Usability of a Business Software Solution for Financial Follow-up Information of Service Contracts

Therese Borg

Supervisor: Ola Leifler
Examiner: Aseel Berglund
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page http://www.ep.liu.se/.

© Therese Borg
Abstract

Enterprise Resource Planning systems have been available since the 1990s and come with several business benefits for the users. One of the major advantages is improved decision-making through current and accessible information about strategical, tactical and operational levels of the organization. Although several Enterprise Resource Planning system vendors provide several features for contract management, more decision support regarding the total profitability of service contracts is desired by the customers. Estimating the total profitability of service contracts is a challenging task for all service providers and implies a lot of manual data processing by the contract manager. This master's thesis is conducted in collaboration with IFS World Operations AB and aims to investigate how functionality for budget and forecasting of the profitability of service contracts can be designed to be usable in terms of effectiveness. The implementation was performed iteratively and the resulting prototypes were evaluated and refined throughout the project. The final high-fidelity prototype for budgeting of service contracts was evaluated using the task success rate in conjunction with the System Usability Scale to assess how well the system conformed to the needs of the users. The study revealed that two of the key characteristics of financial follow-up information of service contracts is the support of creating a budget and graphical visualizations of both budgeted and actual values. The final usability evaluation indicated that the developed functionality was usable in terms of effectiveness and has an overall usability clearly above the average.
I would like to direct a special thanks to Camilla, Hans, Maximilian and Björn for giving me a great start of my thesis work, supporting me with technical issues and answering all my questions throughout the project. Also, a big thank you to everyone at IFS and a special thanks to those of you who have participated in interviews and user tests. Finally, I would like to thank my opponent Linn Abrahamsson for providing constructive feedback at the opposition seminar.

Therese Borg

Linköping, June 2018
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Aim</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Research question</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Delimitation</td>
<td>3</td>
</tr>
<tr>
<td>2 Background</td>
<td>4</td>
</tr>
<tr>
<td>2.1 IFS World Operations AB</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Service Contracts</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2 The Framework</td>
<td>5</td>
</tr>
<tr>
<td>3 Theory</td>
<td>6</td>
</tr>
<tr>
<td>3.1 Related Work</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Decision-making Processes</td>
<td>8</td>
</tr>
<tr>
<td>3.2.1 Behavioral Decision-making</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Managerial Decision-making</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Business Intelligence</td>
<td>9</td>
</tr>
<tr>
<td>3.4 Data Analytics</td>
<td>10</td>
</tr>
<tr>
<td>3.5 Information Visualization</td>
<td>11</td>
</tr>
<tr>
<td>3.5.1 Graph Displays</td>
<td>11</td>
</tr>
<tr>
<td>3.5.2 Design Principles</td>
<td>12</td>
</tr>
<tr>
<td>3.6 Usability</td>
<td>14</td>
</tr>
<tr>
<td>3.6.1 Definition</td>
<td>14</td>
</tr>
<tr>
<td>3.7 Design for Usability</td>
<td>14</td>
</tr>
<tr>
<td>3.7.1 Eight Golden Rules</td>
<td>14</td>
</tr>
<tr>
<td>3.7.2 Human-centered Design Principles</td>
<td>16</td>
</tr>
<tr>
<td>3.8 Usability Evaluation</td>
<td>16</td>
</tr>
<tr>
<td>3.8.1 Questionnaires</td>
<td>17</td>
</tr>
<tr>
<td>3.8.2 User Testing</td>
<td>17</td>
</tr>
<tr>
<td>3.8.3 Heuristic Evaluation</td>
<td>20</td>
</tr>
<tr>
<td>3.8.4 Evaluation at Different Stages of the Process</td>
<td>20</td>
</tr>
<tr>
<td>3.8.5 Validity &amp; Reliability</td>
<td>21</td>
</tr>
<tr>
<td>3.9 Software Development Methodologies</td>
<td>22</td>
</tr>
</tbody>
</table>
3.9.1 Plan-driven Software Development .......................... 23
3.9.2 Agile Software Development ................................ 23
3.10 Requirements Engineering .................................... 24
  3.10.1 Interviews ............................................. 26
  3.10.2 User Stories .......................................... 26
  3.10.3 Prototyping ........................................... 27
  3.10.4 Scenarios ............................................. 28
  3.10.5 Sustainability ......................................... 28
3.11 Design Methods ............................................. 29
  3.11.1 Human-centered Design Process ......................... 29
  3.11.2 Personas & Persona-Based Scenarios .................... 29
  3.11.3 User Journey Mapping ................................ 31
  3.11.4 Conceptual Design ................................... 31
4 Method .......................................................... 33
  4.1 Study Structure ............................................ 33
  4.2 Pre-study .................................................. 34
    4.2.1 Research Methodology ................................. 34
    4.2.2 Requirements Elicitation ............................. 35
4.3 Implementation .............................................. 35
  4.3.1 Overview .............................................. 36
  4.3.2 Sprint 1: Conceptual Design - Implementation ......... 36
  4.3.3 Sprint 1: Conceptual Design - Evaluation ............. 39
  4.3.4 Sprint 2: Detailed Design - Implementation .......... 40
  4.3.5 Sprint 2: Detailed Design - Evaluation ............... 40
  4.3.6 Sprint 3: Competitor Analysis ........................ 41
  4.3.7 Sprint 4: High-Fidelity Prototype - Implementation .. 42
  4.3.8 Sprint 4: High-Fidelity Prototype - Usability Evaluation 42
5 Results .......................................................... 45
  5.1 Pre-study .................................................. 45
    5.1.1 Persona .............................................. 45
    5.1.2 Scenarios ............................................ 46
    5.1.3 User Journey ........................................ 46
    5.1.4 Requirements ....................................... 48
    5.1.5 Product Backlog ..................................... 49
  5.2 Implementation ............................................ 49
    5.2.1 Sprint 1: Conceptual Design - Implementation ....... 49
    5.2.2 Sprint 1: Conceptual Design - Evaluation .......... 50
    5.2.3 Sprint 1: Conceptual Design - Final Results ......... 51
    5.2.4 Sprint 2: Detailed Design - Implementation ........ 53
    5.2.5 Sprint 2: Detailed Design - Evaluation ............. 55
    5.2.6 Sprint 2: Detailed Design - Final Results ........... 57
    5.2.7 Sprint 3: Competitor Analysis ....................... 60
    5.2.8 Sprint 4: High-tech Prototype - Implementation .... 61
    5.2.9 Sprint 4: High-tech Prototype - Usability Evaluation 62
6 Discussion ..................................................... 65
  6.1 Results ................................................... 65
    6.1.1 Conceptual Design Prototype ......................... 65
    6.1.2 Detailed Design Prototype ........................... 65
    6.1.3 Competitor Analysis ................................ 66
    6.1.4 High-Fidelity Prototype ............................. 66
<table>
<thead>
<tr>
<th>Appendix: High-Fidelity Prototype - Sprint 4</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.1 Service Contracts Overview Page</td>
<td>120</td>
</tr>
<tr>
<td>H.2 New Budget - Settings</td>
<td>121</td>
</tr>
<tr>
<td>H.3 New Budget - Estimations</td>
<td>122</td>
</tr>
<tr>
<td>H.4 New Budget - Preview</td>
<td>122</td>
</tr>
<tr>
<td>H.5 Contract Analysis Page</td>
<td>123</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix: User Tests - Sprint 4</th>
<th>125</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1 Form - User Test</td>
<td>125</td>
</tr>
</tbody>
</table>
List of Figures

1 Service contract overview ................................................. 5
2 Structure of the Waterfall Model and Agile software development .............. 24
3 Structure of the Human-centered Design Process ................................ 29
4 Study structure ................................................................. 34
5 Conceptual Design Task Analysis .......................................... 37
6 Conceptual Design Functional Chunks ....................................... 37
7 Conceptual Design Conceptual Elements .................................... 38
8 Conceptual Design Reconfigured Model ..................................... 38
9 The user journey created during the requirements elicitation ..................... 47
10 Final Conceptual Design Prototype ........................................ 53
11 Detailed Prototype: Contract Analysis Page ................................... 54
12 First version of the service contracts overview page in Sprint 2 .................. 58
13 Final version of the service contracts overview page in Sprint 2 ................. 58
14 First version of the contract analysis page in Sprint 2 ........................... 59
15 Final version of the contract analysis page in Sprint 2 ........................... 59
16 First version of the budget estimations page in Sprint 2 ......................... 60
17 Final version of the budget estimations page in Sprint 2 ......................... 60
18 First version of the budget preview page in Sprint 2 ............................ 60
19 Final version of the budget preview page in Sprint 2 ............................ 60
20 Page for the service contracts overview list, implemented in sprint 4 ............ 61
21 Page for the service contracts overview cards, implemented in sprint 4 ........ 61
22 Page for the budget settings, implemented in sprint 4 .......................... 62
23 Page for the budget estimations, implemented in sprint 4 ........................ 62
## List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Definitions of the usability attributes of the first level taxonomy</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Classification of requirements</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>Participants of the user test in Sprint 1</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>Participants of the user test in Sprint 2</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Participants of the final usability test in Sprint 4</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Tasks &amp; success criteria used for the final usability evaluation</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Opportunities of improvements &amp; possible solutions</td>
<td>48</td>
</tr>
<tr>
<td>8</td>
<td>ERP Software Comparison</td>
<td>61</td>
</tr>
<tr>
<td>9</td>
<td>Task Success Rate - Results</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>SUS Evaluation Responses</td>
<td>64</td>
</tr>
</tbody>
</table>
Introduction

This chapter aims to introduce the reader to Enterprise Recourse Planning systems and to provide the problem formulation of this master’s thesis. The problem is motivated from a general perspective and the research question and the delimitations are presented.

1.1 Motivation

The need of Enterprise Resource Planning (ERP) systems was realized during the early 1990s. Organizations required support for fast and qualitative decisions of how to utilize resources in both production departments and supportive departments [1]. ERP systems are component-based business modules with seamless data flow among the units. Each business module provides functionality for different departments of the organization, such as financial management, production management and distribution management. The modules are integrated into a unified system and the system provides real time information about the organization [2]. There are numerous potential business benefits of using an ERP system in an organization. One of the benefits is, emphasized by Shang and Seddon in the article "A Comprehensive Framework for Classifying the Benefits of ERP Systems" [3], improved decision-making through current and accessible information about strategical, tactical and operational levels of the organization.

Making good decisions is crucial for the competitiveness and success of an organization. Bad decisions are often made due to the absence of all information needed to evaluate the alternatives and predict the future [4]. The ability to digitally store massive volumes of data has dramatically increased in recent decades. As the scale of data storage grows beyond the capabilities of the human brain, tools to process and analyze this data are becoming critical to extract useful information [5].

The implementation of an ERP system is associated with major costs and complex processes, which often makes it to an extensive decision for organizations [6]. The main expectations of ERP systems are to reduce efficiency costs and to enhance decision-making [7]. Although decision-making objectives have been important in ERP systems over the time, Holsapple
and Sena suggest that decision-support should be higher prioritized as objectives in the development of ERP systems [8].

Decision support systems (DSS) in general provide potential benefits of decreasing the time spent on making decisions, enhancing the information processing of the decision maker and improving the reliability of decision made to complex problems [9]. Holsapple and Sena outline that many researchers have examined the benefits of DDS, but not in the context of DSSs directly integrated into ERP systems [8]. The addition of more functionality to support decisions in an ERP system may include benefits for both the vendor and the customer. For the vendor by gaining market shares and adapting to customer needs, and for the customer by the ability to make better decisions based on real time information from the ERP system.

Making confident decisions has been rated as the most important managerial practice in several studies [9]. This makes functionality of integrated decision support in ERP systems to an interesting field for further studies. The human brain has limitations to process information for decision-making, according to Badddeley [10]. The working memory of the human brain can be described as the ability to remember and repeat several instructions and Baddeley defines the working memory as follows:

> The term working memory refers to a brain system that provides temporary storage and manipulation of the information necessary for such complex cognitive tasks as language comprehension, learning, and reasoning. [10]

There are clear restrictions on the capacity of the working memory and researchers agree that the average number of instructions that the human brain, of a middle age adult, can use for information processing is limited to seven [11]. In the article "A Human Cognition Framework for Information Visualization", Patterson et al. [12], argue that well-designed information visualization encourages the abilities of the human cognition in reasoning and decision-making, and minimizes the risk of missing important information. Baćić and Fadlalla [13] summarize research on business information visualization and claim that the importance of information visualization to support the working memory is widely acknowledged in previous research.

1.2 Aim

The aim of this thesis is to investigate the qualities and characteristics that are required by a system to sufficiently provide financial information on service contracts to satisfy the needs of the end-users and to be usable in terms of effectiveness and subjectiv satisfaction. As a result, the thesis aims to implement financial follow-up functionality of service contracts in the service and management solution in the ERP system IFS Applications. "Follow-up functionality" refers to functionality that makes it possible to estimate the total profitability of a service contract and to compare the estimations with the actual values at any point during the life cycle of the contract. The developed system should satisfy the identified requirements and provide appropriate functionality that can serve as decision-support for the management of service contracts.

1.3 Research question

It is desirable for an organization to make both effective and efficient decisions. Effectiveness means doing the right thing to achieve a certain goal, while efficiency is about completing the task in an optimal way [14]. In a general perspective, there are economic incentives for both effective and efficient decision-making in all profit-making organizations. Decision support systems aim to improve both effectiveness and efficiency in the decision-making process.
Therefore, there are reasons to believe that functionality for decision support directly integrated in an ERP system will have a positive impact on the effectiveness and efficiency of decisions made by the organization using the system.

A service contract is an agreement between a service provider and customer requesting the services. In the article "Forecasting Service Profitability", Blomberg et al. discuss the challenges of cost and revenue estimations of long term service deliveries. Unlike products, the actual cost for a service delivery is unknown at the time of the sale. Several factors that affect the cost is out of the control of the service provider, and often a service agreement last over several years. To make estimations of the total costs and revenues of a service contract is difficult, but it is important to ensure the profitability of the contract, according to Blomberg et al. To fulfill the aim described above, this thesis will answer the following research question:

**How can a business solution for follow-up information of service contracts, in an ERP system, be designed to be usable in terms of effectiveness and perceived subjective satisfaction?**

The business solution should be used to support decisions related to the profitability of the service contracts in the service and management solution of the ERP system. In this study, "designed" refers to what functionality that is needed and how the functionality should be structured in the developed system.

1.4 Delimitation

This master’s thesis is conducted in collaboration with IFS World Operations AB and therefore, the functionality will be developed within the framework of IFS Applications. The client framework is composed by predefined components that can not be manipulated, which will restrict the design and functionality of the final high-tech prototype. However, these restrictions will not be absolutely considered when designing the detailed design prototype.

The added follow-up functionality will be a budgeting functionality and visualization of financial information regarding service contracts within the service and management solution in IFS Applications. Due to time limitations of this master’s thesis, the resulting product aims to serve as a base for further development rather than a complete solution. The evaluation of the usability in terms of effectiveness will be limited to the developed functionality. Although, the concepts of the implemented functionality should be generic enough to be applied to other areas in the ERP system as well.
This chapter presents the background of the problem description from the perspective of the company in cooperation. The chapter aims to provide a deeper understanding of the context of the problem.

2.1 IFS World Operations AB

This master’s thesis is conducted in cooperation with the department of Research & Development, Service & Asset at IFS World Operations AB in Linköping. The company develops and distributes ERP software, IFS Applications and other software products, to customers around the world [16]. Currently, there is limited functionality present for budgeting and forecasting of specific costs of service contracts within the service and management solution in IFS Applications. Functionality for decision support in the management of service contracts is desirable to increase the benefits for the customers and for competitive advantages for the company.

2.1.1 Service Contracts

The service and management solution in IFS Applications is typically used by companies that offer services in different industries to manage service contracts, among other things. A service contract is established between the service provider (the company using the service and management solution) and the customer requesting the services, see Fig. [1]. The contract contains information about the service scope, validity dates, price, etc. The price can be periodic, i.e. the customer pays a monthly or yearly fee for the services included in the contract, or a current fee that depends on how much the service is utilized. There may also be a preventive maintenance connected to a service contract, including the interval for the service to be performed.
Today, some users of the service and management solution are using external Business Intelligence programs or Excel sheets to create reports of data gathered from the service contracts. This requires a considerable amount of manual data processing and does not encourage quick and accurate decisions by the user.

It would be beneficial for the user to be able to visualize trends and estimate key ratios of specific service contracts directly in the ERP system to facilitate decision-making related to the profitability of service contracts. According to Blomberg et al. [15], it is important to analyze the deviation between actual and estimated costs and revenues for the contract to improve future estimations and ensure the profitability during the contract life cycle. Different users of the software require different information about the contracts. As an example, managers directly want to know if a specific contract is profitable or not, while the analysts would like to analyze the non-profitable causes of the contract.

To conclude, this thesis aims to result in added follow-up functionality to visualize more financial information directly in the service and management solution in IFS Applications. Due to the time constraints of this thesis work the result will be limited in functionality but should act as a base for further development.

2.1.2 The Framework

The architecture of the framework consists of three main layers: a web client, a middleware server and a database and business logic layer. The client is built on a framework developed by IFS, which implicates limitations of the visual design and components that can be used for the user interface. All styles, such as color, fonts, etc. are predefined and there is a limited set of components available for use in the client. Other technical aspects, such as details of the architecture and programming languages of the three architectural layers, are not considered relevant for the results of this study, and will not be explained any further.
The following chapter presents the theoretical framework needed to understand the theoretical and technical aspects of the problem, and to develop the prototype needed to answer the research question. The chapter is introduced by a presentation of previous research performed in the area.

3.1 Related Work

The use of visualization to increase usability and user experience of contracts have been studied by Passera in the article “Enhancing Contract Usability and User Experience Through Visualization: an Experimental Evaluation” [17]. Passera discusses the difficulties of understanding traditional text-contracts due to its nature of complexity. The study was conducted in collaboration with a metal and engineering company to create and test prototypes of visualized contracts.

The company used traditional business to business, text-only contracts without any formatting except for tables and capitalized headings. Interviews and workshops were used to collect information about common misinterpretations and elements that could be visualized to improve the usability of the contracts. A prototype was created containing the exact same text but included the following design adjustments [17]:

1. Improving typography and layout of the document, including highlighting key terms in bold typeface
2. Using color as a cue for finding information
3. Adding charts, diagrams, timelines and flowcharts
4. Redesigning the table of contents using color to map recurring parts
5. Redesigning the existing tables
A group of contract users was used to evaluate the contract prototype by answering a questionnaire before, during and after the test session. Additional feedback was gathered through a focus group session. In total, 22 users participated in the questionnaire sessions and the same group also participated in the focus group session. Half of the group worked with the original text-only contract and half of the group worked with the contract prototype using visualization. The questionnaire, answered before the test, consisted of three sections, the first with general questions about age, gender and years of work experience. The second part, answered during the test, included eight comprehension questions about the issues agreed upon in the contract. The test measured both the time used to answer the questions and the correctness of the answers. The final part, answered after the test, consisted of questions related to how easy it was to find and understand the information in the contract. A semi-structured interview format was used during the focus group session.

The results showed that the group working with the visual version of the contract worked faster and provided more accurate answers. The average time spent per question in the second part of the questionnaire was 146 seconds for the group working with the visual version, while the textual version in average required 224 seconds per question. Also, the accuracy of the answers was higher for the visual contract, 72% compared to 60%. During the focus group session, the participants agreed that visual elements helped to improve the usability of the contract and to understand the information faster. The group working with the textual version also reported a higher average of the difficulty in finding and understanding the information.

Passera concludes that visualization increases the usability and user experience of contracts by providing faster and more accurate understanding of the content of the contract. Further, Passera states that the results should be indicative rather than conclusive, due to the small-scale experiment, but support previous research in the field of knowledge visualization.

Users of ERP systems are faced with a huge amount of displayed information that forms the basis for decisions made by the user. In the article “On the Visual Design of ERP Systems - the Role of Information Complexity, Presentation and Human Factors”, Mittelstädt, Brauner, Blum and Ziefle [18] outline characteristics and issues related to the visual design of ERP systems. Further, the authors address the importance of investigating the visual presentation of complex information with respect to human factor. Mittelstädt et al. have performed an empirical study on how the decision quality is affected by these factors.

According to Mittelstädt et al., the complexity of the data in ERP systems is often constituted by complex underlying business processes that cannot be manipulated by the software designer. Therefore, the visual representation of the data is important to support high usability for the end user. The authors claim that usability is a subordinated criterion for the design of information systems and that this along with the highly complex information is problematic. Further, Mittelstädt et al. argue that there is a close relation between subjective user satisfaction and objective performance, and includes the aspects of information complexity, visual representation and human factors for the decision quality in their study.

The findings of the study showed that a poor visual representation of data decreased the decision speed, especially for participants with lower perceptual speed. Further, the decision speed also was reduced with increasing data volume and complexity of the task. The authors also found that persons with high perceptual speed have the ability to compensate the effects of bad usability and could achieved about the same decision speed as persons with low perceptual speed using good visual data representations. Finally, Mittelstädt et al. suggest that developers of ERP systems should consider the diversity, such as age and cognitive abilities, of the end users to achieve higher effectiveness, efficiency and satisfaction of the system.

Parks has performed a usability study on two visual representations of the user interface in an ERP system, presented in the article “Testing & Quantifying ERP Usability” [19]. The low-
3.2. Decision-making Processes

Decision-making refers to the act of consciously selecting one alternative from a group of alternatives [21]. There are several decision theories, and these can broadly be divided into normative theories and descriptive theories. Normative decision models aim to describe a desirable form of rational decision-making, while descriptive decision models intend to explain the actual behavior of people making decisions [22].
3.3 Business Intelligence

Bell, Raiffa and Tversky [22] suggest a third grouping of decision theories called the **prescriptive** theory. Prescriptive theories are evaluated on the ability to help people make better decisions.

### 3.2.1 Behavioral Decision-making

According to Takemura, author of the book _Behavioral Decision Theory: Psychological and Mathematical Descriptions of Human Choice Behavior_ [21], behavioral decision-making can generally be divided into three subgroups based on the knowledge of the environment of the decision. These subgroups are: decision-making under certainty, decision-making under risk and decision-making under uncertainty. The first subgroup describes decisions where the outcome of all alternatives is certainly identified, the second explains decisions where the probability of the outcome is well-known and the last describes decisions where the probability of the outcome is unknown.

### 3.2.2 Managerial Decision-making

In organizations, decisions are made at strategic, tactical and operational levels. Strategic decisions aim to fulfill the long-term goals of the company and to establish company policies. To achieve the established policies, tactical decisions that focus on specific actions are made, along with operational decisions for the day-to-day functions [23]. Most commonly, several employees of different positions in a company are involved in decision-making processes on a daily basis.

Making fast, effective and correct decisions are essential for organizations. The primary goal of the management of a company is to maximize the profit, according to the Theory of the Firm [24]. In the article "Heuristics and Biases in Data-Based DecisionMaking: Effects of Experience, Training, and Graphical Data Displays Managerial", Hutchinson, Alba and Einstein [25] investigates how biases affect data-based decisions. According to the authors, managers generally base their decision on two types of information, their beliefs about what is true about the market and the data-based information gained from economical systems within the organization. The belief-based information is founded on several sources such as personal experiences, formal education and strategies of the company. Hutchinson, Alba and Eisenstein demonstrate that data-based inferences tend to be strongly biased by the cognitive heuristic used to analyze the data [25]. Furthermore, Biyalogorsky, Boulding and Staelin, state that data-based information is shown to be significantly biased by the beliefs of the manager when data-based information reveals new negative information. This gives the result that the data-based information is ignored or interpreted as more positive to conform to the beliefs of the manager and may result in bad decisions being made [26].

### 3.3 Business Intelligence

Business Intelligence (BI) is a collective term for several tools, methodologies and architectures that aims to collect, manipulate and analyze data to increase the ability to make more accurate decisions and is extensively described in the book _Decision Support Systems and Intelligent Systems_ by Turban, Aronson and Liang [9]. The authors states that BI is a broad term, meaning different things to different people, but in general the term includes the process of transforming data into information, decisions and actions.

The high-level architecture of a BI-system includes four main layers: data source system, data integration, data warehouse and user interface [9], [27], [28]. Data in the **data source system** can be sourced from multiple independent systems, such as back office systems, cloud applications, enterprise applications, databases and spreadsheets [27], [28]. Kimball and Ross provide guidelines for dimensional modeling for the data warehouse in the book _The Data
3.4 Data Analytics

According to Sharda, Asamoha and Ponna, data analytics has several definitions, but can be described as “the process of developing actionable decisions or recommendations for actions based upon insights generated from historical data” [31]. Abbot has a similar description and describes that data analytics has the goal to gain insight by using computational methods to discover influential patterns in historical data to affect decisions [32]. The Institute for Operations Research and Management Science suggests three levels of analytics: descriptive, predictive and prescriptive [33].

