design of our artifact has been iterative. We have conducted two iterations, and the results from the last one argue the need for further iteration to improve both usability aspects and the understandability of the visualizations. If the improved visualizations satisfy these measures, we can extend the scope of evaluation to include the other aspects of the artifact or/and in more realistic contexts.

**Guideline 7: Communication of Research** *"Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences"* [22].

The artifact is designed to be suitable for both non-technical and technical stakeholders. Therefore, we have included and concerned both perspectives in the artifact's evolution and the presentation of the findings.

## 6.5 Future work

The resulting artifact is rudimentary due to a novel research field. Future work on the historical record should pick up and follow from the last stage of our partial DSR research. To do this, one should consider the understandability issues and the improvements mentioned in the findings and the summary (see Table 4.2).

Secondly, one should consider if other functionalities should be explored. For example, an alternative could be the user stories in the backlog that were not implemented, which concerned cost-benefit analysis. Additionally, one should consider exploring the existing representations in more depth. One example can be to look into a forecasting feature that automatically calculates a forecast based on the trends. Finally, besides the specific improvement mentioned for each visualization, one should also consider the ”lessons learned” (see Section 6.2), which discusses practical thoughts on how the artifact should be implemented.

After improving the artifact and implementing a more robust solution, one could conduct an empirical study of the artifact in a natural environment to evaluate its effectiveness. The artifact should be treated as a supplement to existing solutions for managing TD in a backlog. Therefore the artifact must be incorporated into a context where all of the fundamental TD activities (identify, measure, repay) [29] are in place to find out how the solution can improve the TD management.

# Chapter 7

# Conclusion

Technical debt negatively affects the company's ability to deliver customer value. To identify and manage technical debt continuously, one benefits greatly from using a management tool. We identified a potential for improvement in the current tools that handle TD [29]. Non-technical or technical stakeholders should be able to navigate and analyze their historical TD data to improve their management of TD. In order to do this, the historical data needs to be accessible and informative so that all stakeholders can correctly understand the company's TD. Such insight can be crucial to grasping the right future strategies for TD management. The research aimed to identify solutions to how one should visualize the historical data to improve the management of TD (RQ).

We have used design science research to develop an artifact containing six different visualizations of historical project data. As a part of this research design, we have conducted qualitative data collections with seven people from four different companies. The results reveal how, where, and when the visualizations are useful for whom (non-technical or/and technical stakeholders. Additionally, it shows their level of understandability and areas for improvement.

The findings support that stakeholders can use the visualizations to improve TD monitoring and communication. We proposed a new approach to TD monitoring that watched the amount of resolved TD over time compared to other important metrics including unresolved TD, effort goals, TD types, velocity, and risk. Additionally, we improved communication by presenting resolved TD items in useful and understandable graphical representations such as bar charts and pie charts.

# Bibliography

[1] Nicolli S.R Alves et al. "Identification and management of technical debt: A systematic mapping study". eng. In: *Information and software technology* 70 (2016), pp. 100–121. ISSN: 0950-5849.

[2] *Archive*. Oxford Advanced Learner's Dictionary. URL: https://www.oxfordlearnersdictionaries.com/definition/english/archive_1 (visited on 2022).

[3] Atlassan. *Story points and estimation*. July 2020. URL: https://www.atlassian.com/agile/project-management/estimation.

[4] Atlassan. *View and understand the velocity chart: Jira Software Cloud*. July 2020. URL: https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-velocity-chart/.

[5] Paris Avgeriou et al. "Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162)". In: *Dagstuhl Reports* 6 (Jan. 2016). DOI: 10.4230/DagRep.6.4.110.

[6] Terese Besker et al. "Embracing Technical Debt, from a Startup Company Perspective". In: *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (2018), pp. 415–425.

[7] Jan vom Brocke, Alan Hevner, and Alexander Maedche. "Introduction to Design Science Research". In: Sept. 2020, pp. 1–13. ISBN: 978-3-030-46780-7. DOI: 10.1007/978-3-030-46781-4_1.

