Examensarbete

A Comparative Evaluation Between Two Design Solutions for an Information Dashboard

av

Lovisa Gannholm

LIU-IDA/LITH-EX-A--13/058--SE

2013-11-01
Examensarbete

A Comparative Evaluation Between Two Design Solutions for an Information Dashboard

av

Lovisa Gannholm

LIU-IDA/LITH-EX-A--13/058--SE

2013-11-01

Handledare: Johan Blomkvist, IDA, Linköpings universitet
Examinator: Johan Åberg, IDA, Linköpings universitet
Abstract

This study is a software usability design case about information presentation in a software dashboard. The dashboard is supposed to present system information about an enterprise resource planning system. The study aims to evaluate if the intended users of the dashboard prefer a list-based or an object-based presentation of the information and why. It also investigates if the possibility to get familiar with the prototype affects the evaluation’s result.

The study was performed using parallel prototypes and evaluation with users. The use of parallel prototypes is a rather unexplored area. Likewise, little research has been done in the area of how user experience changes over time.

Two prototypes were created, presenting the same information in two different design solutions, one list-based, and one object-based. The prototypes were evaluated with ten presumptive users, with respect to usability. The evaluation consisted of two parts, one quantitative and one qualitative. Half of the respondents got a chance to get familiar with the list-based prototype, and half the object-based prototype, after which they evaluated both sequentially.

The result of the evaluation showed that seven out of ten respondents preferred the list-based prototype. The two primary reasons were that they are more used to the list-based concept from their work, and that the list-based prototype presented all information about an application at once. In the object-based prototype the user had to make a request for each type of information, which opened up in a new pop-up window.

The primary reason that three of the ten respondents preferred the object-based prototype was that it had a more modern look, and gave a cleaner impression since it only presented the information the respondent was interested in at each point in time.

The result also implied that the possibility to get familiar with the prototype by testing it for a couple of days affected the result. Eight out of ten respondents preferred the prototype they got familiar to, and the only ones that liked or preferred the object-based prototype were those who had gotten familiar with it.

The results of the study support the results of the existing research done by Dow et al. (2010) on the use of parallel prototypes, i.e. creating several prototypes in parallel, and conform with the results of the research of Karapanos et al. (2009) on how user experience changes over time.

Some other interesting information that emerged from the study was that all but one of the respondents thought that the prototype they got familiar with had an acceptable level of usability.

The study also validated that all respondents are positive to use a dashboard in their work, and that the presented information was enough for a first version of the dashboard. It also validated that the different groups of users would use the dashboard differently, and therefore are in need of slightly different information.
Acknowledgements

When I got employed at Infor in January 2013, I still had not written my master thesis. My team leader Eva Kallerman, the director of M3 Technology Per Ehnsiö, and Håkan Lundh, who is a business analyst for related components, wanted to evaluate the need of an information dashboard for the M3 system, and what kind of information the users find relevant to view in such a dashboard. The task suited me well since it gave me the possibility to get an understanding for both the products in M3, especially the LCM and the Grid, and for the users. The difficult part was to find a suitable academic orientation for the thesis. Since I like programming I chose to focus on prototyping and design, even if I had little knowledge in these areas. Unfortunately I never got the time to implement the prototypes during the thesis, but I have learned a lot about prototyping, usability, and evaluations.

I would like to thank Infor, especially Eva Kallerman and Per Ehnsiö, for giving me the possibility to perform this study and thereby graduate. In addition I would like to thank Håkan Lundh, who has been responsible for the task, my mentor Ulf Karlsson, who has given me support all the way, and everyone else at Infor who has contributed with their time and knowledge to this study. I would also like to thank my supervisor Johan Blomkvist for his guidance during this long journey.

Finally I would like to thank my family and friends, for their support during my studies. I would especially like to thank my fiancé who has stood by me the whole time; your support has meant a lot to me.