Sharda, Asamoha and Ponna describes that descriptive analytics involves consolidation of all available data sources to know what is happening in an organization and to understand the underlying trends and causes. Further, reports, queries, alerts and trends are developed from the data and visualization is a key tool to gain insight of the operations in an organization [31]. According to Ouahilal et al., descriptive analytics is usually associated with business intelligence [34].

Predictive analytics is data-driven and the process of finding patterns from the data is automated by predictive analytics algorithms [32]. This means that the models are based on key characteristics found in the data itself, and not on assumptions made by an analyst. Algorithms are used to identify which values of variables to use and to discover the form of the model. The power of predictive analytics is the ability to examine all potential combinations of inputs to identify the most interesting for the analysts to focus on [32]. Thus, the main goals of predictive analytics are to identify which inputs that are contributing to the patterns in the data [32], and to predict what is likely to happen in the future [31]. Predictive analytics is based on statistical methods as well as computational techniques such as data
3.5 Information Visualization

Information visualization is used to make huge amounts of data more understandable for the human eye. This enables fast and efficient recognition, classification and organization of abstract information \[35\]. Further, information visualization helps the human cognition to interpret and analyze complex data to enhance decision-making. Common techniques for information visualization are graphs, tables and diagrams.

Tory and Möller have performed an extensive review on research within the field of human factors in visualization, presented in the article "Human Factors in Visualization Research" \[36\]. The authors argue that visualization is a powerful tool for data analysis and decision-making, if the visualization tool is designed with regards to the human factors. Tory and Möller also claim that many areas of human factors-based design for visualization need to be further investigated and are asking for a new methodology developed for visual data presentation.

A study performed by Elting, Martin, Cantor and Rubenstein, showed that the type of data visualization used affected the accuracy of the decisions. In the study, presented in the article "Influence of Data Display Formats on Physician Investigators’ Decisions to Stop Clinical Trials: Prospective Trial with Repeated Measures" \[37\], four different types of visualizations constructed from the same data were used. All participants viewed each visualization separately and were asked to make one decision out of three alternatives. The visualization types used to present the data were: a table, a stacked bar graph, a pie chart and an icon display. The outcome of the study showed that there was no significant difference between the visualizations in time spent of making the decision. Further, the icon display resulted in the highest ratio of accurate decisions but was not preferred by any of the participants. The table was the most preferred but gave a lower rate of accurate decisions compared to the icon display. One of the conclusions made by the Elting et al. was that the choice of the visualization type significantly influences the accuracy of decisions.

3.5.1 Graph Displays

Quispel, Maes and Schilperoord have investigated the relationships between the familiarity, the attractiveness and the ease of use of graphs and charts among experts and laymen in design \[38\]. The authors designed 12 different static graphs (line graphs excluded) that can be used for representing quantitative and nominal data, with identical color set and the same proportions. The pilot study counted the distribution of the frequency of each graph type in everyday mass media to mirror the level of exposure to people for each graph type. The pilot study revealed that bar graphs were dominating with 57\%, followed by pie charts 16\% and divided bars 10\%. The evaluation study is presented in the article "Graph and Chart Aesthetics for Experts and Laymen in Design: The Role of Familiarity and Perceived Ease of Use" \[38\] and was performed to map the relationship between the perceived familiarity, attractiveness and ease of use of the twelve graphs. Both laymen and experts ranked the bar graph as the most familiar and with the highest ease of use. Laymen also ranked the bar graph as the most attractive, while the experts found the doughnut bar as the most attractive. In summary, the laymen and the experts judged the familiarity and the ease of use of the graphs similarly. Also, the familiarity and perceived ease of use were positively correlated for both groups. Finally, the authors argue that the attractiveness does not affect the perceived ease of use. Therefore, the
authors suggest that design experts should be well aware about the differences between their own ideas of attractiveness and understandability and the ideas of their audience.

In the book "Graph Design for Eye and Mind", Kosslyn [39] provides guidelines on how to choose the format of the graphs and claims that the format of the graph is important for the usability. The author claims that several studies have been conducted to compare the efficacies of tables and graphs of numbers. According to Kosslyn, the general finding is that graphs are better than tables for complex comparisons while tables are better to communicate specific amounts. According to Kosslyn, previous studies also indicate that graphic displays resulted in better performance in decision-making than tables.

Kosslyn [39] suggests the following guidelines:

- **Graphs for Percentage and Proportion Data:**
  - **Pie Graph:** Use a pie graph to effectively display general information about proportions, but without any specific details
  - **Divided-Bar Graph:** Use a divided-bar graph to display accurate impressions of the amount of a whole, with specific details

- **Graphs for Quantitative and Rank-Order Data:**
  - **Line Graph:** Use a line graph if the x-axis has an interval scale, if interactions over two levels on the x-axis should be displayed or if it can define a meaningful pattern
  - **Bar Graph:** Use a bar graph if the reader should compare specific measurements or if the scale on the x-axis is not continuous. A horizontal bar graph should be used when the labels are long but if there is a doubt, a vertical bar graph should be used
  - **Side-by-Side Graph:** Use a side-by-side graph to illustrate contrasting trends between levels of independent variables and if comparison between individual pairs are important
  - **Step Graph:** Use a step graph if the reader is supposed to notice relative changes among more than three variables on the x-axis
  - **Scatter Plot:** Use a scatter plot to convey an overall impression of trends and relations between two variables

- **Graphs for Cumulative Totals:**
  - **Stacked Bar Graph:** Use a stacked bar graph to display proportions of a whole that add up to a level on a nominal scale
  - **Layer Graph:** Use a layer graph when the x-axis is an interval scale and to illustrate the changes of different parts of the total

### 3.5.2 Design Principles

In the article "Cognitive Engineering, Cognitive Augmentation, and Information Display", Patterson [40] suggests eight principles for design of information visualization based on knowledge of the human cognition. The principles are developed from the concept of the dual-system theory of human cognition and focus on the knowledge of working memory and pattern recognition as key aspects for analytic reasoning and implicit decision-making, respectively.

**Principle #1:** The performance of a given cognitive task will potentially be degraded if the displayed information divides the attention of the observer. The reason for this is described as an increased cognitive load on the working memory. [40]
**Principle #2:** The performance of a given cognitive task will potentially be improved if the visualization continuously repeats and refreshes the same information. The visual short-term memory is limited and decays after ten seconds, but can be reactivated by rehearsal of the given information. [40]

**Principle #3:** The performance of a given task will potentially be increased if the displayed information helps the viewer in mentally chunking the information by its meaning and presents the information as retrieval cues for long-term memory. Chunking means recording items of low information content into a smaller number of items of high information content. Chunking increases the amount of information that can be processed and remembered. Long-term memory representations are triggered by cues contained in the pattern of stimulation and these representations become a part of the working memory. The result of chunking and retrieval cues is that the cognitive load on the working memory is reduced and therefore, the performance of the cognitive task is potentially increased. [40]

**Principle #4:** Blindness of inattention can be reduced if the displayed information presents images with direct attention to important visual stimuli. In other words, it is essential to direct the attention to the important visual stimuli to help the viewer to focus on the right elements. There is a risk that the stimulus is not perceived at all for information with many elements of potential attention direction. [40]

**Principle #5:** The performance of a given task may be enhanced if the displayed information helps the viewer to minimize attentional distractions. Further, the displayed information should help the viewer to maintain the information in the working memory to give the viewer the ability to remember the information. [40]

**Principle #6:** The performance of a given task will potentially be increased if the displayed information presents cues of singletons that direct the attention to the important information. A singleton is a visual attribute, that is strongly distinguishable from its background, and serve as a trigger stimulus to capture the attention of the viewer. [40]

**Principle #7:** Conclusions made from evidence and reasoning to solve a given problem can be improved if the displayed information helps the viewer to make a mental connection between a correlation and a target. This principle is based on analogical reasoning which is a fundamental thinking skill processed by humans. Analogical reasoning can briefly be described as making correlations between the similarities of relationships between elements in two or more dimensions. [40]

**Principle #8:** Implicit learning pattern-recognition-based decision-making may be encouraged by information statistically exposed over time. Implicit learning contains the process of unintentionally learning statistical regularities without the ability to verbally repeat the information. It improves the ability to make better intuitive decisions and may, in interplay with explicit knowledge, contribute to expertise in a given domain. [40]

In the article “A Human Cognition Framework” Patterson et al. [12] identify six leverage points to offer a broader framework for the information-visualization-design process. These leverage points aim to help visualization designers to influence certain aspects of the human cognition and are closely related to the eight principles described above. Patterson et al. claim that their framework should promote high-level cognitive functions such as reasoning and understanding. Further, the authors mean that well-designed visualizations should help the viewer to focus on important information, reduce distractions, promote chunking and encourage implicit learning. However, it is not possible to provide specific visual attributes that result in good visualization design to promote cognitive functionality, since good design is content specific.
3.6 Usability

The term usability is widely used in software engineering, however, there is no precise definition of the term that are generally accepted or used in practice. Alonso-Ríos, Vázquez-García, Mosquee-Ren and Moret-Bonillo, outline that several definitions have been suggested, but the definitions are often vague, brief and ambiguous [41].

Nielsen states, in the article "Usability Metrics: Tracking Interface Improvements", that “It is only meaningful to speak of usability relative to certain types of users performing certain categories of tasks” [42]. According to Nielsen, a certain program may be rated highly usable by one group of users, but highly unusable by another group of users, depending on what kind of task the users wish to perform.

3.6.1 Definition

The International Organization for Standardization (ISO) defines usability as the “degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [43]. Further, ISO defines effectiveness as “accuracy and completeness with which users achieve specified goals” [43], i.e. effectiveness means that the user can do what the user wants to do in the system. Efficiency is defined as “resources expended in relation to the accuracy and completeness with which users achieve goals” and satisfaction is described as the “degree to which user needs are satisfied when a product or system is used in a specified context of use” [43].

Alonso-Ríos et al. propose a detailed taxonomy, presented in the article “Usability: A Critical Analysis and a Taxonomy” [41], of usability that can be used to support the development process of usable software. To create the taxonomy, the authors performed a study to gather existing definitions and classifications. Some of the existing attributes were used, some were modified and some new attributes were added. Finally, the attributes were defined in detail and structured in different taxonomic levels. The result of the first level taxonomy of usability was six attributes: knowability, operability, efficiency, robustness, safety and subjective satisfaction. The first level attributes, see Table 1, aim to be generic and were further developed into sub-attributes that construct subsequent taxonomy levels.

Alonso-Ríos et al. states that the five attributes for usability, suggested by Nielsen, are considered to be widely accepted by some researchers [41]. The attributes proposed by Nielsen [42] are: learnability, efficiency, memorability, errors and satisfaction. Learnability, according to Nielsen, means that the system should be easy to learn and that the user should be able to begin the work quickly. Efficiency refers to enabling high productivity when the user has learned the system. Memorability indicates the ability of the user to remember how the system works, even after not using the system for a time. Errors mean that the system should contain few errors and that it should be easy for the user to recover from an error encountered. Finally, satisfaction implies that the user should find the system pleasant to use.

3.7 Design for Usability

It is impossible to create a complete list of design principles that needs to be applied to a software to be usable, since the usability of an application is highly affected by the context in which the application is used [12], [44]. Although, several researchers have suggested different rules and guidelines to support the usability of a system.

3.7.1 Eight Golden Rules

In the article “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, Shneiderman and Plaisant [44] describe eight golden rules that can be useful for the
3.7. Design for Usability

Table 1: Definitions of the usability attributes of the first level taxonomy proposed by Alonso-Ríos et al. Adapted from "Usability: A Critical Analysis and a Taxonomy" [41]

<table>
<thead>
<tr>
<th>USABILITY ATTRIBUTE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowability</td>
<td>&quot;the property by means of which the user can understand, learn, and remember how to use the system&quot;</td>
</tr>
<tr>
<td>Operability</td>
<td>&quot;the capacity of the system to provide, users with the necessary functionalities and to permit users with different needs to adapt and use the system&quot;</td>
</tr>
<tr>
<td>Efficiency</td>
<td>&quot;the capacity of the system to produce appropriate, results in return for the resources that are invested&quot;</td>
</tr>
<tr>
<td>Robustness</td>
<td>&quot;the capacity of the system to resist error and adverse situations&quot;</td>
</tr>
<tr>
<td>Safety</td>
<td>&quot;the capacity to avoid risk and damage derived, from the use of the system&quot;</td>
</tr>
<tr>
<td>Subjective satisfaction</td>
<td>&quot;the capacity of the system to produce feelings of pleasure and interest in users&quot;</td>
</tr>
</tbody>
</table>

design of human-interactive interfaces and have been derived from experience over two decades. The design principles are presented below.

**Principle#1: Strive for consistency**, refers to numerous forms of consistency and is the most frequently violated rule. The main message is that similar layout and sequences of actions should be used throughout. For example, identical terminology should be used in menus, help screens and prompts, and the color, font and capitalization should be consistent. High consistency in a system encourages learnability and lets the user predict how the system works which reduce the training needed to sufficiently use the system. [44]

**Principle#2: Carter to universal usability**, involves designing the user interface to satisfy the needs of diverse users. Differences in knowledge, age and experience of technology should be considered for the design requirements. Examples of features that can enrich the user interface design and improve the perceived quality are explanations for novices and shortcuts for experts. Design for a system that is flexible and can easily be shaped improves the efficiency of the tasks performed. [44]

**Principle#3: Offer informative feedback**, means that the system should provide feedback of all actions performed by the user. Minor and frequent actions can show modest response while major actions require more substantial feedback. [44]

**Principle#4: Design dialogs to yield closure**, which gives the user feedback of accomplishment. Sequences of actions should have a beginning, a middle and an end that gives the user information about the process and let the user know when the task is completed. [44]

**Principle#5: Prevent errors**, includes designing the interface to make it as hard as possible for the user to make major errors. The interface should prevent possible errors and reduce the rework needed from the user. As an example, inputs of invalid characters should be detected immediately and guide the user to enter valid information. [44]
3.8 Usability Evaluation

Principle#6: Permit easy reversal of actions, that lets the user know that an error made by the user can be undone. This encourages exploration of unknown options and reviles the anxiety of the user. [44]

Principle#7: Support internal locus of control, since the user desire to be in charge of the actions. Difficulties to obtain necessary information, surprisingly actions of the user interface or unclear responses reduce the satisfaction of the user. [44]

Principle#8: Reduce short-term memory load, requires the elements of the user interface to be simple. Actions where the user needs to alter between pages or scroll to remember information should be avoided. [44]

3.7.2 Human-centered Design Principles

ISO [45], [46] suggests a number of design principles for human-centered interactive systems. The design principles aim to facilitate the effectiveness, efficiency and satisfaction of the system to achieve a good user experience and are listed below:

1. **Suitability for the task:** A dialogue should support the user to effective and efficiently complete a task. [45], [46]

2. **Self-descriptiveness:** It should be obvious for the users which dialogue they are in, where they are within the dialog and which actions that can be performed. [45], [46]

3. **Conformity with user expectations:** A dialogue should correspond to commonly accepted conventions and the contextual needs of the user. [45], [46]

4. **Suitability for learning:** A dialogue should support and guide learning of the system. [45], [46]

5. **Controllability:** The user should be able to control the interaction with the dialogue until the task is completed. [45], [46]

6. **Error tolerance:** A user should not be required to redo actions if an invalid input is entered. [45], [46]

7. **Suitability for individualization:** Users should be able to customize the information to suit their individual needs. [45], [46]

3.8 Usability Evaluation

There are a wide range of techniques and methods used to evaluate usability of systems [47]. Ghasemifard, Shamsi, Kenari and Ahmadi have analyzed the benefits and drawbacks of several usability evaluation methods in the article “A New View at Usability Test Methods of Interfaces for Human Computer Interaction” [48]. The conclusion from the analysis was that each method has unique advantages and that no method is superior over others. Some examples of methods for usability evaluation are: heuristic evaluation, cognitive walk-through, user testing, questionnaires and prototype evaluation [48], [47], [49].

Qualitative usability evaluation are often performed with a low number of participants and used to identify design problems of the user interface, whereas quantitative methods are used to evaluate the usability through different metrics with a higher number of participants [48], [50]. Further, quantitative usability methods can be used for benchmarking and a correctly reported quantitative usability test will include information about the statistical significance of the results to support the reliability of the study [50].
3.8. Usability Evaluation

According to a mapping study performed by Paz and Pow-Sang in 2015 [47], presented in the article “Usability Evaluation Methods for Software Development: A Systematic Mapping Review”, the top three most used methods for usability evaluation are questionnaires, user testing and heuristic evaluation. In the study, 228 journal articles and conference proceedings were selected. The articles were reviewed and the methods used for usability evaluation were collected. Paz and Pow-Sang noted that several methods were often combined to cover all aspects of usability.

In the article “Criteria for Evaluating Usability Evaluation Methods”, Hartson, Andre and Willige [51] discuss comparison criteria and performance measurements useful in evaluation of usability evaluation methods. The authors claim that there is a general lack of understanding of the benefits and drawbacks of each usability evaluation method, and that practitioners need better knowledge about the effectiveness and applicability of the different methods. Moreover, there are no standard criteria for comparing usability evaluation methods, which makes a reliable evaluation difficult to perform. Hartson, Andre and Willige argue that one of the challenges with evaluation of usability evaluation methods is that the methods themselves are not stable and continue to change.

3.8.1 Questionnaires

Standardized questionnaires for usability are often used together with user tests to measure the overall satisfaction of the system under test [52]. The purpose of the questionnaires is to get a quick overview of the usability of a system [53]. There are a number of standardized questionnaires that can be used, each including three to fifty questions. Examples of these are: After Scenario Questionnaire (ASQ), Computer System Usability (CSUQ), System Usability Scale (SUS), Usability Magnitude Estimation (UME) and Software Usability Measurement Inventory (SUMI) [52].

Bangor, Kortum and Miller have performed an empirical evaluation of the SUS metric and claims that no usability metric should be used in isolation to determine the absolute usability of an application [52]. Although, the SUS metric does provide a good overall estimation of the usability of a system according to Bangor, Kortum and Miller. In the article “The System Usability Scale: Past, Present, and Future”, Lewis [53] has investigated the use of the system usability scale from its early history, in the 1980, until the present. Lewis claims that the SUS questionnaire is the most widely used and is likely to remain so in the future. Furthermore, Lewis states that the magnitude of the SUS means should be in comparison with norm values, to assess the perceived usability of a system. The Sauro-Lewis Curved Grading Scale provides general interpretation for the SUS means, where 68 was the mean of the systems used to create the scale [53]. Similar to this, Bangor, Kortum and Miller argue that an acceptable system has a SUS score of at least 70, better products score between 70 and 80, and superior products have a SUS score above 90 [52].

3.8.2 User Testing

According to Matera, Rizzo and Carughi [54], user testing involves the participation of a representative sample of end users and typically the participants of a user test are provided with a number of tasks to perform in the system to be evaluated. Further, the behavior of the participants is observed and difficulties and errors encountered by the users are recorded to identify design flaws. The body language and comments by the participant are also observed and recorded to get a broader view of the perceived usability of the system.

Ritter, Baxter and Churchill [49] describe two types of user-based testing: formative and summative. Formative evaluation is used to identify design problems and potential solutions and can be used in any time of the development process. The aim is to reduce problems that occur when using the system and to improve the usability of the system iteratively, starting from
an early stage in the design process. **Summative evaluation** aims to assess the success of the final system and summarize different aspects of the overall usability of the systems.

Tullis and Albert describes different methods of measuring the usability in the book *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. One metric mentioned is the task completion rate, or task success, which is a quantitative metric to evaluate the effectiveness of a system. The metric is measured by the number of completed tasks divided by the number of undertaken tasks through user testing, see equation (1). According to the authors, the participants of the test are given a set of tasks to complete and the number of completed tasks are measured. To be able to measure the success each task needs to have a predefined success criterion and the answers to each task can be collected by letting the user answer out loud, write down the answer or by using multiple-choice questions.

\[
\text{Effectiveness} = \frac{\text{Number of tasks completed successfully}}{\text{Total number of tasks undertaken}} \times 100 \tag{1}
\]

According to Tullis and Alber, the task success can be measured binary or by task levels. Binary success is the simplest and most commonly used method, but task levels might be a good choice if the tasks are not explicitly right or wrong, according to the authors. In binary success, each task is assigned 0 if failed and 1 if succeeded and, most commonly the average success per task is presented. Tullis and Albert mean that this allows for deeper analysis of the causes of the tasks with low success rate. The approach for measuring task level success is similar to the binary approach but contains more levels than fail and success. For example, a task level test that evaluates the effectiveness of a user interface can include the following levels:

- 1 = No problem. The task was completed successfully without any difficulties
- 2 = Minor problem. The task was completed successfully, but with some small mistakes along the way
- 3 = Major problem. The task was completed successfully, but with major problems along the way
- 4 = Failure/gave up. The wrong answer was provided or the user gave up

The data from a task level evaluation should, according to Tullis and Alber be represented as the frequency of each level of completion per task and the tasks with high frequency of 3s and 4s should be subjects for design improvements. Another metric that can be used to evaluate the usability in terms of effectiveness is the number of errors made by the user during a task completion.

Kortum and Peres have examined the relationship between the subjective usability assessment measured by the SUS and the effectiveness of a system, measured by the task success rate. The study is presented in the article "The Relationship Between System Effectiveness and Subjective Usability Scores Using the System Usability Scale". Kortum and Peres claim that there, in general, is a strong and reliable positive correlation between the SUS score and the effectiveness of a system. Although, the authors claim that a subjective usability evaluation method, as the SUS, does not eliminate the need of evaluate the usability with other methods. The average task completion rate is 78%, according to a study of data from 115 usability tests including 1189 tasks, carried out by Sauro.

According to the article "Writing Tasks for Quantitative and Qualitative Usability Studies" by Meyer, the tasks for a user tests should be created differently depending on whether the usability study is qualitative or quantitative. Writing good tasks are essential for the success...
3.8. Usability Evaluation

of the usability study and Meyer suggests a number of guidelines that can be applied for both qualitative and quantitative studies [59]:

**All Tasks:**
- Understand what the user needs to do with the system
- Avoid giving clues in the task
- Keep the task emotionally neutral
- Pilot test the tasks

Further, Meyer claims that a good task for a qualitative study is exploratory and may be open-ended, while a good quantitative task is concrete and focused [59]. Additional guidelines for qualitative and quantitative tasks are presented in the lists below:

**Qualitative Tasks:**
- Provide enough information to establish the motivation to perform the task
- It is okay to add tasks or change a task if it is not providing insight

**Quantitative Tasks:**
- Make sure the task can only be performed in one way
- Provide details to keep the task narrow and focused
- Provide fake credentials for personal information
- Each task should stand alone
- Each task should have a single success criterion
- Do not change the tasks after the first user test
- Focus on the core tasks

A good user test to evaluate usability should, according to Matera, Rizzo and Carughi, involve the following steps [54]:

1. **Define the goals of the test:** The objectives of a test can be either generic or specific. A generic test can be an evaluation of the satisfaction or the design, while a specific test can be to evaluate the understandability of a certain element. [54]

2. **Define the user sample to participate in the test:** The sample should be representative of the end users and examples of criteria to use are work experience, age and technical experience of similar applications. [54]

3. **Select tasks and scenarios:** The tasks chosen for the test need to be real and represent tasks normally performed within the application. [54]

4. **Define how to measure usability:** It is important to define the measurements used to evaluate the usability of the test. The measurements can be qualitative (e.g. satisfaction) or quantitative (e.g. time to complete a task). During the test, general observations can be recorded by the test leader or a think-aloud protocol can be used. The think-aloud method asks the user to speak out their thoughts during the test. The user test can be completed with a questionnaire to be filled out by the tester after the session to capture subjective measurements, such as satisfaction. [54]
5. **Prepare the material and the experimental environment**: The test environment should be equipped with a computer and a video camera to record the behavior of the participant. A pilot trial should be conducted to test and possibly refine the test procedure.

According to Nielsen [60], three to five test users are enough to maximize the cost-benefit ratio of the test. Adding more testers will only result in identifications of the same issues repeatedly rather than learning new things. After one test with five testers about 80% of the usability problems are found. Nielsen also claims that the overall improvement of the user experience is higher if three user tests are conducted with five testers compared to one test conducted with 15 testers.

Although five participants are widely assumed to be enough for usability testing, there are several opponents against this number. In a study described in the article “Beyond the Five-user Assumption: Benefits of Increased Sample Sizes in Usability Testing”, Faulkner [61] showed that the amount of reported usability problems differed significantly between different groups of five randomly selected participants. The group that reported the least number of usability problems only identified 55% of the problems, while the group with the highest rate identified 90% of the problems. The same study also indicated that when the number of participants in each group was increased to 10, the minimum rate of found problems was 80% and with 20 participants the minimum rate had increased to 95%. Further, Faulkner claims that the number of test participants required to get a valid and accurate result depends on several aspects, such as the experience of the test users available and the criticality of the tasks of the system. The most critical factor in reliable usability testing is that the group of user testers needs to be representative of the target users [61].