[8] Nanette Brown et al. "Managing Technical Debt in Software-Reliant Systems". In: *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*. FoSER '10. Santa Fe, New Mexico, USA: Association for Computing Machinery, 2010, pp. 47–52. ISBN: 9781450304276. DOI: 10.1145/1882362.1882373. URL: https://doi-org.ezproxy.uio.no/10.1145/1882362.1882373.

[9] Mahil Carr, June Verner, and Dr. June Verner. *Prototyping and Software Development Approaches*. 1997.

[10] Victoria Clarke and Virginia Braun. “Thematic analysis”. In: *The Journal of Positive Psychology* 12.3 (2017), pp. 297–298. DOI: 10.1080/17439760.2016.1262613. eprint: https://doi.org/10.1080/17439760.2016.1262613. URL: https://doi.org/10.1080/17439760.2016.1262613.

[11] Ward Cunningham. “The WyCash Portfolio Management System”. In: *SIGPLAN OOPS Mess.* 4.2 (Dec. 1992), pp. 29–30. ISSN: 1055-6400. DOI: 10.1145/157710.157715. URL: https://doi.org/10.1145/157710.157715.

[12] Bill Curtis, Jay Sappidi, and Alexandra Szynkarski. “Estimating the Principal of an Application's Technical Debt”. In: *Software, IEEE* 29 (Nov. 2012), pp. 34–42. DOI: 10.1109/MS.2012.156.

[13] José Osvaldo De Sordi. *Design Science Research Methodology: Theory Development from Artifacts*. eng. Cham: Springer International Publishing AG, 2021. ISBN: 9783030821555.

[14] Georgios Digkas et al. “The Risk of Generating Technical Debt Interest: A Case Study”. eng. In: *SN computer science* 2.1 (2020). ISSN: 2662-995X.

[15] Sávio Freire et al. “Actions and Impediments for Technical Debt Prevention: Results from a Global Family of Industrial Surveys”. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1548–1555. ISBN: 9781450368667. URL: https://doi-org.ezproxy.uio.no/10.1145/3341105.3373912.

[16] Imran Ghani. *Handbook of research on emerging advancements and technologies in software engineering*. IGI Global, 2014, pp. 597–611.

[17] AG Grid. *AG Grid*. ”n.d.” URL: https://react-grid.ag-grid.com/.

[18] Recharts Group. *ReCharts*. Aug. 2022. URL: https://recharts.org/en-US/.

[19] Constantin Guay. *Counting technical debt in velocity or not?* Aug. 2018. URL: https://const.fr/blog/agile/counting-technical-debt-velocity-not/.

[20] Yuepu Guo and Carolyn Seaman. “A Portfolio Approach to Technical Debt Management”. In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. MTD '11. Waikiki, Honolulu, HI, USA: Association for Computing Machinery, 2011, pp. 31–34. ISBN: 9781450305860. DOI: 10.1145/1985362.1985370. URL: https://doi.org/10.1145/1985362.1985370.

[21] Alan Hevner et al. "A pragmatic approach for identifying and managing design science research goals and evaluation criteria". In: *AIS SIGPrag Pre-ICIS workshop on "Practice-based Design and Innovation of Digital Artifacts"*. San Francisco, United States, Dec. 2018, p. . URL: `https://hal.archives-ouvertes.fr/hal-02283783`.

[22] Alan R. Hevner et al. "Design Science in Information Systems Research". eng. In: *MIS quarterly* 28.1 (2004), pp. 75–105. ISSN: 0276-7783.

[23] Barbara Kitchenham. "Procedures for Performing Systematic Reviews". In: *Keele, UK, Keele Univ.* 33 (Aug. 2004).

[24] Tim Klinger et al. "An Enterprise Perspective on Technical Debt". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. MTD '11. Waikiki, Honolulu, HI, USA: Association for Computing Machinery, 2011, pp. 35–38. ISBN: 9781450305860. DOI: `10.1145/1985362.1985371`. URL: `https://doi-org.ezproxy.uio.no/10.1145/1985362.1985371`.