Linköping, November 2013

Lovisa Gannholm
# Table of Contents

1. **Introduction**........................................................................................................... 1
   1.1. The problem........................................................................................................... 1
   1.2. The task................................................................................................................ 2
   1.3. Thesis purpose & research questions...................................................................... 2
   1.4. Definitions............................................................................................................. 3
   1.5. Disposition............................................................................................................ 4

2. **Background**............................................................................................................ 5
   2.1. The enterprise resource planning system............................................................. 5
   2.2. The management tools......................................................................................... 5
   2.3. The users............................................................................................................... 8
      2.3.1. Services......................................................................................................... 8
      2.3.2. Support......................................................................................................... 8
      2.3.3. Demo Environment....................................................................................... 8
      2.3.4. Sales............................................................................................................ 8
      2.3.5. External customers....................................................................................... 9

3. **Theory**................................................................................................................... 11
   3.1. Dashboards........................................................................................................... 11
   3.2. Human perception............................................................................................... 11
   3.3. Prototypes............................................................................................................ 13
      3.3.1. Attributes/characteristics of a prototype....................................................... 14
      3.3.2. Low- and high-fidelity................................................................................ 15
      3.3.3. Mixed-fidelity.............................................................................................. 15
      3.3.4. Parallel prototyping...................................................................................... 16
   3.4. Evaluation............................................................................................................ 16
      3.4.1. What to evaluate........................................................................................... 17
      3.4.2. Usability........................................................................................................ 17
      3.4.3. System usability scale.................................................................................. 17

4. **Method**................................................................................................................... 19
   4.1. Type of study........................................................................................................ 19
   4.2. The design process............................................................................................. 19
   4.3. The evaluation process....................................................................................... 21
      4.3.1. Evaluation approaches................................................................................ 21
4.3.2. The chosen approach ................................................................. 21
4.3.3. Evaluation method ..................................................................... 22
4.3.4. Triangulation ............................................................................. 22
4.3.5. Selection of respondents............................................................. 22

5. **Execution** ....................................................................................... 25
5.1. The properties of the dashboard ................................................... 25
5.2. The design process ......................................................................... 26
5.2.1. The prototypes ............................................................................ 26
5.2.2. The prototypes’ attributes............................................................ 28
5.2.3. Summary....................................................................................... 29
5.3. The resulting prototypes ................................................................. 29
5.3.1. The design concept ..................................................................... 32
5.3.2. The function & content ................................................................. 32
5.3.3. The structure ............................................................................... 33
5.3.4. The interaction ............................................................................ 33
5.3.5. The presentation ........................................................................... 34
5.4. The evaluation process .................................................................... 35
5.4.1. The selected respondents ............................................................. 35
5.4.2. How the evaluations were conducted ........................................ 35
5.4.3. Background questions ................................................................. 36
5.4.4. System usability scale ................................................................. 36
5.4.5. The interviews ............................................................................. 36

6. **Result** ........................................................................................... 39
6.1. The quantitative results ................................................................. 39
6.2. Analysis of the quantitative result .................................................. 40
6.3. The qualitative result ...................................................................... 41
6.3.1. Why some prefer the list prototype .......................................... 42
6.3.2. Why some prefer the object prototype ..................................... 42
6.3.3. The use of a dashboard ............................................................... 42
6.3.4. The different user groups ........................................................... 45
6.4. Suggested improvements ................................................................ 46

7. **Discussion** ..................................................................................... 51
7.1. Do the users prefer a list-based or an object-based dashboard? .... 51
7.2. How does the possibility to get familiar with the prototype before the evaluation affect the result? ............................................................. 52
7.3. Would the respondents like to use any of the prototypes in their work? .............. 53
7.4. The different user groups ........................................................................ 53
7.5. Do the users feel that any of the two prototypes has an acceptable usability? .......... 54
7.6. Did the prototypes fulfill their purpose? ......................................................... 54
7.7. Recommendation on suggested improvements to implement......................... 55
7.8. Reliability & generalizability of the result....................................................... 56
7.9. Future work .................................................................................................. 56

8. Conclusion ........................................................................................................ 59
   Appendix A. The evaluation methods ................................................................. 61
      A.1. Questions for the background information about the respondent ................. 61
      A.2. The statements in the System usability scale .............................................. 61
      A.3. The interview questions about the prototype’s usability ............................ 61
      A.4. The information sent to the respondents .................................................. 62
   Appendix B. The prototypes .............................................................................. 65
      B.1. Images of the list prototype ................................................................... 65
      B.2. Images of the object prototype ............................................................... 76
   References ......................................................................................................... 93
List of Figures
Figure 2.1: A model of an M3 Solution.......................................................................................................................... 6
Figure 2.2: Image of the Life Cycle Managers user interface.......................................................................................... 7
Figure 3.1: The steps of useful dashboard design, by Few (2013, p. 98)........................................................................ 20
Figure 5.1: Image of the list prototype.......................................................................................................................... 30
Figure 5.2: Image of the object prototype....................................................................................................................... 31