Woolrych and Cockton [62] argue that a safe method to estimate the number of test users needed must take severity and frequency data into consideration and should not only rely on statistics on how many usability problems that are likely to be found. Alroobaea and Mayhew [63] state that $16 \pm 4$ testers are valid to discover 90% of the usability problems when performing user testing and argue that a group of five testers are not enough to find an average of 80% of the usability problems [63], as claimed by Nielsen [60].

A quantitative usability test, performed to make a statistical analysis requires at least 30 test users, according to Budiu [50].

### 3.8.3 Heuristic Evaluation

Heuristic Evaluation lets usability experts investigate the user interface to evaluate what might be potential problems for the users [47]. There are several approaches for Heuristic Evaluation and may vary according to the system to be evaluated, according to Matera, Rizzo and Carughi. A list of heuristic rules can be used by the evaluator to identify violations against these rules while the screens of the user interface are observed. Subjective judgment is a method relying on the subjective judgments and experience of the evaluator and if the system is operational, task-based evaluation can be performed by the evaluator. This method has similarities to user testing, despite that the participant is a usability specialist and records the encountered problems [54].

### 3.8.4 Evaluation at Different Stages of the Process

In agile and iterative development, an evaluation should be carried out in each iteration according to Ritter, Baxter and Churchill [49]. They mean that early feedback increases the chance of developing a usable product, that satisfies the needs of the users. The life cycle of a development process often includes one or several prototypes before the final system is finished [46] and depending on the stage of the development process, different methods for
3.8. Usability Evaluation

the usability evaluation are appropriate to use. According to ISO, user based testing can be undertaken in any stage of the design process. According to Ritter, Baxter and Churchill, low-fidelity prototypes of the core concepts and main ideas are usually sketched with paper and pencil in the early stage of the design process. This is a cheap way to visualize the ideas and the sketches can be shown to users to be evaluated. The models and concepts should be evaluated in relation to the real world context. According to Ritter, Baxter and Churchill, the evaluation is usually formative and the data collected are qualitative. Different ideas are still being explored and the evaluation aims to inform the design of the system. Further, Ritter, Baxter and Churchill mean that the ideal is to use real users in this evaluation, but other UX-designers or potential users of other systems can also be used.

The next step of the development process includes the creation of a computer based prototype. The fidelity of the prototype is increased during the process, from low to high. In the book "Foundations for Designing User-centered Systems: What System Designers Need to Know About People" Ritter, Baxter and Churchill present two basic types of approaches to use for prototyping: evolutionary and revolutionary. Evolutionary prototypes are typically refined after each iteration and evolve towards the final system. Revolutionary prototypes are discarded after each iteration and a new prototype is developed. Further, Ritter, Baxter and Churchill claim that computer based prototypes are suitable for many stages of the development process and also can help to identify new requirements.

Knapp, Zeratsky and Kowitz, authors of the book "SPRINT How to Solve Big Problems and Test New Ideas in Just Five Days", suggest that the prototypes are evaluated through one-on-one interviews that provides instant feedback on the prototype and lets the interviewer understand why certain parts are not working for the user. According to the authors, the interview should include a friendly welcome, a presentation of the prototype and some open-ended questions. The recommendation is to use “Five Ws and One H”-questions, i.e. “Who…”, “What…”, “Where…”, “When…”, “Why…” and “How”. If more than one prototype is evaluated, the user should be asked to compare the different solutions and describe what worked and what did not.

3.8.5 Validity & Reliability

Ritter, Baxter and Churchill also discuss the terms of validity and reliability of usability studies. Irrespective of whether qualitative or quantitative data are collected during the usability tests, the validity, reliability and sensibility of the evaluation need to be considered. Validity, reliability and sensibility aim to ensure that the evaluation measures what is supposed to be measured, that the effects are measurable and that the results can be generalized.

According to Ritter, Baxter and Churchill there are different types of Validity that needs to be considered in an evaluation. The different types can be divided into two groups, and each type is listed and described below:

1. Instrument validity: refers to the instruments or measures used in the study.
   - Construct validity: refers to the extent that the measures used in the study actually measure what are supposed to be measured. The total usability of an application cannot be measured by a single measure alone, since the different dimensions (efficiency, effectiveness and satisfaction) need to be measured separately.
   - Content validity: refers to whether the content of the measure corresponds to the content of the artifact it was designed to measure. For usability surveys, this includes making sure that all relevant aspects of the usability of the artifact have been covered by the survey, through a systematic review.
• **Face validity (or surface validity):** is closely related to content validity, and refers to whether the test appears to measure a certain criterion. The difference from content validity is that people make judgments about what they think the test is measuring, rather than performing a systematic review. [49]

2. **Experimental validity:** refers to the generalizability of the results [49]

• **Internal validity:** refers to the ability to draw correct conclusions about the casual effect relationships based on the design of the study. This includes the measurements used and the situation surrounding the study. High internal validity is in general found in controlled studies and needs to consider the maturation and the types of the participants, as well as effects caused by performing the same test a second time on the same participants. [49]

• **External validity:** refers to at which extent the results of the study can be generalized to different user groups or other populations, different places and other times. To achieve high external validity, there must be an awareness of the effects that can have an impact on the results and mitigate these effects. Some examples of effects that might have an impact on the results are: choice of participants, anticipated results, the behavioral changes of the users as an impact by being observed, and the motivation of the participants. [49]

• **Ecological validity:** refers to at which extent the results of the study can be applied to the real world. To achieve high ecological validity, the situation being investigated needs to correspond to real-world situations. In a usability test of a system that will be used in an office environment, the test environment should include constant sources of interruptions, such as phone calls and conversations. [49]

Moreover, Ritter, Baxter and Churchill [49] states that there are several tradeoffs that may need to be considered when deciding which type of validity that is important for the study. For example, the tester may respond more honestly if the test has a low face validity since the focus is on the task rather than the measure. Also, face validity should not be relied on alone, since the judgement of people might be wrong. Further, it is possible to have low face validity, but high construct validity and therefor this tradeoff should be considered when designing the evaluation [49].

According to Ritter, Baxter and Churchill [49], there might also be a conflict between high internal validity and high external validity, and ecological validity. If the usability test is performed in a laboratory environment, Ritter, Baxter and Churchill argue that the internal validity is high since all the interfering variables are under control, but the external validity and the ecological validity are reduced. Since an artificial context is used for the test, the results may not be generalizable to the real world. If the test is performed in a real-world environment, the external validity and the ecological validity will be high, but the internal validity will be reduced. This might be a problem depending on the research strategy used in the study according to Ritter, Baxter and Churchill.

Further, reliability is concerned with the ability of a measure to give consistent results when the same measure is performed again under different conditions [49], [65]. For example, a reliable test should give the same results if the same test is conducted over again, but in a different day or with a different test-group of similar participants.

### 3.9 Software Development Methodologies

There are numerous software development methodologies aiming to provide a framework for planning, executing and managing the development process of software systems. The
methodologies may be grouped into plan-driven methods, agile methods or hybrid methods, depending on the development process approach [66].

In a survey conducted in 2013, the use of agile development methods was reported by 43% of the respondents and the waterfall model (plan-driven) were used by 22% of the respondents [67]. Another study, performed by Vijayasarathy and Butler [66] in 2016, showed that the top four most used methodologies in software development projects were, in descending order: Waterfall, Agile Unified Process, Scrum and Test-driven development. The same study also indicated that hybrid and agile approaches were dominating if a categorization by hybrid, agile, iterative and traditional approaches were applied. Further, companies with a high number of employees tended to use traditional approaches, whilst organizations with a low number of employees used iterative or agile methodologies [66].

Nerur and Balijepally [68] describe an evolution of software development where the process adapts to a shifting in design philosophy. The environment has changed from stable to unpredictable and the goal of problem solving has altered from optimization to responsiveness. Moreover, the view of the problem nature has reformed from deterministic to more complex and the nature of learning has become more generative rather than adaptive. According to these changes, the problem-solving method has moved from linear to iterative. Nerur and Balijepally also claim that this evolution in software development has similarities to the progress of design ideas in architecture and strategic management.

3.9.1 Plan-driven Software Development

Plan-driven software development methodologies are based on a sequential process where each step is completely finished before moving on to the next step. One of the most known development methodologies is the Waterfall Model. The foundation of the Waterfall model was first described by Royce in the article “Managing the Development of Large Software Systems” [69]. The Waterfall model is still one of the most used methodologies in software development projects according to Nerur and Balijepally [68].

In the book Practical Software Development Techniques: Tools and Techniques for Building Enterprise Software Crookshanks [70] describes the structure of the Waterfall Model, see Fig. 2. The first step, containing the requirements, should include all desired functionality and need to be fully documented and approved before the design phase can be initiated. The design phase means that the system is designed to meet all defined requirements. First when the design phase is completed, the implementation phase is started. After the implementation, the software is tested by a testing team that verifies that all requirements are met and that the system behaves correctly. Finally, the product is delivered to the customer or released for production.

Stober and Hansmann [71] outline that the Waterfall model results in heavy documentation and often includes high costs of changes in requirements. Further, they mean that plan-driven software development methodologies are generally efficient in terms of carrying out projects, given that the plan is carefully followed. Although, projects most commonly implies several changes to fully satisfy the customer, and with a plan-driven approach these changes are identified late in the process and require the process to go back to step one [71].

3.9.2 Agile Software Development

To respond to the dynamically changes in software development environments and meet the inability to define final software requirements up front, methods to embrace late changes of software requirements started to be developed [72]. Agile Software Development (ASD) is a collective term for several software development methodologies based on iterative and incremental development [73]. The idea of ASD is to satisfy customers with early and frequent
3.10 Requirements Engineering

Dick, Hull and Jackson [76] describe that requirements are fundamental parts of every project and aim to identify the needs of the stakeholders and define what the system needs to accomplish to satisfy these needs. The specified requirements are the basis for planning the development of a system and for acceptance testing of the final system. Lack of complete requirements is one of the most common reasons for project failures, according to Dick, Hull and Jackson and their definition of requirements engineering presented below:

Requirements engineering: the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction. [76]

Zowghi and Coulin [77] describe that term requirements elicitation is used in requirements engineering for the practice of gathering requirements. The aim of requirements elicitation is to communicate the needs of the stakeholders to the system developers. This includes to learn and to understand the users, and to extract and uncover the wants of the project sponsors. Further, Zowghi and Coulin state that the process of elicitation and the context in which
the requirements are gathered are volatile by nature. Most commonly, the wants of the stakeholders are going to change during the project and the process of elicitation affects the quality of the requirements. For the process of requirements elicitation to be successful, several elicitation techniques need to be used in conjunction, according to Zowghi and Coulin.

In the book *Engineering and Managing Software Requirements*, Aurum and Wohlin [78] states that the requirements do not only reflect the needs of the customers, but also the needs arising from other stakeholders such as general organization, community, government and industry standards. Requirements can be classified in many ways according to literature, but the differences might not be as distinct in practice according to the authors. Common classifications of requirements are presented in Table 2.

Table 2: Classification of requirements. Adapted from “Engineering and Managing Software Requirements” [78]

<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td><em>what the system will do</em></td>
</tr>
<tr>
<td>Non-functional</td>
<td><em>constraints on the types of solutions that will meet the functional requirements e.g. accuracy, performance, security and modifiability</em></td>
</tr>
<tr>
<td>Goal level</td>
<td><em>related to business goals</em></td>
</tr>
<tr>
<td>Domain level</td>
<td><em>related to problem area</em></td>
</tr>
<tr>
<td>Product level</td>
<td><em>related to the product</em></td>
</tr>
<tr>
<td>Design level</td>
<td><em>what to build</em></td>
</tr>
<tr>
<td>Primary requirements</td>
<td><em>elicited from stakeholders</em></td>
</tr>
<tr>
<td>Derived requirements</td>
<td><em>derived from primary requirements</em></td>
</tr>
</tbody>
</table>

Denger and Olsson [79] emphasize that there is a gap in the understanding of how high-quality requirements can be achieved. They also claim that there is a lack of knowledge of how the costs of quality assurance in requirements engineering affect the costs of the overall development of a system.

A survey, conducted in 2013 and discussed by Kassab [67], on requirements engineering practices in the industry reported that the most used methods for requirements elicitation were, in descending order: interviews, user stories, prototyping and scenarios. The respondents were asked to report all methods used in their organization, i.e. several methods could be chosen by each participant. The same survey was also conducted in 2003 and 2008 and the results are analyzed by Kassab in the article “The Changing Landscape of Requirements Engineering Practices over the Past Decade” [67]. In 2003, the most used methods for gathering requirements were scenarios and focus groups. User stories were the least used method and Kassab reports that the overall satisfaction of the requirements engineering practices in general has increased from 67% in 2003 to 82% in 2013.

Saiedian and Dale [80] point out that the most important is not the methods chosen for requirements elicitation, but the understanding of the importance of participation of the end users in the process of defining requirements. Saiedian and Dale outline that better requirements and encourage of the relationship and communication between the developer and the users, are needed to help the users to understand the needs.
3.10.1 Interviews

According to both Kassab [67] and Wang, Zhao, Wang and Sun [81], interviews are the most frequently used method for requirements elicitation. Dick, Hull and Jackson [76] describes that interactive interviews with stakeholders are performed to gather requirements and understand the domain of the stakeholders. The interviews should be prepared in advance and it is important for the interviewer to take notes during the session to be able to process the documentation into requirements. The questions should cover both general and specific areas and it is essential to cover all aspects and to define what is irrelevant, according to Dick, Hull and Jackson.

Further, Dick, Hull and Jackson [76] claim that the results of interviews are highly affected by the skills of the interviewer and the format of the interview. In the article "Research Methods: Interviews", Wilson [82] describes three general types of interviews: unstructured, structured and semi-structured. Structured interviews consist of predefined questions and do not leave the topic. The same questions are asked for all participants of a study. Unstructured interviews are more like a conversation where the interviewer has an idea of what to explore but does not have a set of predefined questions. Semi-structured interviews are a combination of structured and unstructured. There are generally some predefined questions, but there is also room for discussions and conversation. According to Wilson, structured interviews are better suited for quantitative research, since the collected data can be compared, whilst semi-structured or unstructured interviews are more flexible and better suited for qualitative research.

During interviews, there is a risk that ambiguity can inhibit the communication between the stakeholder and the analysts performing the interview, according to Ferrari, Spoletini and Gnesi [83]. In the article "Ambiguity Cues in Requirements Elicitation Interviews" by Ferrari, Spoletini and Gnesi [83], a study of 34 customer-analyst interviews to identify terms of language that trigger ambiguity is presented. The identified term cues that seemed to trigger ambiguity in the study were: under-specified terms, vague terms, quantifiers, pronouns and domain-specific terms. The authors state that their findings require empirical validation but can help analysts to detect and be prepared for ambiguity.

3.10.2 User Stories

User stories are one of the most used methods for requirements engineering by business analysts and product owners [81], [67]. There are different definitions of a user story, but in general it describes a functionality that is valuable to the end user or the purchaser of the system and follows the template presented below [84].

As a <role>, I want <function/feature> so that <benefit> [84], [85]

The book Requirements Writing for System Engineering the basics principles of user stories are presented by Koelsch [85]. The role describes who is interacting with the system, the function/feature is the action and describes what is wanted from the system and the benefit describes why the feature is wanted. A good user story is small enough to be completed within one sprint and is independent, negotiable, valuable, estimable and testable, according to Koelsch. Independent means that the story should be able to be developed, tested and delivered in isolation. In agile development, a user story can be negotiated at any time and be redefined or updated to better suit the needs of the stakeholders. Further, a user story needs to be able to be estimated in terms of complexity and time for development. If a user story is too big to complete in one sprint, it should be broken down into smaller stories. Koelsch also outlines that a user story is not completed until it is explicitly tested successfully.
In the article “A Reference Method for User Story Requirements in Agile Systems Development” by Bik, Lucassen and Brinkkemper [84], a reference method, that can help organizations to integrate user stories more deeply in their development process, was constructed. The Reference Method for User Stories (RMUS) can, according to Bik, Lucassen and Brinkkemper, assist the organization to leverage user stories as effectively as possible. Through case studies of 16 real-world companies, the authors showed that the method could easily be adopted to existing processes.

The RMUS includes five phases: requirements gathering, user story creation, development, testing and deployment. The requirements gathering phase includes collecting requirements from various sources. The sources can be internal meetings, feedback from customers, bug reports, etc. The gathered requirements are filtered to correspond to the overall vision of the product and the requirements gathering phase result in the product backlog. The product backlog is a prioritized list of requirements based on the desired functionality of the system to be developed. The user story phase involves activities such as identifying epics (large user stories), splitting epics into two or more user stories, writing new user stories and refining and validating user stories. The user stories are estimated in the effort required, prioritized and lastly items from the product backlog are selected to create the sprint backlog. During the development phase, the items from the sprint backlog are developed into new software. Each user story tells the developer what to build, how much time is estimated for the implementation of the feature and the definition of done. Unit testing should continuously be written during the development phase and the release documentation is updated. During the testing phase functional, integration and acceptance tests are performed and all selected user stories for the sprint are verified. Finally, a new version of the developed application is delivered to the customer, according to Bik, Lucassen and Brinkkemper [84].

3.10.3 Prototyping

Prototyping is used in requirements engineering to construct a model of an executable system to understand the problem and to identify appropriate and feasible behavior to find solutions, according to Saiedian and Dale [80]. Feasibility analysis and identification of necessary and unnecessary requirements are in focus of the method. Saiedian and Dale outline that prototyping is often used for user interfaces to make a “quick and dirty” model.

According to Andriole [86], the prototyping phase of the requirements elicitation can follow the steps described below:

1. **Design the user interface:** Develop a rough outline of the user interface based on guesses and experience. [86]
2. **Build a “quick and dirty” system:** Build the prototype to let users assess the interface, but there is no need for reliability and error-checking. [86]
3. **Let users play with the prototype:** Do not spend too much time on polishing the prototype before the users get access to test it. [86]
4. **Observe the users:** Observe first reactions and how the users use the interface to accomplish tasks. [86]
5. **Collect user feedback:** Use both specific and open-ended questions and perform anonymous questionnaires as well as face to face interviews. [86]
6. **Back to the drawing board:** Reconsider the design of the interface according to the feedback from the users and start over at step 1. [86]
Zowghi and Coulin [77] state that prototyping is most commonly used in conjunction with other methods for requirements elicitation such as interviews. Examples of techniques that can be used for prototyping systems are storyboards, paper mock-ups and graphical mock-ups [87], and the prototypes are often based on preliminary requirements or existing applications [77]. The method of prototyping is useful for human-interaction systems, especially if the stakeholders are unfamiliar with possible solutions, according to Zowghi and Coulin.

In the article "Low vs. High-fidelity Prototyping Debate", Rudd, Stern and Isensee [88] states that there are in general two different types of prototypes: low-fidelity prototypes and high-fidelity prototypes. Low-fidelity prototypes mainly demonstrate the look of a user interface and are limited in functionality. The main purposes of low-fidelity prototypes are, according to Rudd, Stern and Isensee, to demonstrate the design and facilitate discussion of the concepts rather than modeling the interactions between the user and the system. The advantages outlined by the authors, are the lower costs and lower time consumption in the construction of the prototype, whilst the disadvantages are the limitation of details and error checking. High-fidelity prototypes are detailed, fully interactive and represents the core functionality of the user interface. The main advantage is, according to Rudd, Stern and Isensee, the ability to demonstrate a working product that can demonstrate both the look and the interactions. Contrary to the low-fidelity prototype, the high-fidelity prototype is both time consuming and more expensive to create. Memmel, Reiterer and Holzinger [87] claim that high-fidelity prototypes can be a partial substitute for textual design requirements and can act as a base for programming.

3.10.4 Scenarios

According to Zowghi and Coulin [77], a scenario describes the process of interactions between a user and a system. There is a specific and detailed description of the current or future actions involved in a process and scenarios are also useful for the development of test cases. Also, potential exceptions for each step of the process in a scenario is captured. There are several structured approaches for scenarios in requirements elicitation, such as the Cooperative Requirements Engineering with Scenarios, The Inquire Cycle, Scenario Based Requirements Elicitation and Scenario Plus [77].

Denger and Olsson [79] state that scenarios ensures that the right functional requirements are gathered from the start, since scenarios are easy to understand for both technical and non-technical stakeholders. Further, a scenario-oriented approach improves the verifiability of the requirements since the scenarios are valuable in the definition of acceptance tests, according to Denger and Olsson.

3.10.5 Sustainability

In the article "Framing Sustainability as a Property of Software Quality", Lago, Koçak, Crnkovic and Penzenstadler [89] describe sustainability as the capacity to preserve the functionality of a given system over an extended time. To analyze the sustainability of a system, the authors states that the requirements of the system can be divided into four dimensions: economic, social, environmental and technical. Often the social and the environmental dimensions are neglected. Lago et al. suggests a framework to help requirements engineers to consider the social and environmental dimensions in relation to technical and economic dimensions and how the trade-offs in each dimension affect each other. Social and environmental requirements are not customer-driven but should be considered to reflect the needs of the surrounding community and society at large, according to the authors.
3.11 Design Methods

Design is in general an iterative process where the knowledge of the problem and possible solutions increases with each cycle [90]. Design decisions have a major impact on the user experience and the usability of a system [46].

3.11.1 Human-centered Design Process

ISO [46] proposes a human-centered design process for the design of any interactive system. The process is iterative and contains four design activities, see Fig. 3.

The first activity, define the context of use, aims to gather information about users, other stakeholders and the organizational environment to define the context of the system to be developed. The description about the context of use should include characteristics, goals and tasks of the users and the environment of the system. The context of use should support the activities of requirements, design and evaluation. Specify user requirements is the activity of identifying the needs of the users and stakeholders and should include explicit statements related to the context of use. Producing design solutions should take the whole user experience into consideration and includes activities such as designing user tasks, interactions and user interface to meet the requirements. The design solution should follow the design principles described in section 3.7.2 Human-centered Design Principles. The last activity in the human-centered design process is to evaluate the design. The evaluation is required to be from the perspective of the users and suggested methods to use are user tests and inspection-based evaluation (performed by usability experts).

The Human-Centered Design Process

![Diagram of the human-centered design process]

Figure 3: Structure of the Human-centered Design Process. Adapted from ISO 9241-210 [46]

3.11.2 Personas & Persona-Based Scenarios

In the book About Face: the Essentials of Interaction Design by Cooper, Reimann, Cronin and Noessel [91], personas are described as descriptive models of users intended to use a system. The authors also introduce proto persona, which is a persona that is not created from an actual user, but from gathered knowledge about a typical user. The aim of personas is to communicate how groups of users behave, think, want to accomplish and why, and are based
on research on actual user behavior. According to Cooper et al., Personas create the ability to understand the specific goals of users in specific contexts.

Further, Cooper et al. suggests a structured process to create personas that can be used by both novice and experienced designers to identify behavior patterns and transform the patterns into user archetypes, i.e. personas. The process includes the following steps:

1. **Group interview subjects by role:** After conducting the research on user behavior the interviews should be grouped by the role of the user. For enterprise applications, the role of the user most commonly corresponds to the job title. [91]

2. **Identify behavioral variables:** The distinct aspects of the observed behavior of each role can be listed based on variables: Activities, Attitudes, Aptitudes, Motivations and Skills. [91]

3. **Map interview subjects to behavioral variables:** The most important part of this step is to map the subjects on each variable axis in relation to each other, rather than to a precise point in the range. This step helps to identify significant behavior patterns by identifying clusters of subjects across multiple axes. [91]

4. **Synthesize characteristics and define goals:** For each identified significant behavioral pattern, details from the data should be specified and should include information about [91]:
   - The activities and motivations
   - The environment of use
   - Pain points related to current solutions
   - Demographics related to the behavior
   - Skills related to the behavior
   - Attitudes and emotions related to the behavior
   - Interactions with people or systems
   - Alternative ways of doing the same thing

   At this stage, the persona can be assigned a first and last name and demographics such as age, job title and geographic location to make the persona more alive. The goals of each persona are also defined during this step to make the personas effective as a design tool. [91]

5. **Check for completeness and redundancy:** Ensure that there are no important gaps in the characteristics and goals of the personas and change or discard personas that are too similar to each other. [91]

6. **Designate persona types:** Categorize and prioritize the personas according to the types: primary, secondary, supplemental, customer, served and negative. [91]

7. **Expand the description of attributes and behaviors:** The last step is to write the narrative description of each persona, summarizing the most important details observed. To make the personas more real, chose photos where the subject is involved in an appropriate and realistic activity. [91]

According to Cooper et al., persona-based scenarios focus on behavior and thoughts of people and allows the design to be started from a story of an ideal experience from the perspective of the persona. A persona-based scenario is a concise description of one or several personas using a product to achieve a specific goal. Cooper et al. claims that the imagine of future interactions between the persona and the system are central when working with persona-based scenarios.
3.11.3 User Journey Mapping

In the article "When and How to Create Customer Journey Maps", Kaplan [92] describes that a user journey map is used to understand and address the needs and pain points of the user of a system. It is a visualization of the steps the user is processing to achieve a certain goal and aims to communicate insights for the design process. The user journey map also includes the thoughts and emotions experienced by the user during the process of performing a task. This gives a holistic view of the user experience and creates a shared vision that is concise and easy to remember, according to Kaplan.