[25] P. Kruchten, R. L. Nord, and I. Ozkaya. "Technical Debt: From Metaphor to Theory and Practice". In: *IEEE Software* 29.6 (2012), pp. 18–21. DOI: `10.1109/MS.2012.167`.

[26] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction*. 2nd ed. Cambridge, MA: Morgan Kaufmann, 2017. ISBN: 978-0-12-805390-4. DOI: `http://www.sciencedirect.com/science/book/9780128053904`. URL: `https://www.safaribooksonline.com/library/view/research-methods-in/9780128093436/`.

[27] Valentina Lenarduzzi et al. "A systematic literature review on Technical Debt prioritization: Strategies, processes, factors, and tools". eng. In: *The Journal of systems and software* 171 (2021). ISSN: 0164-1212.

[28] J. Letouzey. "The SQALE method for evaluating Technical Debt". In: *2012 Third International Workshop on Managing Technical Debt (MTD)*. 2012, pp. 31–36. DOI: `10.1109/MTD.2012.6225997`.

[29] Zengyang Li, Paris Avgeriou, and Peng Liang. "A systematic mapping study on technical debt and its management". In: *Journal of Systems and Software* 101 (2015), pp. 193–220. ISSN: 0164-1212. DOI: `https://doi.org/10.1016/j.jss.2014.12.027`. URL: `https://www.sciencedirect.com/science/article/pii/S0164121214002854`.

[30] Md Abdullah Mamun et al. "Evolution of Technical Debt: An Exploratory Study". In: Oct. 2019.

[31] Salvatore T March and Gerald F Smith. "Design and natural science research on information technology". eng. In: *Decision Support Systems* 15.4 (1995), pp. 251–266. ISSN: 0167-9236.

[32] Antonio Martini. "Anacondebt: A Tool to Assess and Track Technical Debt". In: *Proceedings of the 2018 International Conference on Technical Debt*. TechDebt '18. Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 55–56. ISBN: 9781450357135. DOI: 10.1145/3194164.3194185. URL: https://doi.org/10.1145/3194164.3194185.

[33] Antonio Martini, Terese Besker, and Jan Bosch. "Technical Debt tracking: Current state of practice: A survey and multiple case study in 15 large organizations". In: *Science of Computer Programming* 163 (2018), pp. 42–61. ISSN: 0167-6423. DOI: https://doi.org/10.1016/j.scico.2018.03.007. URL: https://www.sciencedirect.com/science/article/pii/S0167642318301035.

[34] Antonio Martini, Terese Besker, and Jan Bosch. "The Introduction of Technical Debt Tracking in Large Companies". eng. In: *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2016, pp. 161–168. ISBN: 1509055754.

[35] 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. DOI: 10.2753/MIS0742-1222240302. eprint: https://doi.org/10.2753/MIS0742-1222240302. URL: https://doi.org/10.2753/MIS0742-1222240302.

[36] Ken Power. "Understanding the Impact of Technical Debt on the Capacity and Velocity of Teams and Organizations: Viewing Team and Organization Capacity as a Portfolio of Real Options". In: *Proceedings of the 4th International Workshop on Managing Technical Debt*. MTD '13. San Francisco, California: IEEE Press, 2013, pp. 28–31. ISBN: 9781467364430.

[37] Jan Pries-Heje, Richard Baskerville, and John R. Venable. "Strategies for Design Science Research Evaluation". In: *ECIS*. 2008.

[38] Dan Radigan. *Supercharge the created versus resolved gadget!* Apr. 2019. URL: http://atlassianblog.wpengine.com/2013/04/supercharge-the-created-vs-resolved-gadget/.

[39] Gary Rafferty. *Jira Dashboards for an Agile Project*. 2020. URL: https://garyrafferty.com/software/agile/Agile-Dashboard-In-Jira/ (visited on 03/22/2022).

[40] React. *Tutorial: Intro to React*. "n.d." URL: https://reactjs.org/tutorial/tutorial.html.