List of Tables
Table 6.1: The result from respondents in group 1........................................................................................................... 39
Table 6.2: The result from respondents in group 2........................................................................................................... 40
Table 6.3: The grading of statement 1 for all respondents................................................................................................. 40
Table 6.4: The values connected to the first research question, calculated from the result of SUS......................................................... 41
Table 6.5: The values connected to the first research question, calculated from the result of the interviews................................. 41
Table 6.6: The values connected to the second research question, calculated from the result of SUS......................................................... 41
Table 6.7: The values connected to the second research question, calculated from the result of the interviews......................... 41
1. Introduction

This chapter gives an introduction to the study, explaining both the problem, and how the study will be conducted. It starts with an introduction to dashboards and to the company. Then it explains the problem, the task that is meant to solve it, what tools will be used, and defines the questions this study is meant to answer. Lastly it lists some useful definitions, and the limitations of the study.

Dashboards are often associated with machinery where they present information for the machine operator. In recent years business leaders have also started to use business dashboards as an auxiliary tool when managing the company. Dashboards are designed to show relevant information in an accessible manner so that the user can quickly identify and respond to problems in order to improve organizational performance (Rasmussen et al., 2009). This can be generalized to a dashboard, which shows other relevant information such as information of an IT system. This information is relevant for the system administrators of large, complex IT systems.

One such IT system developer is Infor which, among other things, develops Enterprise Resource Planning systems, also known as ERP systems. One of their ERP systems is M3. The development teams responsible for M3 are primarily located in Linköping, Stockholm, and Manila. One of these teams is M3 Technology, responsible for many of the tools that are used to manage the system, i.e. install, configure, customize, and run M3. Two of these tools are the Life Cycle Manager, also known as the LCM, and the Grid. The LCM handles the installation and configuration of M3. The Grid handles many of the products and components in the system during runtime. All the tools contain much information about the installed system. The LCM contains primary static information while the Grid contains runtime information.

This information is used by both internal and external customers. Some of the internal customers are consultants at the service team, which help the customer install the customer’s system. There are also consultants at the support team, which help the customers when they need support with the system. Some of the external customers are end-users at customer companies’ information technology departments, responsible for the maintenance of their system. The internal and external customers are a heterogeneous group in terms of both background and responsibilities.

In this thesis two prototypes for an information dashboard presenting some of this information were created. Both prototypes present the same information in two different design solutions. The prototypes were then evaluated with customers to decide which design solution they prefer, why they prefer it, and how they can be improved. During the evaluation half of the group of respondents got a chance to get familiar with the first prototype and the rest got a chance to get familiar with the second prototype before they evaluated both sequentially.

1.1. The problem

In general an ERP system is complex since it normally consists of several products or components. The tools for managing the M3 system need to have much information about the system, presented in different interfaces. This makes it difficult to find specific information at a given moment. It is also difficult to get an overview of the system because of all the detailed information. Some of the detailed information is used more frequently than the rest.
1.2. The task
As a solution to the problem Infor aims to create an information dashboard for M3, which will present relevant information from the tools in a structured and simplified way. The task from Infor is to evaluate what information the users find relevant to view in the dashboard, on what type of device they prefer to view it, and to suggest an approach for creating a dashboard together with a prototype, which presents a proposed solution.

There are several ways to evaluate design solutions. This study will focus on using prototypes. Prototypes are useful to validate requirements, design concepts, and get feedback on how to improve the suggested solution (Arnowitz et al., 2007). In this study prototyping in combination with user evaluation will be used to validate the proposed information, the presentation of it, and to gain knowledge of how to improve it. The study will therefore consist of a design part, where the prototype is designed, and an evaluation part, where the prototype is evaluated with users.

This is a software usability design case, where the relevant theory areas are software prototyping, design of software information dashboards, and software evaluation. Software prototyping is a well-known field (Blomkvist & Holmlid, 2011), and there are many theories about prototyping for graphical user interfaces. The same goes for software evaluation where there are many theories and different approaches. Dashboard design is unfortunately an academically unexplored area, and both the area of design and information presentation are very context dependent.

When creating and evaluating prototypes, research has shown that creating and evaluating more than one prototype in parallel result in design of higher quality, and a better exploration of the design space (Dow et al., 2010). This study will take advantage of this, and develop and evaluate two prototypes in parallel during the design and evaluation process.

In addition, a user’s experience of a product changes over time (Karapanos et al., 2009). Since the users of the dashboard will use it much in their work it is more important that they find the dashboard useful over time than right away. This is an unexplored area, and it is therefore interesting to study if the possibility to get familiar with the prototype affects the user’s experiences and preferences.