3.11.4 Conceptual Design

A conceptual design can be compared to a blueprint and describes what the user does and where, but without any detailed look and feel [93]. Nam, Childs and Sohn [94] states that there is a lack of structured design methods to effectively support conceptual design. This causes several difficulties in the design practice, such as communication issues and misunderstandings that negatively influence the quality of the design decisions being made. In the article "A Design Model and Tackles for Systematic Conceptual Design" Nam, Childs and Sohn [94] propose the linked node design model for a systematic conceptual design. The method uses nodes to represent the properties of an outcome and links between the nodes. The model embraces the non-linear nature and is a process to discover and link visible and invisible properties in harmony, according to the authors.

In the book Conceptual Design for Interactive Systems: Designing for Performance and User Experience, Parush [93] presents an iterative design methodology to create the conceptual design of an application. According to Parush, a conceptual model can be described as a number of layers: the function layer, the configuration layer and the navigation layer. The function layer contains all the parameters, items and actions needed for the functionality of the application. The functionality can be grouped using one of the three following approaches: task-oriented, object-oriented or content-oriented. The task-oriented approach groups functionality that serves a common purpose and achieve a certain goal. The object-oriented approach chunks actions related to a specific object and the content-oriented method collections of information. Typically, several groups of the mentioned approaches are combined into a compound chunk to make the user able to achieve the goal.

Further, Parush describes that the configuration layer creates relations between the functional layer and the places the user needs to visit to perform a task. The layer contains conceptual model elements that aim to provide the user with the opportunity to interact with the system and to support the accomplishment of at least one task. The navigation layer helps the user to get from one element to another and a conceptual navigation map describes the possible or required steps to complete a task. There are two primary considerations influencing the path a user will take between the conceptual model elements: the model of the configuration layer and the workflow of the interaction. The physical places of the conceptual model elements have a significant impact on the human performance and the usability and user experience of the system, according to Parush.

Moreover, Parush presents the following five human performance factors that can be used to assess the implications of the conceptual model: mental models, location awareness, visual search effectiveness, operational load and working memory load. There are strong connections between the human performance factors and the usability of the system. In order to understand an interactive system, the user needs to create a mental model of the system. If the mental representation corresponds to the conceptual model and structure, it is easier to understand. Location awareness is important, since the user needs to know the location to be able to reach the destination. Parush claims that good location awareness includes knowing
the visited places, the actual location and the places to be visited, and increases the probability to faster get to the correct destination.

The visual search is influenced by the number of elements in the same place and all the non-target elements become distractions. Having less elements increases the probability of finding the target faster. However, a high visual search effectiveness may decrease the location awareness, according to Parush. Operational load refers to the amount of mental and physical effort required to perform actions and interact with the system. A lower number of physical places and alternative paths in the conceptual model will reduce the operational load and a simple flow of interactions increases the likelihood of performing a task faster and more accurate. To be able to accomplish a certain task, the user needs to store information about what has been done, what is being done at the moment and what is left to be done. Parush claims that having fewer steps in the interaction process will reduce the working memory load.

The design method described by Parush starts with the process of defining the function-layer. Before the process is started, a research of the needs, goals and behaviors of the intended users needs to be done. The findings from this research will be the basis for the conceptual design. The main steps in the workflow of the method are presented below.

1. Function-layer:
   a) Define functional chunks for the core of the application. The chunks can be task-oriented, that focus on a task of the user, or object-oriented, where the chunk is an object from the product domain.
   b) Link the chunks that has the same goal, are frequently performed together or a part of the same workflow.
   c) Revisit and revise

2. Configuration-layer
   a) Represent each functional chunk as a conceptual model element
   b) Define compound conceptual elements based on the links between the functional chunks.
   c) Look for a pivotal element i.e. elements that fulfill one or several of the following criteria: frequently used, critical and essential, a common steady state or represent the business objective.
   d) Revisit the configuration and revise the sketch to express the pivotal element identified in the previous step.
   e) Revisit and revise

3. Navigation-layer
   a) Define entries and exits in the conceptual model and the directions of the links between the conceptual elements.
   b) Assign the conceptual elements to physical places with consideration to the human performance criteria previously described in this chapter.
   c) Start prototyping a low-fidelity prototype that outlines the conceptual elements.
   d) Evaluate and revisit the previous steps to improve the conceptual model.

Following these steps results in a conceptual design, based on research of the users, that will form the foundation for the detailed design [93].
This chapter describes and motivates the methods used this master’s thesis project. The chapter is divided into two main sections: Pre-study and Implementation. An overview of the study is described in section 4.1 Study Structure and detailed descriptions of the methods used during each phase of the study are provided in the following sections. The theory of the methods used for requirements elicitation, development and evaluation is described in the theory chapter of the thesis. The additional theory presented in this chapter mainly relates to the research methodology.

4.1 Study Structure

The structure of the study consisted of three main objectives that gave rise to the conclusion. An overview of the structure can be seen in Fig. 4.1. The pre-study contained the major part of the literature study and the process of the requirements elicitation. The author of this thesis had access to four supervisors from the department of Research & Development, Service & Asset at IFS throughout the project (henceforth referred to as "the supervisors").

The implementation phase was performed iteratively through four iterations. Iteration one and two focused on the design process and the creation of a conceptual design and a detailed design prototype. Both iterations contained user tests that was the basis of improvements of the prototypes and refinements of the requirements of the system. A competitor analysis was performed in iteration number three to investigate if competitive vendors provide any budget or forecasting functionality for service contracts. Iteration number four included the implementation of the final prototype. The user test in the end of the fourth iteration was the final evaluation of the developed prototype and was conducted to evaluate the usability in terms of effectiveness, as well as the overall usability. Finally, the research question was answered based on the outcome of the study. The method and the results were analyzed and discussed and from this, the conclusions were drawn.
4.2 Pre-study

During the initial phase, a thorough literature study was conducted to form the theoretical foundation of the thesis. The requirements elicitation process was also performed during the pre-study to have all initial requirements specified before the implementation phase. The methods used for the literature study and the requirements elicitation are described in detail in the following sections.

4.2.1 Research Methodology

The book *Research Methodology: Methods & Techniques*, by Kothari [95] presents a number of different research types: descriptive, analytical, applied, fundamental and empirical. According to Kothari, **descriptive research** aims to describe what has happened before or what is happening in the present and methods used in the field are generally surveys or other comparative methods. **Analytical research** has the purpose to analyze and critically evaluate already available data or information. The goal of **Applied research** is to find a solution for a problem in a society or an industrial organization. **Fundamental research** is concerned with formulations of theories and generalizations. Finally, **Empirical research** relies on observations to make conclusions that can be verified by experiments, according to Kothari. Since this master’s thesis project aims to solve an existing problem for an organization, this is considered to be an applied research.

Further, the research types described by Kothari, can be divided into two research approaches: quantitative and qualitative. The **quantitative** approach includes generation of quantitative data that can be measured and used for quantitative analysis. The **qualitative** approach, on the other hand, produces soft values or data that cannot be subject for quantitative analysis, according to Kothari. This study has a qualitative approach since the collected data is primary gathered from interviews and the number of test users in the final usability does not qualify for a quantitative analysis. Usability tests for collecting quantitative data to perform statistical analysis require a large number of test users, at least 30 [50] which was not affordable in this study.
4.2.2 Requirements Elicitation

During the pre-study the requirements elicitation was performed to define an initial specification of the requirements of the functionality to be developed. The requirements elicitation process was conducted according to the RMUS provided by Bik, Lucassen and Brinkkemper [84] (see section 3.10.2). The method used to gather the requirements elicitation was a combination of internal workshops and semi-structured interviews. According to Saiedian and Dale [80], the most important is not the method used for the elicitation, but the participation of end users in the process of defining the requirements. The identified requirements were the basis for the creation of user stories. The user stories were used to represent the requirements in the product backlog and to verify that the desired functionality was implemented. Semi-structured interviews, that allows for conversation and discussion between the interviewer and the interviewee [82], were used to gather requirements and to understand the domain and the needs of the end users. Interviews and user stories are the most frequently used methods for requirements elicitation among requirements engineers [81], [67].

The workshops were conducted together with the supervisors from the company in collaboration. From the workshops, one proto persona (henceforth persona) and six scenarios were created to be used as the basis for discussions during semi-structured interviews held with additional persons from the company. Cooper et al. [91] states that personas create the ability to understand the specific goals of users in specific contexts. According to Denger and Olsson [79], scenarios ensures that the right functional requirements are gathered from the start, since scenarios are easy to understand for both technical and non-technical stakeholders. In total, four employees with close customer relations at IFS were interviewed during the requirements elicitation process to verify the persona and the scenarios. The group of interviewees has a wide knowledge about the customers using the service and management solution in IFS Applications and aimed to represent the overall needs of the end users.

The persona and the scenarios were verified during the interviews and additional feedback were collected to redefine the persona and the scenarios, and to make further delimitations of the scope. From the final version of the persona and the prioritized scenarios, a user journey was created from the scenario with the highest priority. A user journey gives a holistic view of the user experience that is easy to understand and remember [92]. The persona, the scenarios and the user journey were used to represent the requirements and served as the basis for the creation of user stories for the product backlog.

The requirements gathering phase resulted in user stories that were prioritized and assembled in the product backlog to be used in the development phase. The RMUS is easy to adapt and helps to leverage user stories as efficient as possible [84]. The project management tool JIRA was used to manage the product backlog and the sprint backlogs for each sprint. JIRA focuses on agile software development principles and lets the user prioritize, assign, track and report issues as well as visualize the work in progress [96].

4.3 Implementation

This section describes the methods used for the development and usability evaluation processes and the specific methods used in each sprint. The implementation phase is defined as the methods and activities performed during the design process and the implementation of the functionality, including the usability evaluations. Usability evaluations were performed at the end of sprint one, two and four, and was the basis for refinements of requirements and priorities of the user stories in the sprint backlog for the upcoming sprint. The evaluation aimed to identify usability problems to increase the usability of the product during the development process.
4.3 Implementation

4.3.1 Overview

The implementation was performed with an agile approach with inspiration from Scrum and was divided into four sprints. Agile software development allows for the ability to rapidly adapt to changes in software requirements [74] and are frequently used in the industry [66, 67]. Sprint one focused on the development of a conceptual design prototype and this was developed into a detailed design prototype during sprint two. The approach for the prototyping was evolutionary, i.e. the prototypes were refined after each iteration [49], and both paper and pen and the computer were used. The prototypes were evaluated at the end of each sprint through a test session with test users and refined according to the feedback. ISO [46] states that user test can be used in any stage of the development process to increase the usability. Sprint three included a competitor analysis to identify the benefits and drawbacks of the detailed prototype in comparison to similar functionality offered by other competitors. The resulting prototype was implemented as a high-fidelity prototype (see section 3.10.3) in the framework of IFS Applications, during sprint four. A final usability evaluation was conducted in the end of sprint four to measure the usability in terms of effectiveness, i.e. the accuracy and completeness of specific tasks, and perceived subjective satisfaction. The process of planning and performing the user test mainly followed the steps for good user tests described by Matera, Rizzo and Carughi [54] (see section 3.8.2). In agile and iterative development, an evaluation of the result should be carried out in each iteration according to Ritter, Baxter and Churchill [49]. Further, the definition of usability (see section 3.6.1) proposed by the International Organization for Standardization [43] has been applied in this study.

Agile software development methods are frequently used in the industry [66], [67] and minimize the overall risk of the project due to continuous feedback from the stakeholders and the flexibility to change requirements during the process [75]. The method was appropriate for this study since it allows the design to be redefined through each iteration with the aim to increase the usability. Projects most commonly implies several changes to fully satisfy the customer [71]. If a plan-driven method had been used, the final design and requirements would have been settled in the beginning of the project and the usability would not have been possible to increase throughout the project. Also, agile methods allow for continuous feedback from users which increase the probability of the right system being developed [75]. The whole project has been conducted with the Human-centered design process in mind (see section 3.11.1), i.e. the design process has included an iterative process of specifying/updating requirements, producing design solutions and evaluating the solutions until the design meets the needs of the end users. ISO [46] suggests that the Human-centered design process can be used for the design of all types of interactive system.

4.3.2 Sprint 1: Conceptual Design - Implementation

The first iteration included the process of developing a conceptual design. The results from the requirements elicitation (i.e. the persona, the prioritized scenario and the user journey) were used to imagine ideal interactions between the user and the system and formed the basis for the creation of the conceptual design prototype.

Ideas for the conceptual design were generated during a workshop with the supervisors at the company. The persona together with the user journey were used as the starting point to focus on the task of the user. The method used during the workshop was adapted from a worksheet for a crash course in design thinking created by the Hasso Plattner Institute of Design at Stanford University [97]. The workshop included two short sessions of sketching ideas and two sessions of discussing the pros and cons of the generated solutions. All sketches and ideas were documented and collected to be compiled into two conceptual designs.

The process of creating the two conceptual designs was based on the methodology suggested by Parush [93] (see section 3.11.4), in combination with the outcome of the workshop. First
a functional layer was created, by performing a task analysis, defining task-oriented, functional chunks of the core of the task and linking the chunks with the same goal and the same workflow. The task analysis can be seen in Fig. 5 and the identified functional chunks are presented in Fig. 6.

Figure 5: The task analysis performed during the first step of the conceptual design process

Figure 6: The blue areas represent the functional chunks identified during the conceptual design process

The configuration layer was created by defining compound conceptual elements based on the links between the functional chunks. A pivotal element was identified as the most frequently used element, in this case as “view”, see Fig. 7. The process was performed with paper
and pen and the sketches were revisited and revised before moving to the next layer. The reconfigured model is presented in Fig. 8.

**Conceptual Model Elements**

![Conceptual Model Elements Diagram]

Figure 7: The conceptual elements. The pivotal elements were decided to be the “view” and is marked with purple in the figure.

**Reconfigured Model**

![Reconfigured Model Diagram]

Figure 8: The reconfigured model with the pivotal elements and the links between each element.

Finally, the conceptual model was revisited and refined and the navigation layer was created by defining the entry and exit points and the conceptual model elements were structured to facilitate the workflow. Two low-fidelity prototypes were created from the conceptual model using Balsamiq. Balsamiq is a tool for creating fast wire-frames that focuses on the content of the prototype rather than the details [98]. This method combines research of the users and
a structured strategy to generate a conceptual design that will form the foundation for the detailed design.

4.3.3 Sprint 1: Conceptual Design - Evaluation

Three test sessions were conducted to evaluate and discuss the conceptual design. An overview of the participants of the test session, with the title and the country of employment, is presented in Table 3. All participants are employees at IFS and have a good knowledge about the needs and behaviors of the customers using the service and management solution of IFS Applications.

Table 3: Participants of the user test in Sprint 1

<table>
<thead>
<tr>
<th>USER</th>
<th>TITLE</th>
<th>COUNTRY</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Consultant</td>
<td>Germany</td>
<td>No</td>
</tr>
<tr>
<td>User 2</td>
<td>Senior Advisor</td>
<td>United Kingdom</td>
<td>Yes</td>
</tr>
<tr>
<td>User 3</td>
<td>Consultant</td>
<td>Sweden</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study, No = this is the first time the user is involved in the study

Each session was performed individually with each user and begun with a presentation of the persona (for user 1) and the user journey (all users) to verify the process of the task of the user. According to ISO, the models and concepts should be evaluated in relation to real world context. The test sessions had the structure of a semi-structured interview and the presentation of the user journey was followed by a presentation of the two conceptual designs and some questions to identify issues or possible improvements of the concept. The test session with user 1 was performed in-house and the test sessions with user 2 and user 3 were remote through a Skype meeting with a shared screen. All test sessions were recorded and each test session was summarized after the meeting. The pre-defined question used in the sessions were adapted from the book “SPRINT How to Solve Big Problems and Test New Ideas in Just Five Days” by Knapp, Zeratsky and Kowitz and are presented below:

- What did you like about this prototype?
- What did you dislike?
- If you should improve this solution, what would that be?
- How would you compare the two solutions?
- Which parts would you combine to create a new version?
- Which one worked better for you? Why?

The outcomes of the test sessions were summarized and compiled, and a final conceptual design prototype was selected according to the feedback from the participants in the test. Also, the persona and the requirements were updated according to the feedback from the test sessions. Zowghi and Coulin state that prototyping can be used in conjunction with other methods for requirements elicitation, such as interviews. Also, Andriole suggests that “quick and dirty” prototypes can be used together with one-to-one interviews to update
4.3. Implementation

the requirements and refine the interface according to the feedback. The final results from Sprint 1 are presented in the results chapter in section 5.2.3 Sprint 1: Conceptual Design – Final Results.

4.3.4 Sprint 2: Detailed Design - Implementation

The second iteration included the creation of a detailed design prototype of the core functionality of the business solutions and was based on the conceptual design prototype from the first iteration.

A sprint planning meeting was held at the beginning of the sprint with two of the supervisors at the company to decide the framework for development and discuss the architecture of the business solution. The decision to use the framework of IFS Applications for the development was made to make the prototype as real as possible.

A detailed design prototype was created by refining and adding details to the conceptual design prototype. The “Eight Golden Rules” suggested by Shneiderman and Plaisant [44] and the Human-center design principles suggested by ISO [45], [46], both described in section 3.7 Design for Usability, was considered during the design process for the detailed design prototype. The wire-frames of the prototype pages were created in Balsamiq and the visual elements, such as graphs and tables, were created in Microsoft Excel. The final prototype was then assembled in Microsoft Power Point where additional details were added.

As the developed business solution should be implemented using the framework of IFS Applications, most of the color schemes and typography rules are already settled and unchangeable. This was considered when creating the detailed prototype and will make the user interface of the developed business solution consistent with the overall application used for the management of service contracts. The consistency helps the end users to feel familiar with the system and thereby to increase the learnability and understandability perceived by the users [44].

4.3.5 Sprint 2: Detailed Design - Evaluation

Five test sessions were held to evaluate the detailed prototype and to identify possible improvements. One of the testers had also participated in the evaluation of the conceptual design, three were completely new to the project and one had only been interviewed during the requirements elicitation. An overview of the participants in the test sessions, with title and country, can be seen in Table 4.

Table 4: Participants of the user test in Sprint 2

<table>
<thead>
<tr>
<th>USER</th>
<th>TITLE</th>
<th>COUNTRY</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Consultant</td>
<td>Germany</td>
<td>Yes</td>
</tr>
<tr>
<td>User 2</td>
<td>Consultant</td>
<td>United Kingdom</td>
<td>Yes</td>
</tr>
<tr>
<td>User 3</td>
<td>Consultant</td>
<td>Germany</td>
<td>No</td>
</tr>
<tr>
<td>User 4</td>
<td>Product Solution Manager</td>
<td>Sweden</td>
<td>No</td>
</tr>
<tr>
<td>User 5</td>
<td>Consultant</td>
<td>Sweden</td>
<td>No</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study, No = this is the first time the user is involved in the study.
Each session was performed individually with each user and had the structure of a semi-structured interview with three pre-defined questions for each page of the prototype. The pre-defined question used in the sessions were adapted from the book “SPRINT How to Solve Big Problems and Test New Ideas in Just Five Days” by Knapp, Zeratsky and Kowitz [64] and are presented below:

- What did you like about this page? Why?
- What did you dislike about this page? Why?
- If you should improve this page, what would that be?

All test sessions were performed individually with each user through a Skype meeting and the persona, the user journey and the requirements were presented to User 3, User 4 and User 5, since they were completely new in the project. Each user was told to not focus on the colors or the details, such as font sizes, etc., but rather on the information presented. The prototype was presented page by page and the three pre-defined questions were the starting point of discussion for each page. The test sessions were recorded and the outcomes were summarized after each meeting. The outcomes of all five test sessions were compiled and the prototype was refined according to the feedback from the testers.

Finally, the refined prototype was evaluated and discussed by the supervisors at IFS through a heuristic evaluation to identify possible usability issues. The method used for the heuristic evaluation was subjective judgment [54]. Some design decisions, outlined in section 5.2.6 Sprint 2: Detailed Design - Final Results, were made and the supervisors gave additional suggestions of improvements for the final version of the prototype. The prototype was adjusted according to the feedback from the supervisors and the results from Sprint 2 are presented in the results chapter in sections: 5.2.4 Sprint 2: Detailed Design – Implementation, 5.2.5 Sprint 2: Detailed Design – Evaluation and 5.2.6 Sprint 2: Detailed Design - Final Results.

### 4.3.6 Sprint 3: Competitor Analysis

A competitor analysis was performed in sprint three to investigate if the competitors provide any similar functionality for service contracts. If any functionality for budgeting or forecasting of service contracts was detected, the graphical user interface of this functionality was evaluated to identify the benefits and drawbacks to improve the prototype developed in sprint two.

Google was used to identify the main competitors among the different ERP vendors, and to investigate what functionality each vendor offers for service contracts. Only vendors with full ERP solutions were considered. Some of the different search strings used are listed below:

- "ERP systems service contracts"
- "ERP system market shares"
- "service contract management"
- "budget service contracts"
- "estimate profitability service contracts"
- "forecasting of service contracts"
To investigate the functionality provided for service contracts by each vendor, Google and YouTube were used along with the respective website of each vendor. If no functionality for budgeting or forecasting of service contracts were found on any of the given sources the functionality was marked as absent. If any budgeting or forecasting functionality were found for any other types of contract, this was noted.

4.3.7 Sprint 4: High-Fidelity Prototype - Implementation

The user stories in the product backlog from sprint two were used as a guide for the implementation of the functionality. The detailed prototype was followed if possible with regards to the limitations of the framework. Memmel, Reiterer and Holzinger [87] outlines that a high-fidelity prototype can act as the base for the programming. Each user story was implemented and tested before it was moved from "in progress" to "done" in the sprint backlog in JIRA. The remaining user stories at the end of the sprint were left in the product backlog.

4.3.8 Sprint 4: High-Fidelity Prototype - Usability Evaluation

The outcome of the development of sprint four was the final product and was used to analyze the effectiveness and the overall usability of the developed business solution to answer the research question. Six participants were selected to participate in this user test. Even though there are diverging opinions on whether five users are enough or not [61], [63] five participants are suggested by Nielsen [60] and widely accepted in general [61]. Since user tests are conducted in each iteration of the development process, five participants were considered enough for this study. This is also supported by Nielsen, who argue that three tests with five participants are better than one test with 15 participants [60].

Since the sample of the participants in a user test should be representative of the end users [54], the group of testers in this user test was composed of six employees at IFS with close customer relations or previous experience of working with service contracts. One of the testers was used for the pilot test, which is recommended to perform to ensure that the test process works and that the tasks are understandable [54], [59]. Half of the group of the testers work as Business Systems Analysts, and all have experience of working with service contracts. The other half of the test group work as Consultants and work closely with the customers of IFS. They have a good knowledge of how the customer works with the system and are well aware of the needs of the users. Thus, the group of testers was considered to be representative of the end users. An overview of the participants can be seen in Table. 5.

<table>
<thead>
<tr>
<th>TITLE</th>
<th>COUNTRY</th>
<th>EXPERIENCE</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>1-5 years</td>
<td>No</td>
</tr>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>1-5 years</td>
<td>No</td>
</tr>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>+ 10 years</td>
<td>No</td>
</tr>
<tr>
<td>Consultant</td>
<td>Germany</td>
<td>5-10 years</td>
<td>Yes</td>
</tr>
<tr>
<td>Consultant</td>
<td>Germany</td>
<td>1-5 years</td>
<td>Yes</td>
</tr>
<tr>
<td>Consultant</td>
<td>UK</td>
<td>5-10 years</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study, No = this is the first time the user is involved in the study
4.3. Implementation

The final usability evaluation was a summative [49] evaluation that aimed to assess the usability of the final high-fidelity prototype. For the user tests, six tasks were created to be accomplished by the test users. The task completion rate was used to measure and evaluate the usability in terms of effectiveness, see [38] in section 3.8.2 User Testing, and the SUS questionnaire was used to assess the overall usability of the prototype. The tasks were created to represent real tasks to be performed with the implemented functionality in the business solution. The tasks used in the tests along with the success criteria are presented in Table 6.

Since the task completion rate is a quantitative measurement for the usability of a system [55], the tasks were created according to the guidelines for writing quantitative tasks provided by Meyer [59].