[41] Kamma Luise Rodian. *9 ways to measure the health of your sprints*. 2021. URL: https://www.forecast.app/blog/sprint-metrics.

[42] Kenneth S Rubin. *Essential Scrum: a practical guide to the most popular Agile process*. eng. The Addison-Wesley signature series. Addison-Wesley Professional, 2012. ISBN: 9780137043293.

[43] Per Runeson and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering". eng. In: *Empirical software engineering : an international journal* 14.2 (2008), pp. 131–164. ISSN: 1382-3256.

[44] Klaus Schmid. "A formal approach to technical debt decision making". eng. In: *Proceedings of the 9th international ACM Sigsoft conference on quality of software architectures*. QoSA '13. ACM, 2013, pp. 153–162. ISBN: 1450321267.

[45] Carolyn Seaman and Yuepu Guo. "Chapter 2 - Measuring and Monitoring Technical Debt". In: ed. by Marvin V. Zelkowitz. Vol. 82. Advances in Computers. Elsevier, 2011, pp. 25–46. DOI: https://doi.org/10.1016/B978-0-12-385512-1.00002-5. URL: https://www.sciencedirect.com/science/article/pii/B9780123855121000025.

[46] Christian Sonnenberg and Jan vom Brocke. "Evaluation Patterns for Design Science Research Artefacts". eng. In: *Communications in Computer and Information Science*. Vol. 286. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 71–83. ISBN: 9783642336805.

[47] Christian Sonnenberg and Jan vom Brocke. "Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research". In: vol. 7286. May 2012, pp. 381–397. ISBN: 978-3-642-29862-2. DOI: 10.1007/978-3-642-29863-9_28.

[48] Ali Tizkar and Naser M. Tabatabaei. "Rapid Prototyping for Software Projects with User Interface". In: *Scientific Bulletin of University of PITESTI, Electronics and Computer Science Series* 2 (Oct. 2009), p. 85.

[49] Trello. *Archiving and deleting cards*. Mar. 2021. URL: https://help.trello.com/article/795-archiving-and-deleting-cards.

[50] Trello. *Trello Api Documentation*. Apr. 2020. URL: https://help.trello.com/article/756-trello-api-documentation.

[51] Jesse Yli-Huumo, Andrey Maglyas, and Kari Smolander. "How do software development teams manage technical debt? – An empirical study". eng. In: *The Journal of systems and software* 120 (2016), pp. 195–218. ISSN: 0164-1212.

[52] Jesse Yli-Huumo, Andrey Maglyas, and Kari Smolander. "The Sources and Approaches to Management of Technical Debt: A Case Study of Two Product Lines in a Middle-Size Finnish Software Company". eng. In: *Lecture notes in computer science*. Vol. 8892. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 93–107. ISBN: 9783319138343.

[53] Nico Zazworka, Carolyn Seaman, and Forrest Shull. "Prioritizing Design Debt Investment Opportunities". In: *Proceedings of the 2nd Workshop on Managing Technical Debt*. MTD '11. Waikiki, Honolulu, HI, USA: Association for Computing Machinery, 2011, pp. 39–42. ISBN: 9781450305860. DOI: 10.1145/1985362.1985372. URL: https://doi-org.ezproxy.uio.no/10.1145/1985362.1985372.

[54] Nico Zazworka et al. "A Case Study on Effectively Identifying Technical Debt". In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. EASE '13. Porto de Galinhas, Brazil: Association for Computing Machinery, 2013, pp. 42–47. ISBN: 9781450318488. DOI: 10.1145/2460999.2461005. URL: https://doi.org/10.1145/2460999.2461005.

# Appendices

# Appendix A

# Interview Consent From

## Samtykkeerklæring for intervju i forbindelse med masteroppgave om teknisk gjeld

*Beskrivelse av prosjektoppgaven*

Jeg er en masterstudent ved Institutt for informatikk, Universitetet i Oslo. Jeg skriver en masteroppgave under forskningsgruppen Software Engineering på tema teknisk gjeld. Veilederen min er Antonio Martini.