There are many ways to present information in a dashboard. One way is to make it list-based. The user will then be familiar with the interface since many of the existing tools are list-based, and it is an effective way to present much information without making the interface look cluttered.

Another option is to consider Few’s (2013, p. 77) statement “how we see is closely tied to how we think”. It might be more intuitive to view the system as several objects connected to each other, and where each object represents a product or component in the system. Recently the use of objects has become more common when presenting information, as in tablet applications and Microsoft’s Metro design. It is difficult to know which approach the users prefer. One prototype will therefore be list-based, and one will be object-based.

1.3. Thesis purpose & research questions
The purpose of the thesis is to use parallel prototypes and evaluation by users to explore how to present information of an ERP system in a dashboard.
The relevant research questions are:

1. Do the users prefer a list-based or an object-based dashboard?
2. How does the possibility to get familiar with the prototype before the evaluation affect the result?

1.4. Definitions
Definitions of important terms:

- **IT system** – Information Technology system
- **ERP system** – Enterprise Resource Planning system. An ERP system is a cross-functional IT system that consists of several integrated software modules, which support the basic internal business processes in a company
- **Information dashboard** – in this thesis an information dashboard is defined according to Few (2004, in Few 2013, p. 26):
  
  *A dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated, and arranged on a single screen so the information can be monitored at a glance.*

- **Dashboard** – in this thesis dashboard means information dashboard, if nothing else is specified
- **M3** – the ERP system the dashboard will present information about.
- **Products/applications/tools/components** – the different parts of M3
- **Space** – an expansion of the term environment. The term space is new for M3 13.1. A space is a categorization of the products in an M3 installation. There are three types of spaces: Production (PRD), Test (TST), and Development (DEV). All products that are used live in the company’s business belong to a space of type Production, those installed to be used when testing new versions or during education belong to a space of type Test, and those installed to be used for development and customization of M3 belong to a space of type Development
- **M3 Solution** – the customer’s installation of M3, including core products, compatible Infor products, and third party products used by M3. A M3 solution includes all the belonging spaces
- **LCM** core product used to install, manage, and configure the other M3 core products, and a few of the compatible Infor products
- **Grid** – another core product. It handles the distribution of the virtual servers the system runs on. Each Grid belongs to a space, and is responsible for some of the other products belonging to the same space
- **M3BE** – in this thesis M3BE means the product M3 Business Engine Environment, which is the space specific part of M3BE, if nothing else is specified. The product contains all business logic in M3
- **Host** – the virtual/physical servers the system is running on
- **Design solution** – a representation of the dashboard
- **A software prototype** – a model or simulation of a product, used during the development to test ideas or to gain feedback
- **UI** – user interface, the graphical interface by which the user interacts with a program
• **User** – the intended users of the dashboard, if nothing else is specified. This includes both the internal and external customers
• **Internal customer** – Infor employees, primarily from the Service, Support, Demo Environment, and Sales team
• **External customer** – employees at companies that use M3

1.5. **Disposition**
The next chapter is Background, which presents a more comprehensive background to the ERP system, the tools and the users. After that the Theory chapter presents the theoretical framework. It includes theory about information dashboards, dashboard design, prototypes, parallel prototypes, and evaluations. Then the Method and Execution chapters describe the chosen methods, and how the study was conducted. Execution includes a description of the prototypes. After that the Result chapter presents the result of the evaluation process. In the Discussion chapter the result is discussed. Lastly, the Conclusion chapter answers the research questions.

Last in the report the references and the appendices are presented. Appendix A presents the statements and the interview questions in the evaluation and the information sent to the respondents. Appendix B contains images of the prototypes.
2. Background

This chapter gives the reader a deeper understanding of the background to the study. It starts with explaining the ERP system, continues with the management tools, and ends with describing the intended users of the dashboard.

2.1. The enterprise resource planning system

M3, formerly Movex, is an ERP system that has existed since the 80s. It is designed for medium to large companies, and focuses on businesses dealing with chemicals, distribution, equipment, fashion, food & beverage, and industrial manufacturing. The latest version of M3 is M3 13.1, which was released in May 2013. M3 is a feature rich ERP system that consists of tens of core products, is dependent on a few third party products for servers and databases, and is compatible with other Infor products, such as the user interface Smart Office and the search function Enterprise Search.