Table 6: Tasks & success criteria used for the final usability evaluation of the implemented high tech prototype

<table>
<thead>
<tr>
<th>TASK</th>
<th>SUCCESS CRITERION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK1: Find the service contract with the ID 'NA-100'</td>
<td>The contract is identified by the user</td>
</tr>
<tr>
<td>TASK2: Navigate to the details/analysis page for the service contract found in TASK 1</td>
<td>The user clicks on the details page for the contract found in the previous task</td>
</tr>
<tr>
<td>TASK3: At the provided page, initiate a new budget for that service contract</td>
<td>The user clicks on the 'Create New Budget' button in the analysis page</td>
</tr>
<tr>
<td>TASK4: Give the initiated budget a name of your choice and set the budget period from 2018-06-01 to 2020-06-01</td>
<td>The user gives the budget a name and enters a from date and to date</td>
</tr>
<tr>
<td>TASK5: Find the service row with the service line number '1'. Make an estimation of '5' yearly occurrences for that service for budget period no 1. Make an estimation of '10' yearly occurrences for that service for budget period no 2</td>
<td>The user clicks the 'Edit' button and edits the yearly occurrences for the service line in two different budget periods</td>
</tr>
<tr>
<td>TASK6: Finish the budget you have created</td>
<td>The user hits the 'Finish' button</td>
</tr>
</tbody>
</table>

The tasks were measured binary, with a 0 representing a failed task and a 1 for a completed task, since this is the simplest and most commonly used method [55]. The success criteria, i.e. the correct answer of each task, was defined before the usability evaluation was started, see Table 6 and the tasks were identical for all test users. The task success rate was evaluated according to the average task completion rate, claimed by Sauro [58].

A pilot test was performed with one of the test users to ensure that the tasks were understandable and to improve the reliability of the results. Some formulations of the tasks were revised after the pilot test to make the tasks easier to understand. Each participant conducted the test individually and the behavior was observed and documented to identify difficulties related to each task. The test sessions were concluded with the SUS questionnaire to measure the overall usability of the prototype. The SUS questionnaire is well known and provides a good assessment of the overall usability of a system [52], [53].

Each test session began with a short description of the structure of the test. A Google form, consisting of 3 sections, was used to collect the answers from each participant. The first sec-
tion requested some personal information about the test user. The second section presented the tasks and the user had to fill in whether he or she believed that the task was completed or not. The final section included the SUS questionnaire and was filled in by the user after all tasks were completed. The form can be seen in Appendix: User Tests - Sprint 4. The SUS score was evaluated in comparison to the Sauro-Lewis Curved Grading Scale [53] and the guidelines suggested by Bangor, Kortum and Miller [Bangor2008]. The outcomes of the test in this sprint served as the final usability evaluation of the developed system, as well as the base for suggestions of improvements for future development.
This chapter presents the results of the pre-study and the implementation, including the usability evaluation. The chapter follows the structure of the method and are divided into two main sections: pre-study and implementation.

5.1 Pre-study

This chapter contains the results from the pre-study. The problem formulation and the theoretical frame of reference were conducted during the pre-study and are the results are found in chapter 1.3 Research Question and 3 Theory, respectively. The results of the requirements elicitation are presented in the following sections.

5.1.1 Persona

The persona created to illustrate a typical user of the service and management solution are presented in Appendix: Persona - Pre-study. The persona was the result of the interviews and workshops to gather information about the needs of the users and the requirements for the system.

The persona is working as a Customer Relations Manager and has five years of experience of contract management. He likes his job, enjoys being social and has frequent contact with the customers. He has a background as a service technician at the company and has a good knowledge about the customers. The responsibilities of the persona are to manage the service contracts, of one to two large customers, and to reevaluate the contracts on a yearly basis to ensure the profitability of the contract. To make informed decisions for the reevaluation, he uses a spreadsheet to estimate the costs, revenues and expected number of reactive work orders. This requires a lot of manual work that the persona does not enjoy and decreases the reliability of the decisions regarding the reevaluation of the service contracts.

The goal for the persona is to make accurate and informed decisions for the reevaluation of the service contracts to ensure the profitability of the contract. The persona also has a personal
interest in profitable contracts, since he gets a commission on the profitability. To evaluate a contract, the persona needs to budget and forecast the revenues and costs of the contract for periods of one to ten years. The budget needs to be changed and updated continuously to reflect the reality and improve the accuracy. The persona does not have any financial background, and therefore the information needs to be easy to understand. The persona also needs the information to be accessible to be able to instantly respond to questions from the customers.

5.1.2 Scenarios

The requirements elicitation resulted in six scenarios, with different level of abstraction. The scenarios, prioritized in descending order with the most important first, are presented below:

1. **Budget for a service contract**
   The persona needs to make a budget of the costs and revenues of a service contract to estimate the profitability of the contract. He needs to estimate the profitability to make a decision for the reevaluation of the contract.

2. **Manual reevaluation of existing service contract**
   The persona needs to do periodical revaluations of the service contracts to ensure the profitability of each contract. He needs to evaluate an existing contract before he makes a renegotiation with the customer. The contract is activated when the contract manager and the customer agree.

3. **New service contract**
   A new service contract is initiated by the seller. The persona creates the contract, makes the budget and simulates the forecast to ensure a profitable contract. He begins a negotiation with the customer and when they agree he activates the contract.

4. **Customer complaints**
   A customer calls the persona and tells that he is unhappy with the service contract. The persona opens IFS Applications to get an overview of the profitability and the SLA of the service contract. If something is unsatisfying, he initiates a manual reevaluation of the contract.

5. **Deviation from budget**
   Persona visits a lobby for service contracts to get an overview of the contracts he is responsible for. He detects that one of his contracts has a significant deviation from the budget and needs further evaluation. He contacts his manager to discuss potential actions.

6. **Create follow-up information**
   The persona needs to create follow-up information for a customer, his manager or the company management.

To limit the scope of the project, the decision to initially focus on the scenario with the highest priority, i.e. scenario 1, was made in collaboration with the supervisors at the company. This decision affected the prioritization of the user stories in the initial version of the product backlog, presented in detail in section 5.1.5 Product Backlog.

5.1.3 User Journey

The persona and the scenario gave the result of a user journey, created to visualize the process and the emotional experience of how the user solves the task today, see. The emotional
5.1. Pre-study

curve resulted in a number of opportunities of improvements to consider in the design process.

To make a budget for the costs and revenues of a contract to estimate the profitability of the contract, some steps are needed to be completed by the persona. The first step is to find (or create) and open the contract the persona wants to make a budget for. This is easily done by the persona within the system. The second step is to copy all services of the contract into a spreadsheet and fill in the expected costs and revenues for each service before the key ratios can be calculated. This step requires a lot of manual work and makes the persona feel exhausted, thus the emotional curve drops. The persona feels revealed when this step is completed and the emotional curve goes up again.

**USER JOURNEY CHRISTOPHER LARSSON**

**SCENARIO**
Christopher needs to make a budget of the costs and revenues of a service contract to estimate the profitability of the contract. He needs to estimate the profitability to make a decision for the reevaluation of the contract.

**GOALS**
To make a decision about the reevaluation, Christopher needs to:
- Estimate monthly costs, revenues and hours for each service in the contract
- Calculate the profitability
- Analyze the budget for different time periods of the contract

<table>
<thead>
<tr>
<th>PHASES</th>
<th>ACTIONS</th>
</tr>
</thead>
</table>
| OPEN THE CONTRACT | 1. Find or create the service contract  
2. Open the contract |
| MAKE A BUDGET | 3. Copy all services from the contract to a spreadsheet  
4. Fill in expected costs and revenues for each service  
5. Calculate key ratios for different time periods |
| ANALYZE THE BUDGET | 6. Analyze the results of the budget  
7. Change or save the budget |
| MAKE A DECISION | 8. Make a decision for the reevaluation |

**EMOTIONAL EXPERIENCE**

**OCCUPRITIIES**
- Easy access
- Reduce inputs required by the user
- Make it possible to select time periods and reduce manual calculations
- Visual representation of the results of the budget for different time periods
- Make it easy to change the budget
- Increase the reliability of the decision

Figure 9: The user journey created during the requirements elicitation

When everything is filled in and the estimations are done, the outcome of the budget needs to be analyzed to be able to make a decision about the reevaluation as the last step of the process. There are a lot of numbers in the budget and the persona gets confused and doubts that the budget is filled in correctly. To get an understandable overview of the profitability, the user needs to create graphs and calculate the key ratios in the external program used to create the budget. The persona feels revealed when the budget finally looks reasonable and makes a decision to reevaluate the contract. Although the contract seems to be profitable according to the numbers in the budget, the persona is feeling uncertain about the decision and doubts that the outcome of the budget is reliable.
The first opportunity for the design process identified from the emotional curve was that the service contracts need to be accessible by the user. Further, the inputs and manual calculations required by the user need to be reduced and it should be possible to select a time period for the creation of the budget to automatically get the accurate data. To increase the understanding of the analysis of the budget, there should be visual representations of the budget for different time periods. Finally, all these opportunities should contribute to an increased reliability of the decision for the reevaluation. The opportunities and potential solutions are presented in Table 7.

Table 7: Opportunities of improvements identified in the user journey and possible solutions

<table>
<thead>
<tr>
<th>OPPORTUNITY</th>
<th>POSSIBLE SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible</td>
<td>Page with an overview of all contracts</td>
</tr>
<tr>
<td>Reduce manual input</td>
<td>Fetch data from the system. Suggest values for the estimations of reactive service requests</td>
</tr>
<tr>
<td>Select time period</td>
<td>Add a time period to the budget creation and to the analysis</td>
</tr>
<tr>
<td>Increase understandability</td>
<td>Use graphical representations to visualize the actual and budgeted values</td>
</tr>
<tr>
<td>Change budget</td>
<td>Add the ability to save several budgets and to change an existing budget</td>
</tr>
<tr>
<td>Increase reliability of decisions</td>
<td>Use data from the system and suggest estimations for the yearly occurrence of reactive services</td>
</tr>
</tbody>
</table>

5.1.4 Requirements

The requirements elicitation resulted in a number of main requirements of the system and are presented in the following list:

1. **Budget**
   - The lowest level of the budget should be monthly
   - It should be possible to see the total costs, revenues and hours for a contract
   - It should be possible to see the total costs, revenues and hours for each service in a contract
   - It should be possible to change the budget before it is accepted and activated
   - The data for the budget should be generated automatically as far as possible
   - The user should get suggested values for the required estimations

2. **Indexation**
   - An indexation of the revenues and costs should be applied each year

3. **Extra**
   - The budget should predict and visualize a “best case”, a “worst case” and a “most likely” outcome of the profitability
• It should be possible to save different versions of the budgets
• It should be possible to apply a seasonal distribution of the variables
• It should be possible to simulate a forecast of the expected profitability of a contract
• It should be possible to update the variables in the contract from the budgeting tool

The Service Level Agreement (SLA) and penalties corresponding to missed levels of the service agreement were excluded from this project.

5.1.5 Product Backlog

An initial version of the product backlog was also a result of the pre-study. The product backlog contains the user stories created during the pre-study, prioritized with the functionality with the highest priority at the top and are available in Appendix: C.1 Product Backlog - Pre-study.

5.2 Implementation

This chapter presents the results from the implementation process, including the conceptual design, the detailed design, the development process and the usability test of each iteration.

5.2.1 Sprint 1: Conceptual Design - Implementation

This chapter presents the two conceptual design prototypes evaluated by the test users in the user test conducted in Sprint 1. The design process resulted in two conceptual design prototypes. Both prototypes are based on the same conceptual model and the outcomes of the conceptual design workshop. The conceptual design prototypes that were evaluated in the user tests at the end of sprint 1 can be seen in Appendix: Conceptual Design - Sprint 1.

Both prototypes contain a detailed overview of the current service contract, a screen for making a new budget, a dialog to make the estimations and a preview of the generated budget. Prototype number two also includes a screen with an overview of all the contracts that the user is responsible for. Further, both prototypes include the ability to save different versions of a budget. The dotted lines in both figures define the scope of this project. The ability to import and export budgets and the detailed overview from and to excel is desirable, but the decision was made to leave this outside the scope due to the time limitation. Also, there is already functionality for importing and exporting data present in IFS Applications today.

The detailed contract overview in both conceptual design aims to give the user a quick overview of the key ratios by graphical representations and presents the details in a table. The screen shows both the actual and the budgeted values of the costs, revenues and hours of the overall contract and for each service in the contract. From this screen, a new budget can be created and the screen for “make new budget” contains some general settings on how the budget should be generated. For example, the time period for the budget and what the suggested values for the estimations should be based on. The estimations for the expected number of service requests for each reactive service is done in the “estimate” dialog present in both prototypes. The dialog suggests a value for each service in the contract and the user can make manual changes. A preview of the budget will be present in the screen “results of budget generation”, and are similar to the detailed contract overview, but the user has the ability to make manual changes of the details and see the changes in the graphical representations before the budget is accepted, saved and activated.

The screen “all service contracts” in concept 2 shows an overview of all the contracts the user is responsible for and should provide easy access to the service contracts, as well as a quick
5.2. Implementation

overview of the financial status of each contract. This screen is not existing in concept 1, where the contracts are assumed to be found in the system with the already existing functionality.

5.2.2 Sprint 1: Conceptual Design - Evaluation

This chapter presents the outcome of the usability evaluation of the conceptual design prototypes produced in the first sprint. The answers provided by each test user can be found in Appendix: User Tests - Sprint 1.

Outcome User 1

This test session was the first time that user 1 was contacted in this project and to introduce the scope, the persona was presented at the beginning of the test session. User 1 identified several details of the persona that were not reflecting the reality. For example, the real user of this system is much older than 35 years old, probably around 50 years old and is a top manager, according to user 1. The contract manager has made career within the company and has a good knowledge about the businesses of the company and about the needs of the customers.

Further, the contract manager receives a signed contract, from the seller, with all information and puts it into the system. The contract manager does not have direct communication with the customer, but gets questions from the coordinator, whom the customer has contacted. The most common questions are about the invoicing. The contract manager is the person who knows the system well and knows how to find the information in the service contracts.

Furthermore, user 1 did not agree that the contract manager is online all the time, but always busy with phone calls. Also, the contract manager only uses laptop in the management of service contracts and not a Surface or iPad Pro. The most interesting for the contract manager, is to know if a contract will be profitable or not in total during the life-cycle.

According to user 1, the contract manager does not have a commission or bonus in general. If there is a bonus, it is based on the customer satisfaction, not on the profitability. Paper and pen or Excel is used to calculate a budget for a contract and the first opportunity for the contract manager to adjust the contract is at the first revaluation. The contract manager uses experience, knowledge and gut feeling to make estimations on the number of reactive work orders. The typical contract manager is responsible for about 250-2500 service contracts. User 1 agreed on the user journey with no further comments or improvements.

The evaluation of the conceptual design prototypes, revealed that the key aspect of this functionality is to be able to visualize budget variations, according to user 1. A “worst”, “best” and “most likely” case should be represented graphically in the same screen to enable easy comparison. Also, penalty costs (for missed SLAs etc.) are important to consider for the budgeting and the same for the seasonal distribution of the costs. Many service providers have a huge variation in the costs depending on the seasons.

User 1 also claimed that it is also important to calculate the costs for invoice corrections, since this cannot be charged the customer. Further, the budget should be created as automatically as possible, and the system should ask for estimations of the number of reactive work orders and suggest values based on the history. The user of the system wants both graphical representations of the actual and budgeted profitability as well as tables with details. An overview of all contracts that the contract manager is responsible for would be great with the possibility to categorize the contracts on profitability, expiry date etc. User 1 also liked the overview of the budget after it is generated to make manual changes and play with the numbers before it is accepted. Finally, user 1 agreed that it is important to be able to export data to excel and to be able to save different versions of the budget.
5.2. Implementation

Outcome User 2

User 2 was one of the interviewees in the requirements gathering process and the persona had already been presented for user 2 earlier in the process. The test session was started with a presentation of the user journey and user 2 agreed on the user journey with no further comments or suggestions of improvements.

The presentations of the two conceptual designs identified that the estimation will be an important part of the solution, according to user 2. The estimation part is challenging since there might be a huge number of different variables to estimate in each contract. Therefore, user 2 believed that the estimation would be more difficult than expressed in the conceptual design prototypes. However, user 2 agreed on the overall concept of both prototypes but thought that the overview of all contracts (suggested in concept 2) was unnecessary. User 2 did not have any further suggestions for improvements of the conceptual design.

Outcome User 3

User 3 was one of the interviewees for the requirements elicitation and the persona had been presented and agreed by user 3 earlier in the process. The test session begun with a presentation of the user journey and user 3 agreed that the user journey reflects the reality with no further comments.

The evaluation of the conceptual design prototypes, exposed that it is important to suggest estimates to the user based on more information than the history of that specific contract, according to user 3. The suggestions could for example be based on a standard template for that service type, since there are always some contracts that are unprofitable. Further, user 3 highlighted the importance of budget variance and the ability to save different versions as well as the importance of the ability to select different time periods in the detailed overview of a contract. User 3 also liked the overview of all contracts presented in concept 2 and did not have any suggestions on further improvements of the conceptual design.

5.2.3 Sprint 1: Conceptual Design - Final Results

This section presents the final results of the persona, the requirements and the conceptual design prototype refined after the user tests in Sprint 1. These results acted as the base for further development in the upcoming sprint.

Persona

The user tests gave some additional feedback on the persona and this gave the result of an updated version of the persona. The result is presented in Appendix: Persona - Sprint 1.

The age, the experience and the work title of the persona were updated to reflect the information from the test users. The responsibilities got the additional information that the contract manager is responsible for 20 to 2500 contracts and the contract time was adjusted from one to ten years to one to five years. Further, tree more needs were added and one was removed, and the behaviors were updated to reflect that the contract manager does not have a daily contact with the customers and most commonly uses a laptop. The first new need was that the contract manager is most interested to know if the contract will be profitable or not in total during the life-cycle. The need of a prediction of the worst, best and base case of the budget was also added, as well as a seasonal distribution of the budgeted costs.
5.2. Implementation

Requirements

The outcome of the user tests in Sprint 1 resulted in an updated version of the requirements for the business solution. The result of the re-prioritization of the requirements is presented in the list below:

1. **Budget**
   - The lowest level of the budget should be monthly
   - It should be possible to see the total costs, revenues and hours for a contract
   - It should be possible to see the total costs, revenues and hours for each service in a contract
   - It should be possible to change the budget before it is accepted and activated
   - The data for the budget should be generated automatically as far as possible
   - The user should get suggested values for the required estimations

2. **Budget Variance**
   - A worst case, a best case and a most likely outcome of the budget should be displayed in the same screen
   - It should be possible to analyze different time periods of the budget
   - It should be possible to add a seasonal distribution to the budget to distribute the costs unevenly over a year
   - It should be possible to display different budget versions to make a comparison

3. **Indexation**
   - An indexation of the revenues and costs should be applied each year

4. **Extra**
   - It should be possible to simulate a forecast of the expected profitability of a contract
   - It should be possible to update the variables in the contract from the budgeting tool

The highest prioritized requirements remained the same as from the pre-study, but the budget variance was prioritized as number two instead of as an extra functionality. This includes a worst, a best and a base case of the budget and also a seasonal distribution of the costs. The possibility to select different time periods, as well as the ability to save different budget versions were also moved and included in this main requirement. The yearly indexation was moved down to the third place and the simulation of a forecast, as well as the ability to update the contract variables from the budget were kept as extra functionality.

Product Backlog

An updated version of the product backlog was a result of Sprint 1. The user stories were re-prioritized according to the updated requirements and the product backlog for sprint 1 is available in the Appendix: [C.2 Product Backlog - Sprint 1].
5.2. Implementation

Conceptual Design

The outcome of the user tests indicated that an overview of the contracts was desirable, and therefore the second conceptual design prototype (see Fig. 10) were chosen as the final result of Sprint 1.

5.2.4 Sprint 2: Detailed Design - Implementation

This section presents the result of the detailed design, developed during Sprint 2. The result of the implementation was a high-fidelity prototype that was the object for evaluation in the user tests conducted at the end of the sprint. The detailed prototype can be found in Appendix: Detailed Prototype - Sprint 2.

Service Contracts Overview

The start page of the detailed prototype contains an overview of all the contacts that the contract manager is responsible for. Each contract is represented in a list, containing the contract id, contract name, customer name, start date, end date, actual margin and the budgeted margin. There is a possibility to filter the list on different attributes, as well as search for a specific contract in the search box.

Service Contract Analysis

A click on the open button on a contract in the contract overview page will bring the user to a contract analysis page, presented in Fig. 11. Information about the contract, including the name of the customer and the start and end date of the contract, is presented at the top of the page. Below this information, there is a section that lets the user decide details about the displayed information, such as start and end, as well as the budget version. There is also a shortcut to create a new budget and the graphical representations will be updated first when the user presses the update button.
5.2. Implementation

This detailed prototype contained six graphical elements displaying different financial information regarding the actual and budgeted values of the contract. The user can change the view to display the information on a monthly basis instead of a yearly in the bottom of the page. The cost analysis and the revenue analysis display the variance from the budget for each time period. The service breakdown groups the revenues and costs by the service and provides an overview of the actual and budgeted costs for each service. A new budget can be created both from the top and the bottom of the page.

![The first version of the contract analysis page of the detailed prototype for the budgeting functionality](image)

**New Budget - Settings**

An assistant will guide the user through the required steps to create a budget for a contract. The new budget requires the user to type a name for the budget, as well as selecting the time period that this budget will be valid for. The last step is to select how the values of the variables that need to be estimated should be suggested. The user can also choose whether to make the estimations based on one year or one month.

**New Budget - Estimations**

The settings made by the user in the previous step are displayed at the top of the page. A design decision was made to only give the user the ability to estimate the occurrence of a service and not the details of that service. The user will get a list with all the reactive services in this contract that needs to be estimated to create a budget. The list will include information about the work type and hours required, as well as the material needed for the service. The yearly occurrence column will contain suggested values based on the settings made in the previous step and the user has the ability to change the occurrence of each service if desired. The user is also required to select how the cost and revenues should be distributed over a year. The options are to: distribute evenly or to use a template for seasonal distribution.

**New Budget - Preview**

The last step of the budgeting process is a preview of the results of the budget estimations. The preview includes the same elements as the contract analysis page with an additional
5.2. Implementation

table where the user can make changes to the estimations made earlier. When the budget is accepted by the user, the user will be directed to the contract analysis page where the values are updated with the information from the new budget.

5.2.5 Sprint 2: Detailed Design - Evaluation

This section presents the results of the user tests conducted at the end of sprint 2 to evaluate the detailed design prototype. The section is divided into subsections with titles corresponding to each page of the prototype that was evaluated. Summaries of all the outcomes of the user tests with each user can be read in Appendix: User Tests - Sprint 2.

Service Contracts Overview

User 1 believed that the service contracts overview page was an okay start page for the budget functionality and that the actual and budgeted margins are interesting to see at this point. User 2 wished to see the customer ID and to sort the contracts by the customer, but also liked the ability to see the actual margin. User 3 had a lot of suggestions of improvements for the service contracts overview. This user would like to have a picture of the contract manager at the top of the page, together with the title “My Service Contracts” to make a connection between the contracts and the contract manager. Further, the list should also contain the contract type and it should be possible to group the contracts by customer and the contract type.

This page is the first step of this functionality and should be more interesting and spectacular for the user, according to User 3. A suggestion was to include some graphics on this page as well to create a good impression of the functionality from the beginning. User 4 and User 5 thought that the page in total was an okay entry point. User 4 suggested that the open button should be removed from each line and replaced by a button in the top. User 5 pointed out that not all service contracts have an end date on the contract. A contract might get extended by one year at the time as long as it is not canceled by the customer.

Contract Analysis

User 1 liked the first impression of the contract analysis page but would like to see a more detailed breakdown of the costs and revenues. According to User 1, it is important for the customer to be able to analyze the profitability of the contract on object level as well. The analysis should be both yearly and monthly and be performed on the whole contract, on each service and on each object. User 2 liked the graphs and believes that this kind of functionality would be very useful for the customers. A suggestion of improvement was to use rolling years and to help the user to select time periods by adding functionality to select last year, etc. User 2 also would like to drill down the costs and revenues to an object level. User 3 expressed that the customers would be very happy to have this functionality. It could be useful for some customers to use rolling years, since a business year is not always from January to December. User 3 thinks that it is too complicated to drill down the information to an object level and that the total of each service type is okay for this project. User 4 would like the cost analysis broken down into cost type as well and suggested that the bars for the actual and budgeted values should be displayed closer together. User 5 pointed out that a service might have sub-services where the actual costs rise. In that case, it would be interesting to have the service breakdown of each sub-service as well.

New Budget - Settings

All test users understood the settings page and did not have any suggestions for improvements. User 1 was a bit confused whether the budget was for the whole contract or only...
for one object. User 2 asked for the seasonal distribution at this stage. All users liked the possibility to choose how the system will select the pre-filled values for the estimations.