*Kontaktperson:*

Vilde Rabbe Husby +47 94828633 vilderhu@matnat.uio.no

Oppgaven min tar for seg håndtering av teknisk gjeld ved bruk av verktøy og hvordan et slikt verktøy kan forbedres til å være mer nyttig for utviklere/ledere. Som en del av masteroppgaven skal jeg undersøke hvordan teknisk gjeld som er blitt tilbakebetalt kan bidra til å gi verdifull informasjon for utviklere/ledere.

Jeg ønsker dermed å intervjue deg om temaet teknisk gjeld i din organisasjon. Formålet med intervjuet er å få en forståelse av organisasjonen håndtering av teknisk gjeld og hvordan denne håndteringen kan forbedres. Hvis det er ønskelig kan jeg skrive et referat av intervjuet som du kan lese i ettertid og rette opp i eventuelle feil.

*Frivillig deltakelse*

All deltakelse er frivillig, og du kan trekke deg når som helst. Jeg ønsker å ta opptak av dette intervjuet. Dette opptaket vil vi kun bruke som informasjon til masteroppgaven og det er bare jeg som kommer til å høre dette. Vi ønsker også å notere under og etter intervjuet. Du kan når som helst avslutte intervjuet eller trekke tilbake informasjon som er gitt under intervju. Hvis du tillater opptak, vil dette opptaket slettes senest 01.06.2022.

*Anonymitet*

Alle vi intervjuer vil være anonyme i rapporten. Det vil ikke bli notert eller lagret noen form for personopplysninger. Notater fra intervjuene vil bare leses av med og veilederen min. Notatene og rapporten vil bli anonymisert: ingen andre enn meg og veilederen min vil vite hvem som er blitt intervjuet, og det du sier i intervjuet vil ikke kunne tilbakeføres til deg. Før intervjuet begynner, ber jeg deg om å samtykke i deltagelsen ved å undertegne på at du har lest og forstått informasjonen på dette arket, og ønsker å stille opp til intervju.

*Samtykke*

Jeg har lest og forstått informasjonen over og gir mitt samtykke til å bli intervjuet

__________ ________________________________________

Sted og dato Signatur

# Appendix B

# Interview guide for initial interviews

# Interview guide

**Goal:** The goal of this interview is to gain insight into how an organization deals with technical debt with a focus on aspects concerning repaid debt items. We want to find out what information is valuable to technical and non-technical stakeholders and how this best can be visualized to the user.

1. Technical
    a. Consent form
    b. Turn on recordings (if allowed)

2. Work and role
    a. Work/school background
    b. Years of experience
    c. Role in team
    d. Team environment

3. Technical Debt in the workplace
    a. Definition/Description
    b. TD management/assessment
    c. Refactoring/analyze
    d. Repaid TD/Archiving
    e. Analyze repaid TD

4. Cost-benefit analysis
    a. Ask about knowledge and usage
    b. Explain how it can be done
    c. Examples of how this may be relevant to a historical record

5. TD backlog
    a. Illustration of a TD backlog with current and repaid technical debt.
    b. Valuable information about the TD items
    c. Valuable information about the relationship between the items.

6. Burndown chart
    a. Knowledge of graph
    b. Explain the graph & its purpose
    c. Explain how such a graph can be applied to technical debt
    d. The usefulness of the proposed graph

7. Created vs. resolved chart
    a. Knowledge of graph
    b. Explain the graph & its purpose
    c. Explain how such a graph can be applied to technical debt
    d. The usefulness of the proposed graph

5. Historical Record
    a. Thoughts on motivation & purpose
    b. Thoughts on planned functionality
    c. Ideas for other functionalities

6. Concluding questions
    a. Questions from the respondent
    b. Turn off recordings

# Appendix C

# Evaluation Interview guide

# Evaluation Interview guide

**Goal:** The main goal of the evaluation is to determine how well the artifact achieves its expected main purpose. The blue text represents high-level topics that tell us what information we are looking for from the respondent. The green text represents the steps in the demonstration.