The needs of the customer differ greatly, and therefore do their installations of M3 as well, both in which third party products they use and how they configure the system as well as which compatible Infor products they use. In an M3 solution the customer often views the compatible Infor products and third party products as part of M3. Many customers also run very old versions of M3, which are much less evolved than the latest version, but the dashboard will primarily be made for the latest version, M3 13.1.

An M3 solution includes several spaces. A space, also known as an environment in the system, is a categorization of the installed products. There are three types of spaces, Production (PRD), Test (TST), and Development (DEV). Each customer has at least one space of every type. In each space there can be different products installed of different versions. Usually the customer uses a space of type Test to test new fixes or changes to the solution, before they apply the change in a space of type Production, which is the live-version. In a space of type Development the customer can develop their own modifications and customizations of the system.

Large ERP systems tend to be complex by nature. Today’s M3 has old roots and the code has been rewritten. It has also evolved and expanded greatly during its life time. This makes the system somewhat complex and difficult to understand, since some solutions are a result of old solutions and backward compatibility. Unfortunately M3 is too large and too technically advanced to fully understand in the time frame of a master thesis.

2.2. The management tools

One of the core products in M3 is the Life Cycle Manager, LCM. The LCM is used to install, manage, and configure the other M3 core products, and a few of the compatible Infor products. One other important product is the Grid, which was introduced 2010. The Grid handles the distribution of workload across the virtual servers the system runs on. Each Grid belongs to a space, and is responsible for some of the other products belonging to the same space. Figure 2.1 presents a model of an M3 solution.
The LCM’s user interface is run on a PC, and does not have a web interface or tablet interface. It shows static information about the products that it handles. In the LCM’s user interface the user can also view the Grid’s user interface, which is also available on the web. The Grid’s user interface presents dynamic runtime information about the products that it handles. The Grid is made by and for very technically skilled people. In the Grid’s user interface the information is flooding, and it requires much practice to easily find what one is searching for, and much technical knowledge to understand what one find. Some of this information is used more than the rest. There are also other products such as the M3 Adaptation Kit and the Business Engine that have useful information, not to mention all non-existing information that would be useful to have.

There are different versions of the LCM. In the most recent, LCM 10.1, which was released in May 2013 and only few customers have seen yet, there are two tabs, as seen in figure 2.2 below. The first tab presents the hosts and the second presents the applications. Each host is either a virtual or a physical server. For each host a tree-view can be opened, presenting the products installed on that server together with the version of the product, and a green or red symbol that represents the status of the product, i.e. if the product is up and running or stopped. In the second tab there is a list of spaces. For each space a tree-view can be opened, with all the products belonging to that space not handled by a Grid. One of these products is the Grid. When opening the Grid in the tree-structure all the products that it handles are presented. Static information about the applications or the hosts opens up as a tab in the right part of the window. Runtime information is presented in the Grid’s user interface pages, also opened up as a tab in the right part of the window. Figure 2.2 presents an image of the LCM’s user interface.

For every product some general information is available, for example what version of the product is installed, if there are any fixes applied, where the product is installed, what server it runs on, if it is up and running or stopped. For some products the LCM also has information about what database and other products they are connected to. The Grid has runtime information about all the products it handles, available in the Grid’s user interface. Some products also have their own page with additional information in the Grid’s user interface.
2. Background

Figure 2.2: Image of the Life Cycle Manager’s user interface.
One of the problems today is that each installation includes many products in several spaces. Another problem is that both the LCM and especially the Grid contain much information, and the relevant information is located at many different places in the LCM and the Grid, making it difficult to get an overview of the system. A third problem is that not all users that would be helped in their work by an overview of the system and some basic information about the products have access to the LCM. A fourth problem is that it is difficult to find information in the Grid’s user interface since there is so much information on several different pages. It works well for technicians that work with it daily and need a lot of detailed information, but not for the common user to easily find some specific information. A fifth problem is that the LCM does not have a web interface, and is not suitable to view in a tablet, which means the users must either have the LCM Client installed on their computer or have access to it via Remote Desktop.

2.3. The users
The intended users of the M3 dashboard are both internal and external customers. Internal in the form of Infor employees at Service, Support, Demo Environment, and Sales team. External in the form of employees at the external customers’ Information Technology teams. These users have different backgrounds, use the system differently, and therefore search for different kind of information in it. There are also a group of users that don’t have access to the LCM but still could be helped by the overview of the system in the dashboard.