**New Budget - Estimations**

User 1 liked that the system provides values for the estimations and found it reasonable to make the estimation of the average yearly occurrence for one object. Both User 1 and User 2 suggested that a column with the number of objects connected to each service should be added. User 3 claimed that it will be more complicated in reality, but that this way of estimating the occurrences are okay for this project. User 4 found it confusing to make the estimation only for one year and to estimate the number of occurrences for one object. User 4 was also confused whether the budget is a budget or a forecast and argued that it is "weird" to make a budget for only one year if the contract is for several years. Therefore, User 4 claimed that the budget is not a budget but a forecast. One suggestion of improvements from User 4 was to add the periodic services here as well, but without the possibility to change the occurrence. User 5 pointed out that an object may have an object structure connected, i.e. contains several objects that the service will be executed on. A suggestion of improvement was to add a column with the number of objects connected to the service, as well as the number of sub-objects.

**New Budget - Preview**

All test users liked the graphical preview of the budgeted values as well as the table with the estimations. User 1, User 2 and User 4 were only interested to see the budgeted values at this stage while User 3 and User 5 liked to see the actuals as well. User 1 wanted the ability to compare the budget with actuals, but in another location in the business solution. User 2 and User 4 would like to be able to compare different budget versions. User 2 also requested the ability to analyze the budget on a more detailed level, such as on asset class. A timeline at the top of each screen to remind the user of the remaining time of the contract was also suggested by User 2. All test users liked the ability to change the estimations in the table here. User 3 suggested that it should be clearer that the table presents the total number of occurrences for all objects connected to each service. User 2 also pointed out that some kind of approval process could be required by some customers before the budget is activated.

**Heuristic Evaluation**

The heuristic evaluation with the supervisors at IFS gave the result of several design decisions as well as suggestions for improvements. The outcome of the evaluation is presented in detail in the section G.9 Outcome Group Evaluation in Appendix: User Tests - Sprint 2 and the main decisions and improvements are presented in the following list:

- **Contract Overview**
  - A scenario where no budget is made for a contract needs to be displayed
  - Change the heading to "My Service Contract Analysis"

- **Contract Analysis**
  - Do not display the numbers for the margins in the graph
  - Add the total costs, revenues, margin and hours for the selected time period
  - Restrict the monthly view to only display a 12-month period
  - Might be better to use line graphs for the service breakdown since the user is only interested in the trends and not the specific numbers
5.2. Implementation

- Break down the revenues and costs in different types

- **Budget Settings**
  - Rename "History" to "Last Year Outcome"

- **Budget Estimations**
  - Remove: material, quantity, work type and task template id
  - Add: task template description, total service cost and total service revenue
  - Add one step in the assistant for each year to make it possible to vary the budget over different years
  - Add the seasonal distribution on each service line

- **Budget Preview**
  - Remove the graphs at this step
  - Only display a table with the total budgeted costs, revenues, margin and hours
  - It should not be possible to change the budget at this stage

5.2.6 Sprint 2: Detailed Design - Final Results

This section presents the final results of the requirements and the detailed prototype updated after the evaluation by the test users in Sprint 2. The persona and the user journey remained unchanged after the user tests in this sprint and the requirements and the detailed design prototype acted as the base for the implementation started in the next sprint.

**Requirements**

The outcome of the user tests in Sprint 2 resulted in an updated version of the requirements for the business solution. The result of the updated requirements presented in the list below:

1. **Create Budget**
   - The lowest level of the budget should be monthly
   - It should be possible to see the total costs, revenues and hours for a contract
   - It should be possible to see the total costs, revenues and hours for each service in a contract
   - It should be possible to change the budget before it is accepted and activated
   - The data for the budget should be generated automatically as far as possible
   - The user should get suggested values for the required estimations

2. **Save Budget**
   - It should be possible to save different budgets

3. **Visualize Budget**
   - It should be possible to visualize different budgets in the same screen
   - It should be possible to compare the budgeted values with the actual values
   - It should be possible to select different time periods for the visualization, for example monthly, quarterly and yearly (or accounting periods)

4. **Visualize Budget Variance**
5.2. Implementation

- A worst case, a best case and a most likely outcome of the budget should be displayed in the same screen
- It should be possible to add a seasonal distribution to the budget to distribute the costs unevenly over a year

5. Yearly Indexation

- An indexation of the revenues and costs should be applied each year

6. Extra

- It should be possible to simulate a forecast of the expected profitability of a contract
- It should be possible to update the variables in the contract from the budgeting tool

Cost and revenue breakdown for each service on object level were added to the list of excluded items for this project. Also, the decision to primarily focus on number 1 to 3 for the continued development were made in cooperation with the supervisors.

Product Backlog

An updated version of the product backlog was a result of Sprint 2. The user stories were re-prioritized according to the updated requirements and the product backlog for sprint 2 is available in the Appendix: C.3 Product Backlog - Sprint 2.

Detailed Design

The prototype was refined according to the feedback from the test users and the supervisors at the heuristic evaluation. To improve the first impression and the usability of the service contracts overview, some additional financial information was added and the layout was redesigned. A time progress bar was added to display the remaining time of the contract. The first version of the detailed prototype is presented in Fig. 12 and the redefined version can be seen in Fig. 13. The arrow for each information box in the refined version aims to instantly give the user an indication about whether the contract is doing better than budget or if it needs to be further inspected. The blue-green color indicates a positive variance from budget, while a red-pink color indicates a negative variance. The contract type and a timeline that represents the time elapsed of each contract were added in the heading of the individual contracts. A contract without a budget were also added in the refined prototype to illustrate the scenario where no budget is made.

![Figure 12: The first version of the detailed design of the service contracts overview page](image1)

![Figure 13: The final version of the detailed design prototype of the service contracts overview page, refined after user tests and the heuristic evaluation](image2)
The first version of the contract analysis page is presented in Fig. 14 and the final version for sprint 2 is displayed in Fig. 15. The contract analysis page was updated with the addition of the same information as the service contracts overview page in the top of the page to remind the user on the current financial status of the contract. The settings for the view mode were moved to the selection of the date and the budget version. The decision was made to have the yearly view and the total contract period (or five years if there is no end of the contract) as the default for the analysis page. The date selection is disabled for the yearly view and gets enabled when the monthly view is selected and the date selection is restricted to a maximum of 12 months.

The colors in the graphs were updated to use the color scheme from IFS Applications and two new graphs for cost and revenue breakdown were added. An hour analysis was added and the service breakdown was changed to only display one service at a time. Three new buttons to instantly display the analysis of the service with the "highest margin deviation", "highest margin" or "lowest margin" were added to help the user identify interesting services. The contract analysis graph and the graphs for each service were updated with a box for the period total to display the accumulated values for the selected time period.

The budget settings page had minor changes, for example, one of the checkboxes was renamed from "history" to "actual outcome last year". The budget estimations page was updated to also include the PM-services and the seasonal distribution were added to each line, see Fig. 16 and Fig. 17. The "Service" column was renamed to "Service Contract Line" and...
the service id and service name were both included in this column. The task template id was removed and replaced by a task template description. The costs, revenues and hours were updated to display the total according to the estimations. Finally, one step for each year of the budget period were added to make it possible to make different budgets for different years.

Figure 16: The first version of the detailed design of the budget estimations page

Figure 17: The final version of the detailed design prototype for the budget estimations page, refined after user tests and the heuristic evaluation

The budget preview page is the final step of the creation of a budget and after the user tests and the heuristic evaluation the decision to remove all graphs from this page was made, see Fig. 18 and Fig. 19. The reason for this is that the budget is already visualized in graphs at the contract analysis page. The ability to change the estimations at this stage was also removed and the preview now only displays a total summary of the budget created.

Figure 18: The first version of the detailed design of the budget preview page

Figure 19: The final version of the detailed design prototype for the budget preview page, refined after user tests and the heuristic evaluation

5.2.7 Sprint 3: Competitor Analysis

The result of the investigation of the service contract management functionality provided by the main competitive ERP vendors is presented, in alphabetic order in Table. Seven ERP vendors were investigated and the result was that none of the vendors provided functionality
for budget and forecasting of key ratios of service contracts within the ERP system. SAP does provide predictive analytics for contract consumption of purchase contracts, but not for service contracts. Moreover, no other similar functionality for budget and forecasting within the ERP systems were identified during this study.

Table 8: Overview of budget functionality in service contract management solutions in ERP systems of competitive vendors. N = no, Y = yes. *For service contracts (functionality found in this study)

<table>
<thead>
<tr>
<th>VENDOR</th>
<th>BUDGETING FUNCTIONALITY*</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epicor</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>IBM</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>Infor</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>Microsoft</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>Oracle</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>SAGE</td>
<td>N</td>
<td>No similar functionality found</td>
</tr>
<tr>
<td>SAP</td>
<td>N</td>
<td>Does have predictive analytics and graphical visualization for contract consumption of purchase contracts [99]</td>
</tr>
</tbody>
</table>

Since no functionality for budget and forecasting of service contracts were found at any of the competitive ERP vendors, no further evaluation of the functionality of the competitors could be performed.

5.2.8 Sprint 4: High-Tech Prototype - Implementation

The result of sprint 4 was a high-tech prototype for the budget and forecasting functionality of service contracts in IFS Applications. A page containing all the service contracts that the current user are responsible for was implemented. The service contracts can be displayed as a list, see Fig. 20 or as cards, see Fig. 21 depending on the choice of the user.

Figure 20: Implemented page for the service contracts overview displayed as a list

Figure 21: Implemented page for the service contracts overview displayed as cards

Each row in the list, or each card, displays the information about one contract. The information shows the id of the customer, the company, the id and name of the contract and the contract type. Also, the from date and the expiry date along with a progress bar that displays
the time elapsed and left of the contract. Furthermore, there are a budgeted and actual margin together with the margin deviation from the budget. The margin deviation is displayed with a green color and an up-arrow if the actual margin is higher than the budgeted. If the deviation is negative, a red color and a down-arrow will be displayed together with the number. A gray color and a neutral arrow will be displayed if the actual margin and the budgeted margin is equal. When a contract line is marked in the list the buttons ‘Create New Budget’ and ‘Details’ appear. If the card view is used, the two buttons are always visible on each card, but the ‘Details’ button is represented by an arrow.

![Figure 22: The first page of creating a new budget for a service contract containing the main settings for the budget](image1)

![Figure 23: The second step of creating a new budget for a service contract containing the generated periods, service lines and estimations for the budget](image2)

### 5.2.9 Sprint 4: High-Tech Prototype - Usability Evaluation

The results of the final usability evaluation for the developed high-tech prototype are presented in Table. [9] and Table. [10]. The pilot test was performed by User 1 and the decision was made to include the outcome from the pilot test in the final evaluation. The changes made to the tasks between the pilot test and following test sessions were considered negligible, which supports the decision to include the outcome. User 1 to 3 (BSA:s) had not been involved in the project before, whilst User 4-6 (Consultants) all had been participating in the evaluation of the detailed design prototype.

As shown in Table. [9], the task success rate was 100% for task one, three, four and six, but 83% for task two and 67% for task five. The task success rate of task number two is slightly above the average completion rate of 78% [58], whilst the tasks success rate of task five is clearly lower than the average completion rate. The average effectiveness of all tasks performed in the high-tech prototype was calculated to 92%.
Table 9: Results of completed and failed tasks in the final usability test

<table>
<thead>
<tr>
<th>Q</th>
<th>User1*</th>
<th>User2</th>
<th>User3</th>
<th>User4</th>
<th>User5</th>
<th>User6</th>
<th>Eff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Average Task Success Rate 0.92

1 = completed task
0 = failed task

All test users managed to navigate to the page "My Service Contracts Overview" and to find the contract asked for in task number one. At the overview page, a list of nine service contracts, representing all contracts that the contract manager is responsible for, was provided. User 5 directly found the requested contract in the list, whereas all other test users immediately looked for somewhere to search for the contract. Some minor difficulties to find the search functionality in the client were observed for all users searching for the contract, except from user six.

Task number two asked the user to navigate to the details/analysis page for the service contract found in task one. User 3 navigated to the "Service Contract Analysis" page, using the main menu to the left. This resulted in the analysis page for another contract being opened, and the mistake was not detected by the user, resulting in the task being failed. The "Details" button in the top of the overview page disappeared when the user searched for a contract. This behaviour is built in for lists in the framework and required the user to use the three dots to the left in the list item to navigate further. User 1 struggled to figure out how to do this but managed to complete the task after some time. User 4 opened the analysis page for the requested service contract before even reading the task.

All users completed task three and four without any problems. Although, both User 1 and 4 completed all steps and finished the budget before reading the instructions for the next task.

Task number five asked the user to estimate the yearly occurrences of a service line for two budget periods and turned out to be the task with the lowest effectiveness. User 1 did not figure out what to change and how to do it, resulting in a failed task. User 5 had no problems to change the yearly occurrences, but only did this for one budget period, which did not fulfill the success criteria for that task. All other test users tried to click in the cell for the yearly occurrences and spent some time on finding out how to make the values editable. Despite this difficulty, the task was successfully completed by four out of six test users. User 5 completed this task without even reading task three, four or five. All test users completed task number six without any problems.

The test users were asked to think aloud during the test sessions and some of the major comments from the testers are listed below (in randomized order):

- I can not see the "Contract Id" anywhere so I am not sure this is a budget for the contract NA-100, but I will assume that
5.2. Implementation

- The customers will really appreciate this functionality if it will be in our core application
- I believe the analysis page is great if it can be implemented like this
- The search-functionality is really messy
- I found it hard to find some of the buttons and details
- I felt a bit afraid to do something wrong
- The system is very easy and intuitive
- How do I change the values for the yearly occurrences?!

The results of the SUS questionnaire showed an average score of 89.2, as presented in Table. Question number 9, "I felt very confident using the system", got the lowest average score of 3.2. The same question also indicated a difference between the consultants and the BSAs as groups, where all consultants strongly agreed, but none of the BSAs gave the highest score. Generally, the consultants gave considerably higher scores than the BSAs. The total average score of 89.2 was highly above the average of 68 using the Sauro-Lewis Curved Grading Scale [53] and close to the limit of a ‘superior product’ using the norm values claimed by Bangor, Kortum and Miller [52].

Table 10: SUS evaluation responses from the final usability test

<table>
<thead>
<tr>
<th>Q</th>
<th>User1*</th>
<th>User2</th>
<th>User3</th>
<th>User4</th>
<th>User5</th>
<th>User6</th>
<th>Avr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4(3)</td>
<td>4(3)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>4.5(3.5)</td>
</tr>
<tr>
<td>2</td>
<td>2(3)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1.5(3.5)</td>
</tr>
<tr>
<td>3</td>
<td>4(3)</td>
<td>4(3)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>4.5(3.5)</td>
</tr>
<tr>
<td>4</td>
<td>1(4)</td>
<td>3(2)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1.3(3.7)</td>
</tr>
<tr>
<td>5</td>
<td>4(3)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>4.7(3.7)</td>
</tr>
<tr>
<td>6</td>
<td>2(3)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1.2(3.8)</td>
</tr>
<tr>
<td>7</td>
<td>5(4)</td>
<td>4(3)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>4.5(3.5)</td>
</tr>
<tr>
<td>8</td>
<td>2(3)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1.2(3.8)</td>
</tr>
<tr>
<td>9</td>
<td>3(2)</td>
<td>3(2)</td>
<td>4(3)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>5(4)</td>
<td>4.2(3.2)</td>
</tr>
<tr>
<td>10</td>
<td>2(3)</td>
<td>2(3)</td>
<td>2(3)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1(4)</td>
<td>1.5(3.5)</td>
</tr>
<tr>
<td>SUS</td>
<td>77.5</td>
<td>75</td>
<td>85</td>
<td>100</td>
<td>97.5</td>
<td>100</td>
<td><strong>89.2</strong></td>
</tr>
</tbody>
</table>

*pilot tester, user 1-3: Business Systems Analysts, user 4-6: Consultants
Values without parenthesis: the actual selected values
Values in parenthesis for odd numbers: the selected value minus 1
Values in parenthesis for even numbers: 5 minus the selected value
SUS = 2.5 times the sum of the calculated values in parentheses
Discussion

This chapter includes a discussion about the results and the methods from the different parts of this study. Decisions that might have affected the outcome are highlighted and motivated. The work is put in a wider context and the reliability, validity and replicability are discussed, as well as ethical considerations.

6.1 Results

The results from the different sprints of this study will be discussed and evaluated in this chapter.

6.1.1 Conceptual Design Prototype

The conceptual design prototype was important as a basis for the design process of the detailed design prototype. Creating the conceptual design allowed the flow of the application and the ideas of the core functionality to be evaluated and agreed or denied early in the process. A conceptual design should conform to the mental models of the user [93] and the user tests performed on the conceptual design confirmed that the pre-study had captured the most important requirements and needs of the users.

6.1.2 Detailed Design Prototype

ERP systems contain huge amounts of data and complex business processes that makes information visualization of data important to support high usability for the end user [18]. The analysis page of the detailed design prototype was created to help the user to make a quick assessment of the current and estimated profitability. The choice of graphical displays significantly influences the accuracy of the decisions [37] and according to Quispel, Maes and Schilperoord [38] bar graphs was ranked as the most familiar and with the highest ease of use. This supports the decision of using bar graphs in the analysis page for the detailed design prototype. Moreover, bar graphs enable the possibility to display data over time and
to compare specific amounts [39], which was considered very important for the developed functionality. Displaying the actual values and the budgeted values in the same graph allows the user to instantly detect deviations from the budget and is important to identify risks and improve future estimations and ensure the overall profitability of the contract [15]. Since both the budgeted and actual costs and revenues of a service contract is generated over time, the visual representation was required to support presentation of the data on a timeline. An alternative to bar graphs, to display the values for a time interval, was to use line graphs. A line graph could have improved the ability to detect patterns, but might have decreased the ability to precisely compare the deviations between the budgeted and actual values. This resulted in a combined graph for the overall contract analysis, displaying the margin as a line graph and the costs and revenues as bar graphs.

The functionality to create a budget in the application was designed to guide the user through a number of steps to end up with a completed budget. A user interface should be self-descriptive [45], [46] and the display of all steps in the top lets the user understand what is going on at the moment and what will happen next. This also reduces the short-term memory load, since the user can focus on one task at the time, which is important to support the usability of a system [44]. Although, the responses from the user test were positive, there are some limitations with this design solution. If the user would like to create a budget for many years, the number of steps will increase with the number of years applied. An alternative to reduce this is to use only one step for all budget estimations, which was also the case with the implementation of the high-fidelity prototype.

According to ISO, the users should be able to customize the information in the user interface to suit their individual needs [45], [46]. This could have been considered to a greater extent and there is a lack of customization in the developed detailed prototype. More settings to the analysis page could have been added to make it possible for the user to exclude or include specific information. Customization would also have increased the support for a diversity of age and knowledge of the end users, which is one of the Eight Golden Rules for human-interaction design [44].

6.1.3 Competitor Analysis

The results of the competitor analysis indicated that budget and forecasting of service contracts are not prioritized by ERP vendors. Implementing this functionality would give competitive advantages for IFS, since it is a desired functionality by the user within the service industry. Most likely, this desire applies to this type of customers of other ERP systems as well, since the costs and revenues of service deliveries are in general difficult to estimate [15]. If IFS implements this kind of functionality in the core application, it is important to promote the functionality to make potential customers aware of the business benefits and to clearly distinguish themselves from their competitors.

6.1.4 High-Fidelity Prototype

Implementing the high-fidelity prototype in the framework of IFS Applications caused some restrictions on the design and even though the detailed design prototype was used as a guide for the implementation, some redesign was needed. According to the final usability test, the functionality for estimating the yearly occurrences for different budget periods could have been better. Both how the different budget periods were displayed and how to edit the yearly occurrences caused problems for the test users, and should be subjects for redesign to increase the usability. In the implemented functionality, the user had to click on the edit button (multiple row editing) to be able to change the values of yearly occurrences in the list of services. An alternative was to use single row editing instead, allowing the user to click in the cell containing the value to make it editable, and according to the observed behavior of
6.1. Results

the final user test this would have been a better solution. It would also have been beneficial to use a different color for the editable fields, to direct the attention of the user to the important elements. Displaying the selected budget periods in a separate list turned out to be confusing to the test users. This result was also the concern during the implementation and the decision to keep it this way was only due to restrictions caused by the framework of development. However, a feasible alternative could have been to display one list of services for each budget period. This would have increased the data being displayed to the user at one time and had required a lot of scrolling on the page, although it might have increased the understanding of the different budget periods.

6.1.5 Final Usability Evaluation

The results of the final usability evaluation showed an average task success rate of 92% for all the tasks in the user test. This can be compared to the average task success rate of 78%, assessed through a study presented by Sauro [58], to discuss the usability in terms of effectiveness of the developed functionality. The average task success rate for all the tasks is significantly higher than 78% and the budget functionality can therefore be considered usable in terms of effectiveness. However, evaluating the system task by task indicates that the functionality for estimating the yearly occurrences for different budget periods should be a subject for design improvements. This task got a task success rate of 67%, which is clearly below average. This functionality is also the most important part of the budget and forecasting functionality and is crucial to the accuracy of the created budget. However, one limitation of using the average of 78% as the norm for comparison of the result in this usability evaluation, is that the study performed by Sauro [58] does not provide any information about what kind of systems the data is collected from, which can be considered as a threat to the reliability.

The average score for the SUS questionnaire was assessed to 89.2, which indicates that the overall usability of the system is considerably above the average SUS score of 68 [53]. The SUS questionnaire showed an interesting difference between the BSA and the consultants participating in the usability evaluation. The minimum score assessed by the consultants were 97.5, whilst the BSA gave the evaluated system scores between 75 and 85. Even though no statistical analysis can be applied to determine the differences, there are some potential explanations of the diverging results. All Consultants had been involved in one or several user tests earlier in the process and was somewhat familiar with the concept of the functionality. Also, the consultants work closer to the customers and have a more detailed knowledge about both the desires of the users as well as the process of managing service contracts. On the other hand, the BSAs both have the experience of working with service contracts as well as evaluating and testing functionality in IFS Applications. There is a risk that the consultants rated the usability of the system higher due to being an active part of influencing the design and functionality earlier in the development process, and therefore were biased by their own expectations of the system. The question "I felt very confident using the system" was, evaluated alone, indicating the largest difference between the consultants and the BSA as groups. All consultants strongly agreed with that statement, whilst none of the BSAs gave the highest score. This might also be explained by the fact that the consultants had been evaluating the system before, and therefore remembered how the functionality was intended to work. Finally, it is important to remember that the SUS score is not a percentage, which is a common misconception.

Another limitation of the user test was the restricted test data available in the database. Adding customized data could have highlighted some of the features, for example the progress bar displaying the time elapsed of the contract. Also, the estimations of the yearly occurrences and the preview of the created budget would have benefited from more data on costs, revenues and hours for the different services. An attempt to add customized test data was made, but turned out to be too time consuming to be worth the effort. However, this is
something that should have been prioritized higher if the project was redone to improve the accuracy of the user tests.

6.2 Method

This section critically discusses the methods used in this study to highlight the potential consequences for the results. The replicability, reliability, and validity of this study will be discussed and a section including source criticism is presented. The section is concluded with a discussion of the work in a wider context and suggestions for future work.

6.2.1 Pre-Study

This study was conducted with a qualitative approach, making it impossible to draw any conclusions with statistical significance. Although the study is qualitative, the results can give indications that can be used to answer the research question and draw content specific conclusions. A quantitative study requires a large set of data or test users [48], [50], [95] and was, due to limited resources, never a realistic option for this study. The method used for the literature study is discussed in section 6.2.5 Source Criticism.

One limitation of the requirements elicitation is that no real end users were involved in the interviews. Although, the interviewees work closely with the customers and have a good understanding of the needs of the end users, the participation of some real customers could have increased the accuracy of the created persona. Saiedian and Dale [80] argue that what is most important during the requirements elicitation is not the method used, but the participation of the end users. However, there could be a risk that the requirements would have been too customer specific if a few real end users had been involved, according to the breadth of different customers using IFS Applications. Therefore, people within the company, having the knowledge of the needs of several different types of customers were considered a better choice.

The persona, the scenarios and the user journey created during the requirements elicitation turned out to be a good way of validating and refining the needs and behaviors of the real end users. Showing the mentioned artifacts to the test users made it easy to communicate the requirements and the test user could immediately recognize or deny the persona with the scenarios and the user journey. This is supported by Kaplan [92] who claims that the user journey gives a holistic view of the user experience, is concise and easy to remember. Further, this also proves the aim of the persona: to communicate how groups of users behave, think and what they want to accomplish, and why [91]. Using the method iteratively, definitely increased the chance of prototyping the right functionality early in the process. However, there is still a risk that the persona created is not completely corresponding to the needs and behaviors of the real end users. This risk was reduced by verifying the persona with the test users, which ensured that the persona was valid for at least some real users of the ERP system.