1. Technical
    a. Consent form
    b. Turn on recordings  (if allowed)
2. The participant
    a. Work and role
    b. TD knowledge and practice
3. Artifact background
    a. Purpose and motivation of the study
    b. Assumptions: backlog & item properties
    c. Example backlog: Trello
    d. Data relevant to the historical record
4. Historical record: list view
    a. Updating.
    b. Properties.
    c. Sorting.
    d. Searching.
5. Historical record: Modal view
    a. More information about each debt item
    b. Change story points
    c. Archive/unarchive
    d. Open in Trello
6. Historical record: Graph overview
    a. Graphs that visualize the data in different ways
7. Graph: Technical Debt Project progress
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements
8. Graph: Identifier vs.resolved debt items
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements.
9. Graph: Velocity including and excluding TD
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements.
10. Total risk
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements.
11. Distribution of total effort in percentages
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements
12. Distribution of TD effort for each module in percentages
    a. The overall usefulness of the graph
    b. Who will use the graph
    c. When the graph would be used
    d. Improvements
13. Concluding questions
    a. Questions from the respondent
    b. Ideas to other functionalities
14. Technical
    a. Turn off recordings

# Appendix D

# Final backlog

## Final backlog

| Num. | User story | Motivation / How to / Assumptions | Prio. |
|---|---|---|---|
| 1. | As a team member, I want my TD items marked as done to be available in an archive, so that the repaid data is accessible for later use. | *Motivation*: The repaid TD items are the fundamental data for our archive.<br>*Assumption*: The team specifies which of the TD items are done.<br>*How to:* This can be done automatically when a user marks it as done to minimize their workload. | High |
| 2. | As a team member, I want to look up and search for repaid TD items in an archive so that I can find specific repaid TD items at any time, for example for reporting reasons. | *Motivation*: An archive will eventually grow to a big extent, and therefore it can be useful to search for items to find what you are looking for. We will also argue that this is a common functionality for an archive. | Low |
| 3. | As a team member, I want to restore a repaid TD item from the archive so that I can modify/redo the missing functionality. | *Motivation*: We will argue that this is a common functionality for an archive. It can happen that people mark a TD item as done, and later find out that it wasn't solved right/completely. Then the user needs the functionality that enables them to restore the item and put it back in the TD backlog. | Low |
| 4. | As a user, I want to know at what time a TD item was created and marked as done so that it can be sorted chronologically. | *Motivation*: For the user to get an overview of the repayment activities, they need to know this information. For example, this information can be useful to make a visual overview of how the TD items are created and repaid over time.<br>*Assumption*: the properties created and resolved are stored as TD item properties | High |
| 5. | As a team member/business leader I want to know the story points/time used to repay an item, for each TD item, so I am aware of how resources are used. | *Motivation*: The motivation for this user story is from the data collection and the literature review. Story points/Time used in development is a metric that is important for developers since it says a lot about the complexity of a task and therefore also the application's complexity. This metric can also be valuable for the leaders since it tells them where and how the resources are used.<br>Data collection:<br>- *"The biggest spaghetti coder… But he made it sick fast. He got a lot of credit for working so fast, from the* | High |