2.3.1. Services
Service consultants install the system from the beginning. They are called in when it is time to install a new product, upgrade an old one or in some other way configure the system. For this they need to see what products and what versions are installed, especially to be able to check the prerequisites for the new products/versions. During the implementation phase they also need to be able to monitor the system, and to manage problems. To manage and troubleshoot problems they need to be able to view the log files and, through the Grid’s Management page, execute, kill, or monitor individual jobs and subsystems in M3BE.

2.3.2. Support
Support analysts are called in when something is not working. They need to see what products and versions are installed, and which fixes that are applied. They also need to be able to drill down through the information. There are both technical and functional analysts at the support team. The technical analysts will need a deeper insight into the products and versions installed. Functional analysts focus on the business logic and are mostly interested in the level of corrections for a given data flow. Their need for detailed technical information is not as important as a general overview of the system, and the level of patches concerned by their respective domain.

2.3.3. Demo Environment
The demo environment team handles the internal demo environments. There are also other environment teams that handle the other internal environments such as those used for test and development.

2.3.4. Sales
The sales team is working with new customers. They present the user interfaces to new customers.
2.3.5. **External customers**

Some customers are large companies with very large solutions and a large IT-department. Others are small, with a smaller solution but also with much smaller IT-department with less expertise. They need a dashboard both to get an overview but also to make it easier to find important information when managing the system. When something does not work they need to find out if it is as simple as a product that has stopped or something worse, meaning it is time to call someone at the service or support team. External customers would prefer to be able to monitor the system, but this is a huge area, out of scope for this master thesis.
3. Theory

This chapter presents the theoretical framework of this study. It includes theory about dashboard design, prototypes, and evaluations.

3.1. Dashboards

A definition of an information dashboard is made by Stephen Few:

   A dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated, and arranged on a single screen so the information can be monitored at a glance.
   (Few, 2004, in Few, 2013, p. 26)

Information dashboards have become very popular, especially for business management, and for monitoring the performance of a company (Rasmussen et al., 2009). They are often used as part of a company’s business intelligence solution, presenting Key Performance Indicators (KPIs). KPIs measure important data in the company, which are supposed to indicate how well the company is performing, and are therefore different for different companies. KPIs can be both quantitative and qualitative.

Information dashboards are not new; they have existed since the 1980s under the name of Executive Information Systems (Few, 2013). The reason that they have not become popular until now is that they have suffered from a lack of sophisticated technology, and therefore not been able to fulfill their purpose. According to Few (2013), one problem still remains for dashboards to fulfill their purpose: dashboards often fail to communicate the information.

Dashboards primarily use graphics to present the information, with the support of text, since graphics communicate more efficiently (Few, 2013). However, it is difficult to design a dashboard with so much information that must fit on one single screen, and still be easily perceived. According to Few (2013) the following requirements are fundamental to make the dashboard useful. Firstly, fitting the dashboard into one single screen will help the user to get an overall understanding of the situation, and relieve the short-term memory. Secondly, the display media need to be small, and communicate the information in a clear way. Thirdly, the dashboard must be customizable.

3.2. Human perception

To understand the design guidelines described later in chapter 4, it can be useful to have some basic knowledge about human perception. Human perception is a huge area that may fill more than a thesis in itself. Some basic areas applicable for this thesis are the memory, the impact of colors, and some of the basic Gestalt principles.

In the citation below Few describes why perception is important.

   Our eyes do not sense everything that is visible in the world around us. Only a portion of what our eyes take in becomes an object of focus, and only through focus does what we see become more than a vague sense. Further, only a fraction of what we focus on becomes the object of attention, and only a portion of that is further processed as conscious thought. Finally, only a little bit of what we attend to gets
According to Few (2013) the human memory is split into three types: iconic, working, and long-term. In the iconic memory there is a preconscious processing of what we see called preattentive processing. Attributes that are recognized during the process are differences in color, spatial position, form, and motion (Ware, 2013). Some of these categories of attributes can be split into more attributes. Ware (2013) mentions 17 such attributes. Few (2013) selects the following 11 of these attributes as especially relevant in dashboard design. The category color can be split into hue and intensity. Form can be split into orientation, length, width, size, shape, added marks, and enclosure. Position is the 2-D location, and motion is flickering. Something to keep in mind is that many of these attributes are perceived relative to the context. For example the same color appears different on a different background. For each attribute except length and 2-D position, humans can only distinguish between a limited number of different expressions with ease. It is best to use no more than five. All the attributes can be used both to group, and to highlight information.