6.2.2 Implementation

The agile approach, with three iterations of development and evaluation and one iteration containing a competitor analysis was successful in this study. Although a plan driven approach has some advantages of being efficient in terms of carrying out projects [71], the agile method increases the chances of developing a system that conforms to the needs and requirements of the end users [74], [75]. The changes in requirements and design that evolved during the design process indicates that a plan driven approach could have resulted in implemented functionality that does not support the users to accomplish what they want to do in the system. Lack of complete requirements is one of the most common reasons for project failures
and the agile approach allowed changes to be made throughout the project, which minimized the risk of incomplete requirements.

As stated by Nam, Childs, and Sohn ([94]) there is a lack of structured design methods to effectively support conceptual design. However, the method based on five human performance factors and described by Parush ([93]) turned out to be successful in the conceptual design process in this study. One drawback of the method, encountered in this project, was that once the task analysis was performed it was hard to produce several different concepts. Using a linked node model, proposed by Nam, Childs, and Sohn ([94]) and described as "a journey towards an unknown destination", might have resulted in more creative solutions. However, the workshop in design thinking, performed in collaboration with the supervisors, collected design ideas from different perspectives and minimized the risk of the conceptual design being limited by the method used.

The major benefits of using the framework of IFS Applications for the implementation turned out to be the possibility to use the existing database and that the developed functionality became consistent with the application it is intended to be used in. "Strive for consistency" is the first rule, and also the most violated one, of the Eight Golden Rules proposed by Shneiderman and Plaisant ([44]). Using the framework eliminated this risk and most likely helped the users to feel familiar with the system. However, there were also some disadvantages, including the inability to implement all design decisions of the detailed design prototype. An alternative was to implement the high-fidelity prototype in another framework, independent of the framework for IFS Application. Nevertheless, this had also caused other disadvantages, such as creating a database that conforms to the structure of actual service contracts and the inability to reuse the code for further development.

Even though this study can be categorized as an applied research, that aims to solve a problem for a specific organization ([95]), the pre-study revealed that the functionality is requested by the customers, whilst the functionality is not provided by ERP vendors, according to the competitor analysis performed in sprint three. However, one major limitation of the competitor analysis was the method used. No literature on methods for conducting comparisons of competing products, in terms of comparing actual functionality, was found. The available literature on competitor analysis, primarily refers to the strengths and weaknesses of a company as a whole and is intended to be used in the area of marketing. The inability to get access to the real products of the competitors limited the research to use Google and the available product specification provided by each competitor, as the primary source. There is a chance that competitors actually do provide some functionality for budget and forecasting of service contracts, that was not found in this study. If any functionality for estimating the total profitability of service contracts had been found in the competitor analysis, new ideas of desired functionality could have been identified and added to the prototype. Although, the competitor analysis was performed systematically, the result is not completely reliable and can only be seen as an indication that functionality for budget and forecasting of service contracts is a low priority for ERP vendors. An option to increase the reliability of the competitor analysis could have been to contact the competitors and ask questions about their functionality for this. However, this would have come with some ethical dilemmas, since this thesis is conducted on behalf of a competitive organization.

### 6.2.3 Evaluation

One of the major limitations in this usability study is that the test users and the interviews used throughout the project are not the real end users. Ritter, Baxter and Churchill mean that the ideal is to use real end users for prototype evaluation. Ritter, Baxter and Churchill ([49]) mean that the ideal is to use real users in this evaluation, but that users of other systems also can be used. Although it was not possible to use real end users in this study, the different samples of the participants in the user tests were selected to represent the end users as well.
6.2. Method

as possible. Moreover, the relative small number of participants in the user tests limits the possibility to draw any statistical significant conclusions and therefore, the results of this study should only be seen as an indication.

The semi-structured interview format, used as the method for evaluating the conceptual and detailed design prototypes, also used by Passera [17] to evaluate the usability of visualization of contracts, was considered successful for this study. A structured interview could have limited the discussion between the interviewer and the test user and important input might have been missed. Structured interviews would have been more suitable for a quantitative study, since it is possible to statistically compare the collected data [82]. The unstructured interview format could have been used, since both semi-structured and unstructured interviews are suitable for qualitative studies [82], but the risk of influencing the answers of the test users would probably have been higher. The semi-structured format kept the focus on the topic, as it also allowed to follow up questions and more discussions with the test users.

The final usability evaluation of the high-fidelity prototype measured the task success rate as well as the perceived overall usability of the system through the SUS questionnaire. According to Ghasemifard et al. [48], all methods have unique benefits and drawbacks and no method is superior to others. Questionnaires were also used in the usability evaluations by both Passera [17] and Jooste, Van Biljon and Mentz [20]. The SUS questionnaire is widely accepted as an essential tool for the overall usability of a system, according to both Lewis [53] and Bangor, Kortum and Miller [52]. Kortum and Peres [57], and Bangor, Kortum and Miller [52] both argue that subjective questionnaires should not be used in isolation to determine the absolute usability of a system. The task success rate was used by Parks [19] to compare the usability of two visual representations of a user interface and Kortum and Peres [57] claims that there is a strong and reliable positive correlation between the SUS score and the effectiveness of a system. This supports the choice of using the task success rate, to evaluate the effectiveness of the implemented functionality, in conjunction with the SUS questionnaire.

It is difficult to estimate how the usability of the high-fidelity prototype would have been assessed if another usability questionnaire had been used instead of the SUS, without testing it. An alternative could have been to use a task-based questionnaire to evaluate the usability of each task. An assumption is that a task-based questionnaire would have given lower scores for the tasks that the individual users struggled with. Although, some of the test users did not struggle with, or even notice, that a task was failed which could have given a misleading result of the score of that task. The effectiveness was estimated using the task success rate measured binary. Two alternatives to this method are to use the task success rate with task levels [55] or to measure the number of errors made by the user [56]. Using task levels would have taken the severity of the task into consideration and the result would probably have been affected by this. Although, task levels provide a more detailed assessment of the completed tasks, binary task success is the simplest and most used method [55]. Task levels could have been a good option to prioritize subjects for design improvements for further development. Measuring the number of errors made by the users would not have been a good method in this study, since most of the errors observed during the test sessions were connected to framework specific functionality (e.g. search functionality and buttons) that could not be modified during the implementation of the functionality.

Since the tasks in the user test only focused on the implemented budgeting functionality, a task-based questionnaire would not have taken the mocked visualizations of the analysis page into account. However, the tasks could have been created to include the graphical visualizations in the analysis page. The lack of tasks for the analysis page can be considered a weakness of the final usability test in this study. Also, the tasks were not as independent as desired, since task four, five and six required task three to be completed. According to Meyer [59], each task should stand alone, and that could be improved in this usability study.
A suggestion is that task 3-6 could have been compiled into only one task. However, this would limit the ability to explicitly test the estimations of different budget periods, since a new budget can be created successfully without any estimations being made at all. Due to the requirements, the system should suggest values for the estimations based on the actual outcome of the previous year, which makes it possible to create the budget without any changes of the values.

6.2.4 Replicability, Reliability, and Validity

To keep a replicability, i.e. that it is possible for someone else to repeat the study by following the method description and compare if the obtained results are similar [65], the method chapter has been written with a high level of details. The questions used for the user tests has been explicitly stated and the form used for the final usability evaluation has been attached in the appendix. Moreover, it is quite likely that the results of the final usability evaluation will be similar if the test is conducted over again on a different day or with a different sample of similar participants. This indicates that the result is reliable [49], [65] in that aspect and that claim is based on the fact the test results did not differ a lot between the participants with the same job titles. One drawback of the qualitative approach, used in this study, is that the results can’t be generalized due to the low experimental validity [49]. Both the reliability and the validity of the study would have been increased if a larger sample of participants had been used in the usability tests. Even though the result is not statistically generalizable, it can be used as an indication.

6.2.5 Source Criticism

The sources used in the literature study are a mixture of peer reviewed journal articles, textbooks and websites. Secondary sources have been used to some extent and the information in the articles has not always been collected for the same purpose as this master’s thesis. To reduce the risk of misinterpretation of the information, triangulation has been used for the major parts of the thesis. Triangulation provides a broader picture of the studied object and is important in qualitative studies, since it can help to improve the validity and reliability of the study [65]. Peer reviewed journal articles have been selected over text books and websites as far as possible. An attempt has been made to use the most recent information found in the area, but some older publications has been used as well. The reason for using as new information as possible, is that the research within software engineering is continually under development.

6.3 The work in a wider context

Ethical considerations must always be taken into account when designing a scientific study [65]. This becomes even more important when dealing with confidential information in an applied research, to solve a specific problem for an organization, and when the study will be published. To avoid revealing confidential information, this master’s thesis has been reviewed by the supervisors at IFS before being published. When conducting user tests, the users should be clearly informed that the test aims to test the system and not the tester, to avoid uncomfortable feelings of the test user. The test users were also told to be honest and not kind, to reduce the risk of the results being affected by observing the test sessions. All test users in this study have volunteered to participate in the tests with the ability to end the test sessions at any point. The test users have also been informed about the purposes of the tests and that no names or other personal data has been mentioned in the results of the tests. The full job titles of the test users have been simplified and the gender and age of the testers have been left out to preserve the integrity of the test users.
6.4 Future Work

Since this study had some limitations of resources, all the desired functionality was not implemented and there are several features left for future work. The analysis page is one of the most important functionalities within the developed business solution and should be implemented to make it possible for the user to get fast and reliable access to the current and estimated profitability of a service contract. Further, the seasonal distribution of the costs and revenues in a budget is a key to increase the accuracy of the budget and reflect the reality. It would be interesting to use predictive analysis as an alternative way of creating a budget, where the data could be based on real outcomes of the service contract or on statistical data derived from the contract type. Service level agreements and penalties related to missed service levels should also be considered to be included, since that is an important part of the service contract and will affect the total profitability. The customers also need the ability to analyze the contract on a more detailed level. Therefore, future development of this kind of solution should include cost and revenue breakdown on object level as well. It could also be useful to use graphical displays to compare different services in relation to each other. If this kind of functionality is actually implemented, it would be interesting to measure if the system improves the accuracy of the decisions regarding the profitability of the contract.
This chapter presents the conclusions being made from this master’s thesis project and answers the research question provided in the introductory chapter.

This master’s thesis aimed to investigate the qualities and characteristics that are required by a system to sufficiently provide financial information on service contracts to satisfy the needs of the end-users by being usable in terms of effectiveness and subjective satisfaction. To fulfill the aim, this thesis intended to answer the following research question: **How can a business solution for follow-up information of service contracts, in an ERP system, be designed to be usable in terms of effectiveness and subjective satisfaction?**

To answer the research question, two of the general key characteristics of an appropriate system for financial follow-up information turned out to be the support of creating a budget and graphical visualizations of both budgeted and actual values. Visualization of the business information gives the user an instant picture of the financial status of the contract, supports the working memory load and improves the ability to be proactive and detect and analyze deviations from the budget. The ability to compare budgeted values with actual outcomes at any point of the contract life cycle is important to improve future estimations and ensure the total profitability of the contract. Moreover, the system should minimize the inputs required by the user and thereby reduce the workload as well as increase the reliability of the analysis of the estimated profitability. It is also important the the user interface is intuitive and structured in a way that supports the workflow of the user. Further, the system should support seasonal distribution of costs and revenues to reflect the reality of the service industry. The business solutions should also include the ability to break down the costs and revenues and analyze certain time periods and specific services to be able to identify the causes of deviations between budgeted and actual values.

The study indicated that there is a general lack of decision support for service contracts in ERP systems, even though it is highly requested by the users active in the service industry. This gives an opportunity of potentially gained market shares for the company providing usable functionality for budget and forecasting of service contracts. The final usability evaluation performed on the high-fidelity prototype developed in this study showed that the system was
highly usable in terms of effectiveness and that the subjective satisfaction was clearly above the average. The conclusion from this is that developed functionality meets the needs of the customers and will increase the ability to make accurate and informed decisions, regarding the profitability of service contracts, and will reduce the risk of making bad decisions due to the lack of complete and reliable information.


Appendices
A.1 Persona - Pre-study

The persona created during the requirements elicitation.
The revised persona updated after the user tests in Sprint 1
## Appendix: Requirements

### B.1 Requirements - Pre-study

<table>
<thead>
<tr>
<th>PRE-STUDY REQUIREMENTS</th>
<th>VER. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. BUDGETING</strong></td>
<td></td>
</tr>
<tr>
<td>- Data for reactive MT from task templates (estimate number)</td>
<td></td>
</tr>
<tr>
<td>- Data for preventive maintenance from PM (get data)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2. INDEXATION</strong></td>
<td></td>
</tr>
<tr>
<td>- Yearly/translation of the costs and revenues of a contract</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3. EXTRAS</strong></td>
<td></td>
</tr>
<tr>
<td>- Budget variances/variants/laws/low case</td>
<td></td>
</tr>
<tr>
<td>- Seasonal distribution of the costs</td>
<td></td>
</tr>
<tr>
<td>- Different versions of the budget</td>
<td></td>
</tr>
<tr>
<td>- Simulation/forecast</td>
<td></td>
</tr>
<tr>
<td>- Update costs on PM and IT from budget</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EXCLUDED:**
- SLA
- Penalties

[86]

Requirements Version 1] The requirements defined from the requirements elicitation
### B.2 Requirements - Sprint 1

#### SPRINT 1 REQUIREMENTS

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| **1. BUDGETING** | - Data for reactive WO from task templates (estimated number)  
- Data for preventive maintenance from PM (project data)  
- Costs: per service and total of the contract  
- Revenues: per service and total of the contract  
- Hours: per service and total of the contract  
- Costs: lowest level: monthly  |
| **2. INDEXATION** | - Yearly indexation of the costs and revenues of a contract |
| **3. INDEXATION** | - Yearly indexation of the costs and revenues of a contract |
| **4. EXTRAS** | - Simulation/forecast  
- Update costs on PM and IT from budget |

EXCLUDED:  
- SLA  
- Penalties

The refined requirements after the user test performed in Sprint 1

### B.3 Requirements - Sprint 2

#### SPRINT 2 REQUIREMENTS

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| **1. CREATE BUDGET** | - Data for reactive WO from task templates (estimated number)  
- Data for preventive maintenance from PM (project data)  
- Costs: per service and total of the contract  
- Revenues: per service and total of the contract  
- Hours: per service and total of the contract  
- Costs: lowest level: monthly |
| **2. BUDGET VARIANCE SAVE BUDGET** | - It should be possible to save different versions of the budget |
| **3. INDEXATION VISUALIZE BUDGET** | - Visualize different versions of the budget in the same screen  
- Compare budgeted values with the actuals  
- Select different time periods (year and month/year/ accounting periods) |
| **4. VISUALIZE BUDGET VARIANCE** | - Visualize yearly indexation for the budget in the same screen  
- Possibility to add a seasonal distribution for the costs and revenues |
| **5. YEARLY INDEXATION** | - Yearly indexation of the costs and revenues of a contract |
| **6. EXTRAS** | - Simulation/forecast  
- Update costs on PM and IT from the budget |

EXCLUDED:  
- SLA  
- Penalties  
- Breakdown on object level

The refined requirements after the user test performed in Sprint 2
Appendix: Product Backlog

C.1 Product Backlog - Pre-study

Product backlog created during the pre-study

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>As a contract manager, I want to make a budget for the costs and revenues of a contract so that I can estimate the profitability of the contract</td>
</tr>
<tr>
<td>1.2</td>
<td>As a contract manager, I want to analyze the budget on a monthly level so that I can see the financial status of each month</td>
</tr>
<tr>
<td>1.3</td>
<td>As a contract manager, I want to see the budgeted costs and revenues for the total of the contract so that I can estimate the total profitability of the contract</td>
</tr>
<tr>
<td>1.4</td>
<td>As a contract manager, I want to see the budgeted costs and revenues for each service of the contract so that I can compare the profitability of different services</td>
</tr>
<tr>
<td>1.5</td>
<td>As a contract manager, I want to see the budgeted hours for the total of the contract so that I can estimate the total hours needed for the contract</td>
</tr>
<tr>
<td>1.6</td>
<td>As a contract manager, I want to see the budgeted hours for each service of the contract so that I can compare the hours needed for different services</td>
</tr>
</tbody>
</table>
Product backlog created during the pre-study

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7</td>
<td>As a contract manager, I want to see the budgeted and actual margin for the contract so that I instantly know the current profitability of the contract</td>
</tr>
<tr>
<td>1.8</td>
<td>As a contract manager, I want to see the actual and budgeted outcomes in the same screen so that I can compare the budget with the reality</td>
</tr>
<tr>
<td>1.9</td>
<td>As a contract manager, I want values for the estimations of the occurrences of the reactive work orders to be prefilled, so that my manual work is reduced</td>
</tr>
<tr>
<td>1.10</td>
<td>As a contract manager, I want to get the data for the preventive maintenance automatically included in my budget so that my workload is reduced</td>
</tr>
<tr>
<td>1.11</td>
<td>As a contract manager, I want an overview of all the contracts that I am responsible for to have an easy access to each contract</td>
</tr>
<tr>
<td>1.12</td>
<td>As a contract manager, I want to see the actual and budgeted margin of each contract in the overview of all the contracts so that I can sort my contracts on profitability</td>
</tr>
<tr>
<td>1.13</td>
<td>As a contract manager, I want to be able to select different time periods for the budget so that I can analyze specific periods</td>
</tr>
<tr>
<td>2.1</td>
<td>As a contract manager, I want the budgeted costs and revenues to have an yearly indexation so that the numbers are accurate</td>
</tr>
<tr>
<td>3.1</td>
<td>As a contract manager, I want to be able to visualize a worst, best and most likely case of my budget so that I can take the budget variance into account for my decisions</td>
</tr>
<tr>
<td>3.2</td>
<td>As a contract manager, I want to apply a seasonal distribution of the costs so that the budget reflects the reality</td>
</tr>
<tr>
<td>3.3</td>
<td>As a contract manager, I want to save my budgets so that I can compare different budgets</td>
</tr>
<tr>
<td>3.4</td>
<td>As a contract manager, I want to simulate a forecast of the expected costs and revenues of a contract to estimate the profitability of the contract</td>
</tr>
<tr>
<td>3.5</td>
<td>As a contract manager, I want to be able to update variables in the contract from the budget tool</td>
</tr>
</tbody>
</table>
# C.2 Product Backlog - Sprint 1

Product backlog updated during Sprint 1

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>As a contract manager, I want to make a budget for the costs and revenues of a contract so that I can estimate the profitability of the contract</td>
</tr>
<tr>
<td>1.2</td>
<td>As a contract manager, I want to analyze the budget on a monthly level so that I can see the financial status of each month</td>
</tr>
<tr>
<td>1.3</td>
<td>As a contract manager, I want to see the budgeted costs and revenues for the total of the contract so that I can estimate the total profitability of the contract</td>
</tr>
<tr>
<td>1.4</td>
<td>As a contract manager, I want to see the budgeted costs and revenues for each service of the contract so that I can compare the profitability of different services</td>
</tr>
<tr>
<td>1.5</td>
<td>As a contract manager, I want to see the budgeted hours for the total of the contract so that I can estimate the total hours needed for the contract</td>
</tr>
<tr>
<td>1.6</td>
<td>As a contract manager, I want to see the budgeted hours for each service of the contract so that I can compare the hours needed for different services</td>
</tr>
<tr>
<td>1.7</td>
<td>As a contract manager, I want to see the budgeted and actual margin for the contract so that I instantly know the current profitability of the contract</td>
</tr>
<tr>
<td>1.8</td>
<td>As a contract manager, I want to see the actual and budgeted outcomes in the same screen so that I can compare the budget with the reality</td>
</tr>
<tr>
<td>1.9</td>
<td>As a contract manager, I want values for the estimations of the occurrences of the reactive work orders to be prefilled, so that my manual work is reduced</td>
</tr>
<tr>
<td>1.10</td>
<td>As a contract manager, I want to get the data for the preventive maintenance automatically included in my budget so that my workload is reduced</td>
</tr>
<tr>
<td>1.11</td>
<td>As a contract manager, I want an overview of all the contracts that I am responsible for to have an easy access to each contract</td>
</tr>
<tr>
<td>1.12</td>
<td>As a contract manager, I want to see the actual and budgeted margin of each contract in the overview of all the contracts so that I can sort my contracts on profitability</td>
</tr>
<tr>
<td>2.1</td>
<td>As a contract manager, I want to be able to visualize a worst, best and most likely case of my budget so that I can take the budget variance into account for my decisions</td>
</tr>
<tr>
<td>2.2</td>
<td>As a contract manager, I want to apply a seasonal distribution of the costs so that the budget reflects the reality</td>
</tr>
</tbody>
</table>
Product backlog updated during Sprint 1

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>As a contract manager, I want to save my budgets so that I can compare different budgets</td>
</tr>
<tr>
<td>2.3</td>
<td>As a contract manager, I want to be able to select different time periods for the budget so that I can analyze specific periods</td>
</tr>
<tr>
<td>3.1</td>
<td>As a contract manager, I want the budgeted costs and revenues to have an yearly indexation so that the numbers are accurate</td>
</tr>
<tr>
<td>4.1</td>
<td>As a contract manager, I want to simulate a forecast of the expected costs and revenues of a contract to estimate the profitability of the contract</td>
</tr>
<tr>
<td>4.2</td>
<td>As a contract manager, I want to be able to update variables in the contract from the budget tool</td>
</tr>
</tbody>
</table>
C.3  Product Backlog - Sprint 2

Product backlog updated during Sprint 2

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>As a contract manager, I want to make a budget for the costs and revenues of a contract so that I can estimate the profitability of the contract</td>
</tr>
<tr>
<td>1.2</td>
<td>As a contract manager, I want to estimate the yearly occurrence of each reactive service in the contract, so that I can estimate the total costs and revenues of the contract</td>
</tr>
<tr>
<td>1.3</td>
<td>As a contract manager, I want values for the estimations of the occurrences of the reactive work orders to be prefilled, so that my manual work is reduced</td>
</tr>
<tr>
<td>1.4</td>
<td>As a contract manager, I want to get the data for the preventive maintenance automatically included in my budget so that my workload is reduced</td>
</tr>
<tr>
<td>1.5</td>
<td>As a contract manager, I want to be able to select a time period for the budget so that I can make the budget for a certain period</td>
</tr>
<tr>
<td>1.6</td>
<td>As a contract manager, I want to get a preview of the total costs, revenues and hours of my budget so that I can accept or change the budget</td>
</tr>
<tr>
<td>1.7</td>
<td>As a contract manager, I want an overview of all the contracts that I am responsible for to have an easy access to each contract</td>
</tr>
<tr>
<td>1.8</td>
<td>As a contract manager, I want to see the actual and budgeted margin of each contract in the overview of all the contracts so that I can sort my contracts on profitability</td>
</tr>
<tr>
<td>2.1</td>
<td>As a contract manager, I want to save my budget so that I can compare different budgets</td>
</tr>
<tr>
<td>3.1</td>
<td>As a contract manager, I want to analyze the budget on a monthly level so that I can see the financial status of each month</td>
</tr>
<tr>
<td>3.2</td>
<td>As a contract manager, I want to analyze the budget on a yearly and total level so that I can see the budgeted profitability of each year and the total of the contract</td>
</tr>
<tr>
<td>3.3</td>
<td>As a contract manager, I want to visualize the budgeted and actual margin for the contract of different time periods so that I can estimate the profitability of the contract</td>
</tr>
<tr>
<td>3.4</td>
<td>As a contract manager, I want to visualize the actual and the budgeted costs and revenues for the total of the contract so that I can estimate the total profitability of the contract</td>
</tr>
<tr>
<td>3.5</td>
<td>As a contract manager, I want to visualize the budgeted and actual costs and revenues for each service of the contract so that I can compare the profitability of different services</td>
</tr>
</tbody>
</table>
Product backlog updated during Sprint 2

<table>
<thead>
<tr>
<th>NO</th>
<th>USER STORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>As a contract manager, I want to visualize the actual and budgeted hours for the total of the contract so that I can estimate the total hours needed for the contract</td>
</tr>
<tr>
<td>3.7</td>
<td>As a contract manager, I want to visualize the actual and budgeted hours for each service of the contract so that I can compare the hours needed for different services</td>
</tr>
<tr>
<td>3.8</td>
<td>As a contract manager, I want to visualize the actual and budgeted outcomes in the same screen so that I can compare the budget with the reality</td>
</tr>
<tr>
<td>4.1</td>
<td>As a contract manager, I want to be able to visualize a worst, best and most likely case of my budget so that I can take the budget variance into account for my decisions</td>
</tr>
<tr>
<td>4.2</td>
<td>As a contract manager, I want to apply a seasonal distribution of the costs so that the budget reflects the reality</td>
</tr>
<tr>
<td>5.1</td>
<td>As a contract manager, I want the budgeted costs and revenues to have an yearly indexation so that the numbers are accurate</td>
</tr>
<tr>
<td>6.1</td>
<td>As a contract manager, I want to simulate a forecast of the expected costs and revenues of a contract to estimate the profitability of the contract</td>
</tr>
<tr>
<td>6.2</td>
<td>As a contract manager, I want to be able to update variables in the contract from the budget tool</td>
</tr>
</tbody>
</table>
Appendix: Conceptual Design - Sprint 1