| | | | |
|---|---|---|---|
| | | - *project manager. .. and you want to preserve your "credit" .. not be the slowest in the class."*<br>- *"A frustration in the cloud, where you spend a lot of time and suddenly have to throw away everything that has been done.".* We can document "time lost" due to TD to create understanding.<br>Literature review:<br>- Curtis et al. \cite{CurtisSappidiSzynkarski} argues that the time to fix a violation could be available from historical effort data. In their article, the number of hours to fix each violation is one of three factors in calculating the principal.<br>*Assumption*: Time spent on repaying TD is recorded in some sort of way.<br>*How to:* This can for example be done when the developer makes a commit, by specifying an argument "time" followed by a number of hours used on TD and optionally total time used. Another option is to specify it when the developer moves the item to "done" in the backlog. | |
| 6. | As a team member/business leader I want to know what was the estimated time to solve the task, for each TD item, so that I can compare it to the actual time used (in story points). | *Motivation*: The motivation for this user story is from the data collection. This gives both the leaders and developers more insight into whether their expectations are right or not. This information can potentially be used as a reference point when estimating expected time use for new TD items.<br>*Assumption*: Estimated time to solve the task is recorded for each TD item. | High |
| ~~7.~~<br>Not completed | As a team member/business leader I want to know the expected and actual cost-benefit of solving a TD item for each TD item so that I know if we solved the right item. | *Motivation*: I consider this a potentially useful functionality. Showing how much time was saved by repaying the item.<br>*Assumption*: The team uses cost-benefit analysis on the TD backlog. We also assume that each TD item is associated with an anticipated cost-benefit variable and an actual cost-benefit variable.<br>*How to:* These two values have a positive or a negative result that can increase or decrease over time. Based on this we can say something about our analysis and determine if the calculation was right, and if we solved the right item. | Low |
| 9. | As a team member, I want to know who was responsible for repaying an item, for each TD item, so | *Motivation*: I consider this a potentially useful property of the element. Showing who repaid which item could also be an incentive for developers to solve more items. | Medium |

| | | | |
|---|---|---|---|
| | that questions or information related to it can be asked to the right people. | | |
| ~~10.~~<br>Deleted | ~~As a team member, I want the items in the archive to be categorized by the type and size of the TD item so that I can differentiate them and compare them.~~ | *Motivation*: We consider this a potentially useful property of the element. This can be valuable information for analyzing and visualizing purposes. Size can make the proportions between the items more accurate and type can help differentiate them giving more detail.<br>*Assumption*: The properties type, size, and location are TD items properties. | High |
| 11. | As a team member/business leader I want to see a visual overview of repaid TD items in the archive so that I get a holistic overview of payment progress. | *Motivation*: data collection and literature review. Both sources emphasize that visual representations are a good way to get a better understanding of TD.<br>Example from the data collection:<br>- The management benefits from it [visualization of TD], and often the developers are aware of it…<br>Literature review:<br>- Significant problems arise when debt is not visible to business stakeholders since they are the ones responsible for paying the debt. Visualization of TD is used as a means to improve stakeholder communication in the decision-making process. …. transform intangible software entities and their relationship into visual metaphors that are easily interpreted by human beings. | High |
| ~~12.~~<br>Deleted | ~~As a team member/business leader I want to know the total number of TD items solved for each TD size, for a given period, so that I know how much resources are used on TD.~~ | *Motivation*: We consider this as useful information explaining how resources are used. The data can also be used for other purposes, such as calculating the trend for the number of solved TD items and comparing it to a goal.<br>*How to:* If we use the total number of TD items, it has to be divided into size categories, so that the number is representative of the amount of work.<br>that people just solve small TD items to achieve a higher number.<br>Alternatively use another metric, such as time used on repaying debt or story points. | ~~Low~~ |
| 13. | As a team member, I want to have a predefined/optimal goal for number | *Motivation*: The motivation for this user story is from the data collection. An incentive to repay TD.<br>Example: | High |