The working memory is conscious but limited in both time and size. Humans can only store three to nine pieces of information at a time. This is why a graph often is better than numbers. A graph presenting many numbers can be stored as one piece, while a number is stored as one piece in itself. This is also why all relevant information must be kept at the same view. (Few, 2013)

Differences in colors distract the viewer, and the brain automatically tries to find a meaning of that difference. If a dashboard is designed with the same color at different components, the brain tries to relate them. When the designer uses different colors to relate parts in a graph to additional information, the colors need to be different enough so they easily can be separated, for example by using different colors instead of different shades of the same color. However, colorblind people often easier distinguish between different hues of the same color than between different colors. Bright colors distracts more than soft colors. Too many sharp colors are stressful. (Few, 2013)

Gestalt theory concerns how we organize information to understand it, and how this is connected to how we perceive patterns. It is useful in dashboard design because of the two great challenges:

... 1) making the most important data stand out from the rest, and 2) arranging what is often a great deal of disparate information in a way that makes sense, gives it meaning, and supports its efficient perception. (Few, 2013, p. 91)

Four of the basic principles in Gestalt theory, according to Sternberg (2006) and Few (2013) are:

- The Principle of Proximity: “We perceive objects that are located near one another as belonging to the same group” (Few, 2013, p. 87).
- The Principle of Similarity: “We tend to group objects that are similar in color, size, shape, and orientation” (Few, 2013, p. 88).
- The Principle of Closure: “When faced with ambiguous visual stimuli – objects that could be perceived either as open, incomplete, and unusual forms or as closed, whole, and regular forms – we naturally perceive them as the latter” (Few, 2013, p. 89).
3. Theory

- The Principle of Continuity: “We perceive objects as belonging together, as part of a single whole, if they are aligned with one another or appear to continue one another” (Few, 2013, p. 90).

Few (2013) adds two more basic principles that are relevant in dashboard design:

- The Principle of Enclosure: Objects can be enclosed by either a border or by having a different background color.
- The Principle of Connection: Two objects can be perceived as a group if they are connected by a line. “The perception of grouping produced by connection is stronger than that produced by proximity or similarity (color, size, and shape); it is weaker only than that produced by enclosure” (Few, 2013, p. 91).

3.3. Prototypes

According to Arnowitz et al. (2007) prototypes have been used for a long time in product development. Some of the reasons they present are product innovation, idea refinement, using the prototype as a requirement specification, for evaluating the requirements, evaluation of design, and communication with stakeholders. Both Arnowitz et al. (2007) and Verplank (1992 in Muñoz, 1992 p. 579) state that the primary purpose of a prototype is to convert a design idea into a tangible artifact that others can give feedback on.

Prototyping in software development has also been used for a long time, and it is widely known that prototyping is an important tool in the development process (Arnowitz et al., 2007). Since this thesis concerns the use of prototypes in the design of an interactive user interface the rest of this section will focus on prototyping for human-computer interaction (HCI). Both Lim et al. (2008) and Houde & Hill (1997) claim that the use of prototyping in the field of HCI is both important and common practice. A definition of a prototype for an interactive system has been made by Beaudouin-Lafon & Mackay (2007):

> We define a prototype as a concrete representation of part or all of an interactive system. A prototype is a tangible artifact, not an abstract description that requires interpretation. (Beaudouin-Lafon & Mackay, 2007, p. 1018)

The purpose of a prototype is to answer questions during the development of the final product (Arnowitz et al., 2007; Houde & Hill, 1997). Houde & Hill (1997) continues that by focusing on the purpose of the prototype it is easier to decide what kind of prototype to build, and what attributes it should have. It is important to decide “what the prototype is intended to explore; and equally important, what it does not explore” (Houde & Hill, 1997, p. 369). Lim et al. (2008) also hold that it is important to not lose the overall picture:

> ... the purpose of designing a prototype is to find the manifestation that, in its simplest form, will filter the qualities in which the designer is interested without distorting the understanding of the whole. (Lim et al., 2008, p. 7:10)

Prototypes can be used for many reasons. They can be used to explore an idea or design concept, as well as a living requirement specification since it can guide the developers during the implementation (Arnowitz et al., 2007). They can also be used to increase the usability and the
look-and-feel of the product by letting the user test it and provide feedback. When developing software some requirements are uncertain. By creating a prototype, the requirement can be tested and verified by a user. A prototype can also help the user to find out what they like and not:

*It is often said that users can’t tell you what they want, but when they see something and get to use it, they soon know what they don’t want.*