D.1 Conceptual Design 1

The result of the first conceptual design prototype created during sprint 1
D.2 Conceptual Design 2

The result of the second conceptual design prototype created during sprint 1
Appendix: User Tests - Sprint 1

E.1 Participants

Participants of the user test in Sprint 1

<table>
<thead>
<tr>
<th>USER</th>
<th>TITLE</th>
<th>COUNTRY</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Consultant</td>
<td>Germany</td>
<td>No</td>
</tr>
<tr>
<td>User 2</td>
<td>Senior Advisor</td>
<td>United Kingdom</td>
<td>Yes</td>
</tr>
<tr>
<td>User 3</td>
<td>Consultant</td>
<td>Sweden</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study; No = this is the first time the user is involved in the study
### E.2 Outcome User 1

Summary of the test session with User 1 - Sprint 1

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did you like about this prototype (no 1)?</td>
<td>The overall concept reflects the real process&lt;br&gt;• Graphical and tabular representations&lt;br&gt;• That the system will ask for estimations</td>
</tr>
<tr>
<td>What did you like about this prototype (no 2)?</td>
<td>The overall concept reflects the process and the overview would be great&lt;br&gt;• The overview of all contracts&lt;br&gt;• Graphical and tabular representations&lt;br&gt;• That the system will ask for estimations</td>
</tr>
<tr>
<td>What did you dislike (no 1)?</td>
<td>Nothing special</td>
</tr>
<tr>
<td>What did you dislike (no 2)?</td>
<td>Nothing special</td>
</tr>
<tr>
<td>If you should improve the solutions, what would that be?</td>
<td>Would like to have more prediction. Important with prediction of the future, &quot;best&quot;, &quot;worst&quot; and &quot;most likely&quot; case of the outcome in the same screen</td>
</tr>
<tr>
<td>How would you compare the two solutions?</td>
<td>They are quite similar but I like the overview of the contracts in the second prototype</td>
</tr>
<tr>
<td>Which one worked better for you? Why?</td>
<td>Number 2, since it provides an overview of all the contracts the contract manager is responsible for</td>
</tr>
</tbody>
</table>
### E.3 Outcome User 2

Summary of the test session with User 2 - Sprint 1

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did you like about this prototype (no 1)?</td>
<td>The overall concept is good</td>
</tr>
<tr>
<td>What did you like about this prototype (no 2)?</td>
<td>Same as above</td>
</tr>
<tr>
<td>What did you dislike (no 1)?</td>
<td>Nothing special</td>
</tr>
<tr>
<td>What did you dislike (no 2)?</td>
<td>Unnecessary with an overview of all the contracts</td>
</tr>
<tr>
<td>If you should improve the solutions, what would that be?</td>
<td>The estimation needs to be well considered since there will be a huge number of estimates to be done</td>
</tr>
<tr>
<td>How would you compare the two solutions?</td>
<td>The overall process is similar in both</td>
</tr>
<tr>
<td>Which one worked better for you? Why?</td>
<td>Number 1, since it is unnecessary with the overview</td>
</tr>
</tbody>
</table>

### E.4 Outcome User 3

Summary of the test session with User 3 - Sprint 1

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>What did you like about this prototype (no 1)?</td>
<td>The overall concept reflects the needs of the user</td>
</tr>
<tr>
<td>What did you like about this prototype (no 2)?</td>
<td>Same as above and the overview is good</td>
</tr>
<tr>
<td>What did you dislike (no 1)?</td>
<td>Nothing special</td>
</tr>
<tr>
<td>What did you dislike (no 2)?</td>
<td>Nothing special</td>
</tr>
<tr>
<td>If you should improve the solutions, what would that be?</td>
<td>Would like to be able to analyze different time periods in the detailed overview of the contract. It is also important with prediction of the future and a &quot;best&quot;, &quot;worst&quot; and &quot;most likely&quot; case of the outcome</td>
</tr>
<tr>
<td>How would you compare the two solutions?</td>
<td>No big differences</td>
</tr>
<tr>
<td>Which one worked better for you? Why?</td>
<td>Number 2, since it provides an easy overview of all the contracts</td>
</tr>
</tbody>
</table>
Appendix: Detailed Prototype - Sprint 2

F.1 Detailed Prototype - Before Evaluation

Start Page

The first version of the starting page of the detailed prototype for the budgeting functionality.
Contract Analysis - Yearly View

The first version of the contract analysis page of the detailed prototype for the budgeting functionality. The figure shows the yearly view of the analysis.
The first version of the contract analysis page of the detailed prototype for the budgeting functionality. The figure shows the monthly view of the analysis.
New Budget - Settings

The first version of the budget settings page of the detailed prototype for the budgeting functionality

New Budget - Estimations

The first version of the budget estimations page of the detailed prototype for the budgeting functionality
The first version of the budget preview page of the detailed prototype for the budgeting functionality.
The redesigned version of the starting page of the detailed prototype for the budgeting functionality
The redesigned version of the contract analysis page of the detailed prototype for the budgeting functionality. The figure shows the yearly view of the analysis.
Redesigned Prototype: Contract Analysis Page - Yearly View Continuation
Contract Analysis - Monthly View

The redesigned version of the contract analysis page of the detailed prototype for the budgeting functionality. The figure shows the monthly view of the analysis.
F.2. Redesigned Prototype - After Evaluation

Redesigned Prototype: Contract Analysis Page - Monthly View Continuation
New Budget - Settings

The redesigned version of the budget settings page of the detailed prototype for the budgeting functionality

New Budget - Estimations

The redesigned version of the budget estimations page of the detailed prototype for the budgeting functionality
New Budget - Preview

The redesigned version of the budget preview page of the detailed prototype for the budgeting functionality
G.1 Participants - User Test

Participants of the user test in Sprint 2

<table>
<thead>
<tr>
<th>USER</th>
<th>TITLE</th>
<th>COUNTRY</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Consultant</td>
<td>Germany</td>
<td>Yes</td>
</tr>
<tr>
<td>User 2</td>
<td>Consultant</td>
<td>United Kingdom</td>
<td>Yes</td>
</tr>
<tr>
<td>User 3</td>
<td>Consultant</td>
<td>Germany</td>
<td>No</td>
</tr>
<tr>
<td>User 4</td>
<td>Product Solution Manager</td>
<td>Sweden</td>
<td>No</td>
</tr>
<tr>
<td>User 5</td>
<td>Consultant</td>
<td>Sweden</td>
<td>No</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study, No = this is the first time the user is involved in the study.
G.2 Participants - Heuristic Evaluation

Participants of the group evaluation in Sprint 2

<table>
<thead>
<tr>
<th>TITLE</th>
<th>COUNTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Engineer</td>
<td>Sweden</td>
</tr>
<tr>
<td>Product Architect</td>
<td>Sweden</td>
</tr>
<tr>
<td>Senior Business Systems Analyst</td>
<td>Sweden</td>
</tr>
</tbody>
</table>

Participants of the final usability test in Sprint 4

<table>
<thead>
<tr>
<th>TITLE</th>
<th>COUNTRY</th>
<th>EXPERIENCE</th>
<th>INVOLVED EARLIER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>1-5 years</td>
<td>No</td>
</tr>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>1-5 years</td>
<td>No</td>
</tr>
<tr>
<td>Business Systems Analyst</td>
<td>Sweden</td>
<td>+ 10 years</td>
<td>No</td>
</tr>
<tr>
<td>Consultant</td>
<td>Germany</td>
<td>5-10 years</td>
<td>Yes</td>
</tr>
<tr>
<td>Consultant</td>
<td>Germany</td>
<td>1-5 years</td>
<td>Yes</td>
</tr>
<tr>
<td>Consultant</td>
<td>UK</td>
<td>5-10 years</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Yes = the user has been involved in an earlier stage of this study, No = this is the first time the user is involved in the study

G.3 Mapping of Questions

Mapping of Questions and Prototype Pages

<table>
<thead>
<tr>
<th>Page</th>
<th>1. What did you like about this page? Why?</th>
<th>2. What did you dislike about this page? Why?</th>
<th>3. If you should improve this page, what would that be?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Page</td>
<td>1.1</td>
<td>1.2</td>
<td>1.3</td>
</tr>
<tr>
<td>Contract Analysis</td>
<td>2.1</td>
<td>2.2</td>
<td>2.3</td>
</tr>
<tr>
<td>Settings</td>
<td>3.1</td>
<td>3.2</td>
<td>3.3</td>
</tr>
<tr>
<td>Estimations</td>
<td>4.1</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td>Preview</td>
<td>5.1</td>
<td>5.2</td>
<td>5.3</td>
</tr>
</tbody>
</table>
### G.4 Outcome User 1

Summary of the test session with User 1 - Sprint 2

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Good with an overview of all the contracts that the contract manager is responsible for. Like the ability to instantly see the actual and budgeted margin for all your contracts since that will help the manager to identify contracts that need further analysis.</td>
</tr>
<tr>
<td>1.2</td>
<td>The page is an okay start page, but it could be made a bit more interesting to the user.</td>
</tr>
<tr>
<td>1.3</td>
<td>Maybe add some graphical elements. Would like to be able to sort on customer, expiry date and margin as well.</td>
</tr>
<tr>
<td>2.1</td>
<td>The overall impression is good.</td>
</tr>
<tr>
<td>2.2</td>
<td>Would like the service breakdown to be on object level as well. The breakdown should be done both yearly and monthly for the total of the whole contract, each object and each service. It is important for the customer to be able to analyze on object level.</td>
</tr>
<tr>
<td>2.3</td>
<td>The cost analysis and the revenue analysis should also be on more detailed levels, such as per service and per object.</td>
</tr>
<tr>
<td>3.1</td>
<td>Like that the system will suggest values and that the user has the ability to select how the values should be estimated.</td>
</tr>
<tr>
<td>3.2</td>
<td>Get a bit confused whether the budget is for the whole contract or if it is for one object.</td>
</tr>
<tr>
<td>3.3</td>
<td>Clearify that the budget is for the total of the contract.</td>
</tr>
<tr>
<td>4.1</td>
<td>Like that the values are prefilled because it reduce the inputs required by the user. Agree that the estimations should be done on a yearly basis.</td>
</tr>
<tr>
<td>4.2</td>
<td>Nothing.</td>
</tr>
<tr>
<td>4.3</td>
<td>Would like to have information about the number of objects that are connected to each service.</td>
</tr>
<tr>
<td>5.1</td>
<td>Like the possibility to change the estimations made in the previous step.</td>
</tr>
<tr>
<td>5.2</td>
<td>Don not like to see the actuals at this page. Only interested to see the future here.</td>
</tr>
<tr>
<td>5.3</td>
<td>Would like to have a more detailed service breakdown here as well. Like the ability to compare the budget with the actuals, but this should be done in another place in the business solution.</td>
</tr>
</tbody>
</table>
## G.5 Outcome User 2

Summary of the test session with User 2 - Sprint 2

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Like the overview and believe that the most important information is present here</td>
</tr>
<tr>
<td>1.2</td>
<td>Would like to see the customer ID as well, and would rather have it sorted on the customer</td>
</tr>
<tr>
<td>1.3</td>
<td>Add the ability to group the contract by customer</td>
</tr>
<tr>
<td>2.1</td>
<td>Like the graphical presentation because it helps the user to get an instant overview of the contract</td>
</tr>
<tr>
<td>2.2</td>
<td>Would like to split the costs and revenues on PM and Reactive as well. The service breakdown gets quite messy.</td>
</tr>
<tr>
<td>2.3</td>
<td>Could be helpful with some help for the user to select the time period, for example a button for &quot;last year&quot; etc. Would like the breakdown from the costs, revenues and services to be on object level as well. Rolling years should be used since the accounting period is not always from January to December. Would like to be able to switch the graphs to a pie chart and present the data as a percentage of the total</td>
</tr>
<tr>
<td>3.1</td>
<td>The page is okay and easy</td>
</tr>
<tr>
<td>3.2</td>
<td>Do not understand the exact meaning of &quot;Template&quot;</td>
</tr>
<tr>
<td>3.3</td>
<td>(User 2 asked for the seasonal distribution at this stage)</td>
</tr>
<tr>
<td>4.1</td>
<td>The possibility to add a seasonal distribution is really important</td>
</tr>
<tr>
<td>4.2</td>
<td>Miss the number of object connected to each service here</td>
</tr>
<tr>
<td>4.3</td>
<td>Add a column with the number of objects connected to each service. Could also be useful to add a percentage to a previous budget and use that as the suggestions for the estimations</td>
</tr>
<tr>
<td>5.1</td>
<td>Good with a graphical overview of the budget</td>
</tr>
<tr>
<td>5.2</td>
<td>Only need to see the budgeted values at this stage</td>
</tr>
<tr>
<td>5.3</td>
<td>Would like to be able to analyze the budget on object level or asset class as well. It would be good to be able to compare the budget with a previous budget. Could be useful to have something that represents the time left of the contract in the top of the page (for all pages actually). An approval process could be required before the budget is activated. This could be a check-box or something that tells that the budget needs to be approved by a manager before it is activated</td>
</tr>
</tbody>
</table>
### G.6 Outcome User 3

Summary of the test session with User 3 - Sprint 2

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Like the search functionality and the ability to group or sort the contracts</td>
</tr>
<tr>
<td>1.2</td>
<td>The screen is quite boring and could be more interesting to the user</td>
</tr>
<tr>
<td>1.3</td>
<td>Would like to see the contract type here as well and have the ability to sort on the contract type. Some graphical presentations could be added to make it more interesting. Maybe a picture of the contract manager could be in the top. A button for the lowest budgeted margin and highest budgeted margin with a visual display could be added as well</td>
</tr>
<tr>
<td>2.1</td>
<td>Think this is very nice and the customer will be very happy to have this functionality.</td>
</tr>
<tr>
<td>2.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>2.3</td>
<td>It could be useful for some customers to use rolling years since the accounting year may differ between the companies. Do not think is necessary to drill down the cost and revenue analysis or the service breakdown on object level</td>
</tr>
<tr>
<td>3.1</td>
<td>Good to select the basis for the suggestions of the estimations</td>
</tr>
<tr>
<td>3.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>3.3</td>
<td>No suggestions</td>
</tr>
<tr>
<td>4.1</td>
<td>This part is complicated but this is a good solution for now</td>
</tr>
<tr>
<td>4.2</td>
<td>If the company have 200+ services this can be more difficult</td>
</tr>
<tr>
<td>4.3</td>
<td>It is more difficult in reality, but in the same time it has to be easy for the user. Could make it more obvious that each estimation is for only one object</td>
</tr>
<tr>
<td>5.1</td>
<td>Like that both the actuals and the budgeted values are displayed here so that a comparison could be done. Good to be able to change the values of the estimations and update the graphical displays to see how the result is affected</td>
</tr>
<tr>
<td>5.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>5.3</td>
<td>Could make it more clear to the user that the estimations here are the total number of occurrences for all objects connected to each service</td>
</tr>
</tbody>
</table>
# Outcome User 4

Summary of the test session with User 4 - Sprint 2

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>It is an okay overview</td>
</tr>
<tr>
<td>1.2</td>
<td>Do not like the repeating &quot;open&quot; button</td>
</tr>
<tr>
<td>1.3</td>
<td>Instead of the open-button, a multi-selection button could be used in the top</td>
</tr>
<tr>
<td>2.1</td>
<td>Good to be able to break down everything on a monthly basis as well</td>
</tr>
<tr>
<td>2.2</td>
<td>Would prefer to have a total, and then be able to zoom in into different time periods</td>
</tr>
<tr>
<td>2.3</td>
<td>Would like the costs to be broken down into cost types as well. The budgeted and actual values could be displayed closer together and red and green should be avoided together due colorblindness</td>
</tr>
<tr>
<td>3.1</td>
<td>Okay, easy to understand</td>
</tr>
<tr>
<td>3.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>3.3</td>
<td>Nothing</td>
</tr>
<tr>
<td>4.1</td>
<td>Okay</td>
</tr>
<tr>
<td>4.2</td>
<td>It might be a bit confusing to estimate on one year and on only one object. All objects are not having the same amount of services and a budget is usually made for the whole contract period. Maybe the &quot;budget&quot; is actually a forecast?</td>
</tr>
<tr>
<td>4.3</td>
<td>Could be useful to see the estimation for the preventive maintenance here as well, even if they should not be able to be changed</td>
</tr>
<tr>
<td>5.1</td>
<td>Like the table where the totals are displayed and the ability to change the values here</td>
</tr>
<tr>
<td>5.2</td>
<td>Only want to see the budgeted values here</td>
</tr>
<tr>
<td>5.3</td>
<td>Would like to be able to compare different budget versions</td>
</tr>
</tbody>
</table>
## G.8 Outcome User 5

Summary of the test session with User 5 - Sprint 2

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Good to have an overview of all the service contracts</td>
</tr>
<tr>
<td>1.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>1.3</td>
<td>Sometimes there is no end-date of a contract and this needs to be considered if the start and end date will be displayed here</td>
</tr>
<tr>
<td>2.1</td>
<td>Good information</td>
</tr>
<tr>
<td>2.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>2.3</td>
<td>Each service may have sub-services where the actual costs arise. In this case, it would be interesting to have the breakdown on these child-services since only the invoice plan is on the parent row in the contract</td>
</tr>
<tr>
<td>3.1</td>
<td>Easy to understand, nice to have the budget creation in an assistant</td>
</tr>
<tr>
<td>3.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>3.3</td>
<td>Nothing</td>
</tr>
<tr>
<td>4.1</td>
<td>Seems good</td>
</tr>
<tr>
<td>4.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>4.3</td>
<td>Sometimes an object has an object structure, i.e. one object is connected to the service but the service is executed on each sub-object. This will affect the occurrences of the service. You might want to add column for the number of objects as well as a column for the number of sub-objects in the structure to be able to make more accurate estimations</td>
</tr>
<tr>
<td>5.1</td>
<td>Like that the actuals are displayed here</td>
</tr>
<tr>
<td>5.2</td>
<td>Nothing</td>
</tr>
<tr>
<td>5.3</td>
<td>A budget could automatically be generated when a contract is created. This would just be another entry point and is probably outside the scope of this thesis project</td>
</tr>
</tbody>
</table>
G.9 Outcome Heuristic Evaluation

Summary of the heuristic evaluation - Sprint 2

<table>
<thead>
<tr>
<th>SCREEN</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Need to display a scenario where no budget is made</td>
</tr>
<tr>
<td>PAGE</td>
<td>Change the heading to “My service contract analysis”</td>
</tr>
<tr>
<td>CONTRACT</td>
<td>Do not display the numbers for the margin in the graph</td>
</tr>
<tr>
<td>ANALYSIS</td>
<td>Would like to know if the service is a PM or a reactive service</td>
</tr>
<tr>
<td></td>
<td>Cost analysis and revenue analysis on separate lines</td>
</tr>
<tr>
<td></td>
<td>Add the total costs, revenues and hours for the selected time period</td>
</tr>
<tr>
<td></td>
<td>Restrict the monthly view to only display 12 months</td>
</tr>
<tr>
<td></td>
<td>Move the view button to the top</td>
</tr>
<tr>
<td></td>
<td>Use line graphs instead of bar graphs for the service breakdowns,</td>
</tr>
<tr>
<td></td>
<td>the users are most interested in the trends</td>
</tr>
<tr>
<td></td>
<td>The first screen can show the yearly overview and then the user</td>
</tr>
<tr>
<td></td>
<td>could click in the graph to show the details</td>
</tr>
<tr>
<td></td>
<td>Split revenues in periodic, reactive and fixed price revenues</td>
</tr>
<tr>
<td></td>
<td>Split costs in investment costs, preventive costs and reactive costs</td>
</tr>
<tr>
<td></td>
<td>Display the hours in a graph with comparable numbers</td>
</tr>
<tr>
<td>BUDGET</td>
<td>Rename &quot;Actual&quot; to &quot;Last year outcome&quot;</td>
</tr>
<tr>
<td>SETTINGS</td>
<td>Clarify &quot;Template&quot;, but this is not a priority</td>
</tr>
<tr>
<td>BUDGET</td>
<td>Remove material, quantity, work type and TTID</td>
</tr>
<tr>
<td>ESTIMATIONS</td>
<td>Add task template description, total service cost, total service revenue</td>
</tr>
<tr>
<td></td>
<td>Rename &quot;Service&quot; to &quot;Service Contract Line&quot;</td>
</tr>
<tr>
<td></td>
<td>Add one step in the assistant for each year, since the budgeted values</td>
</tr>
<tr>
<td></td>
<td>might vary between the years</td>
</tr>
<tr>
<td></td>
<td>Add the distribution to each line to be able to use different distributions for different services</td>
</tr>
<tr>
<td>BUDGET</td>
<td>Remove the graphs from this screen, use only a table with the total values for the budget</td>
</tr>
<tr>
<td>PREVIEW</td>
<td>The values should not be changeable at this stage</td>
</tr>
<tr>
<td></td>
<td>Rename the &quot;Accept&quot;-button to &quot;Finish&quot;</td>
</tr>
<tr>
<td>OTHER</td>
<td>It should be possible to change a budget</td>
</tr>
<tr>
<td>INPUT</td>
<td>Would like to see the total margin per service and the total</td>
</tr>
</tbody>
</table>
H.1  Service Contracts Overview Page

Implemented page for the service contracts overview displayed as a list
H.2 New Budget - Settings

Implemented page for the service contracts overview displayed as cards

The first page of creating a new budget for a service contract containing the main settings for the budget
H.3 New Budget - Estimations

The second step page of creating a new budget for a service contract containing the generated periods, service lines and estimations for the budget.

H.4 New Budget - Preview

The final step of creating a new budget for a service contract containing a summary of the generated budget information.
The final analysis page for a service contract, displaying budgeted and actual values of the contract. Note that the functionality is not implemented but the images demonstrate how the data can be displayed.
Appendix: User Tests - Sprint 4

I.1 Form - User Test
Budget & Forecasting of Service Contracts - User Test

This usability test is a part of my master's thesis project for budget and forecasting of service contracts. The aim of this user test is to evaluate the usability of the system in terms of effectiveness, i.e. that the user can do what the user wants to do. The test is completely voluntary and evaluates the system and not you as a test user. Individual answers will be anonymized before being published in the thesis. Thank you for participating!

*Required

Personal Information

1. Country of Employment *

2. Job Title *

3. Experience of working with Service Contracts (in any way) *
   Mark only one oval.
   - No experience
   - Less than 1 year
   - Between 1 and 5 years
   - Between 5 and 10 years
   - More than 10 years

Tasks

Try to complete the following tasks in the system

4. TASK 1: Find the service contract with the ID 'NA-100' *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ________________________________

5. TASK 2: Navigate to the details/analysis page for the service contract found in TASK 1 *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ________________________________
6. TASK 3: At the provided page, initiate a new budget for that service contract *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ____________________________

7. TASK 4: Give the initiated budget a name of your choice and set the budget period from 2018-06-01 to 2020-06-01 *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ____________________________

8. TASK 5: Find the service row with the service line number ‘1’. Make an estimation of ‘5’ yearly occurrences for that service for budget period no 1. Make an estimation of ‘10’ yearly occurrences for that service for budget period no 2 *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ____________________________

9. TASK 6: Finish the budget you have created *
   Mark only one oval.
   - I believe I have completed the task
   - I could not complete the task
   - Other: ____________________________

SUS Evaluation
The System Usability Scale includes 10 questions and is used to measure the overall usability of a system. Imagine that you are a contract manager, responsible for 2 large customers and all their service contracts, when answering the following questions.

System Usability Scale Questionnaire

10. I think that I would like to use this system frequently *
    Mark only one oval.
    [ ] 1 2 3 4 5
    Strongly Disagree  [ ]  [ ]  [ ]  [ ]  [ ]  Strongly Agree

11. I found the system unnecessarily complex *
    Mark only one oval.
    [ ] 1 2 3 4 5
    Strongly Disagree  [ ]  [ ]  [ ]  [ ]  [ ]  Strongly Agree
12. I thought the system was easy to use *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

13. I think that I would need the support of a technical person to be able to use this system *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

14. I found the various functions in this system were well integrated *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

15. I thought there was too much inconsistency in this system *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

16. I would imagine that most people would learn to use this system very quickly *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

17. I found the system very cumbersome (besvärligt) to use *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree

18. I felt very confident using the system *
   *Mark only one oval.
   
   1 2 3 4 5
   
   Strongly Disagree ○ ○ ○ ○ ○ Strongly Agree
19. I needed to learn a lot of things before I could get going with this system *

*Mark only one oval.*

1 2 3 4 5

Strongly Disagree ☐ ☐ ☐ ☐ ☐ Strongly Agree