| | | | |
|---|---|---|---|
| | solved debt items for a given period, so that I get motivated to achieve this goal together with my team. | - *“Two things that apply to a team's success: common goals and role distribution. Need active measures to get people to take on tasks no one wants to take. A common goal so that everyone knows it must be done”*<br>*Assumption*: This assumes that the team defines a goal.<br>*How to:* Another option is to calculate a goal based on team velocity, if the velocity is high, then the goal should be higher and vice versa. Should be tailored to each team and what suits them best. | |
| ~~14..~~<br>Deleted | ~~As a team member/business leader I want to see a visual overview of the average total number of TD items solved, for each TD size, for a predefined interval (for example a sprint).~~ | *Motivation*: I consider this as potentially useful information for the members and business leaders. This can be used in comparison with the number for a given sprint, to get to know if the team is doing more or less than average.<br>*How to:* As in 12. If I am using the total number, it has to be divided into size-categories, or it would be misleading. | Low |
| 15.<br>Not completed | As a team member, I want to know the anticipated principal for a TD category (type/location), so I can improve the accuracy of the cost-benefit analysis for TD items in the backlog. | *Motivation*: The motivation for this user story is from the literature review:<br>- Zazworka et al. \cite{ZazworkaSpinolaVetroShullSeaman} propose that historical effort data can be used to make a more accurate and reliable estimation beyond the initial high/medium/low assessment. For example, if a TD item is a set of classes that need to be refactored, the historical cost of modification of those classes can be used as the future modification cost (principal of the TD item) estimation.<br>*Assumption*: there exists some form of cost-benefit analysis. | Low. |
| 16.<br>Not completed | *As a team member, I want to know the anticipated interest of a TD category (type/location), so that I can improve the accuracy of the cost-benefit analysis for current TD items.* | *Motivation*: The motivation for this user story is from the literature review:<br>- Zazworka et al. \cite{ZazworkaSpinolaVetroShullSeaman}  propose that historical usage can be used to improve the estimate of both of these interest conditions. The factor they list as important for this is TD principal, data on past defects, effort, and changes.<br>*Assumption*: that there exists some form of cost-benefit analysis. | Low. |
| 17. | As a team member/business leader I want to | *Motivation*: The motivation for this user story is from the literature review and data collection: | High. |

| | | | |
|---|---|---|---|
| | know the relation between team velocity and the number of repaid TD items so that I can monitor how development speed is affected by the amount of TD. | - ... it is very difficult. Been a theme for many years. To try to convince what needs to be done, but those who are not technical can not keep up.<br>Visualizing how velocity can be affected by TD can possibly clarify the importance of prioritizing TD for non-technical stakeholders.<br>*Assumption*: team velocity is recorded in the project management tool.<br>*How to*: for each sprint, in the end, team velocity is saved and compared with the number of TD items. | |
| 18. | As a team member/business leader I want to see the relationship between created vs. resolved TD items, so I know if there is positive or negative progress. | Motivation: Inspiration from Jira´s created vs resolved gadget, as mentioned in the tool survey \cite{radigan2019}. | High |
| 19. | As a team member/business leader I want to see the total risk reduced for each sprint compared with the total risk left in the system, so I get an overview of risk progress. | Motivation: The motivation for this user story is from the literature review:<br>- Risk analysis is a known technique used to decide on which TD items should be repaid first \cite{DigkasGeorgios2020}. The advantage of using this metric is that it is non-technical and easy to comprehend and therefore a very suitable metric to reach business leaders<br>But also from the data collection as an incentive to repay TD. Example:<br>- *“Two things that apply to a team's success: common goals and role distribution. Need active measures to get people to take on tasks no one wants to take. A common goal so that everyone knows it must be done”*<br>*Assumption*: Risk analysis is a part of the team's workflow, and the risk-value is a TD item property. | High |
| 20.<br>(new user story) | As a team member/business leader I want to see a visual overview of the effort on repaid TD items compared | *Motivation*: The motivation for this user story is from the data collection and literature review.<br>- \cite{Power} argues that pie charts and bar charts that show how much capacity the team is investing in reducing and managing TD are very useful. This is justified by the fact that “they quickly show the relative | High |

| | | | |
|---|---|---|---|
| | to the team's total capacity, for a predefined interval, so that I get more insight on how the resources/effort were distributed." | effort in proportion to the other areas the team is spending their time.<br>This can be used in comparison with the number for a given sprint, to get to know if the team is doing more or less than average.<br>*Assumption*: the team specifies the capacity used on other activities in addition to TD. | |
| 21.<br>(new user story) | "As a business leader/team member I want to see a visual overview of the distribution of TD effort for each module so that I get more insight on how the resources/effort were distributed." | *Motivation:* The motivation for this user story is from the data collection:<br>- Participants in the first evaluation emphasized the benefits of showing how the team's effort has been distributed over time, and that visualization on this is very useful, especially from a business perspective.<br>*Assumption*: the team specifies which module each TD item belongs to. | High |