(Sharp et al., 2007, p. 530)

### 3.3.1. Attributes/characteristics of a prototype

Prototypes are often described with different attributes or characteristics. Arnowitz et al. (2007) has divided the attributes into eight categories listed below together with their possible values:

- **Audience** – internal/external
- **Stage** – early/midterm/late
- **Speed** – rapid/diligent
- **Longevity** – short/medium/long
- **Expression** – conceptual/experiential
- **Style** – narrative/interactive
- **Medium** – physical/digital
- **Fidelity** – low/medium/high

The first attribute is audience, which can be either internal or external. It is preferable to start testing the ideas on the internal stakeholders with rapid and iterative prototyping. When the ideas start to get more complete it is time to create an interactive prototype that look more like the final product to present to the external stakeholders. The second attribute is stage, which is at what time in the design process the prototype is created, either early, midterm or late. In the early stage the prototype can be used to explore the conceptual design, and in the midterm the prototype can be used to validate it. In the late stage the prototype is used to refine the design, and “… should only be conducted when the design concepts, product scope, and product vision are firmly established …” (Arnowitz et al., 2007, p. 116). The third attribute is speed, which considers how long it takes to create the prototype, and how much details it includes. Diligent means more detail but longer time, and rapid means quick but with few details. (Arnowitz et al., 2007)

The fourth attribute is longevity, which is whether it is created just to answer some questions and then thrown away, a throw-away prototype, or if its influence will last, perhaps as a specification or as a basis for future prototypes. The fifth attribute is expression, which refers to what level of look-and-feel the prototype should express. Conceptual prototypes present the main concept and are used for idea generation and evaluation in the early stages of the design process. Experiential prototypes try to communicate the user experience of the product and are used for communication and validation with stakeholders. The sixth attribute is style, which refers to whether the prototype is interactive or not. In a narrative prototype the prototype presents a scenario while in an interactive prototype the user can affect the scenario by making choices. The seventh attribute is medium, which refers to whether the prototype is digital or not. (Arnowitz et al., 2007)
The last attribute, fidelity, was one of the first being discussed in the field of prototyping. The meaning of fidelity is ambiguous. In the earlier articles it seems as if the level of fidelity refers to whether it can be done quick and cheap, and whether it is implemented or not (Rettig, 1994). Implemented and non-implemented prototypes are also referred to as online and offline prototypes (Beaudouin-Lafon & Mackay, 2007). Fidelity also refers to the level of details and “the degree to which the prototype accurately represents the appearance and interaction of the product ...” (Rudd et al., 1996, p. 78). This is later split up where the first is referred to as resolution and the later as fidelity (Houde & Hill, 1997). The level of detail has also been referred to as precision by Beaudouin-Lafon & Mackay (2007).

### 3.3.2. Low- and high-fidelity

Prototypes can have a low or high level of fidelity. Low-fidelity prototypes, for example paper prototypes, are quick and cheap to create. They are suitable in the early stages of the design process when there is still much uncertainty, to be able to iterate and create several (Rettig, 1994; McCurdy et al., 2006; Arnowitz et al., 2007). High-fidelity prototypes take more time and effort to create and are more difficult to modify. They are therefore more suitable in the later stages of the design process (McCurdy et al., 2006), when the overall concept is decided and there are only a few design approaches left to investigate (Rudd et al., 1996; Arnowitz et al., 2007). If the product looks as if it is almost finished early in the design process there is a risk that the users focus on the details instead of the important issues (Rettig, 1994). There is also a risk that they believe that the product is almost finished and therefore expects it to be released soon (Rettig, 1994; McCurdy et al., 2006). According to Houde & Hill (1997) the purpose of the prototype should decide how much time that is worth spending on creating the prototype and the suitable level of detail.

Since high-fidelity prototypes take more time and effort to create one can choose to only prototype a part of the product. Rudd et al. (1996) and Beaudouin-Lafon & Mackay (2007), among others, mention vertical and horizontal prototypes. A vertical prototype only presents a subset of the functionality. “The purpose of a vertical prototype is to ensure that the designer can implement the full, working system, from the user interface layer down to the underlying system layer” (Beaudouin-Lafon & Mackay, 2007, p. 1024). A horizontal prototype on the other hand only presents one layer of the system, for example the user interface without any underlying functionality (Beaudouin-Lafon & Mackay, 2007). “Horizontal prototypes of the user interface are useful to get an overall picture of the system from the user's perspective and address issues such as consistency ..., coverage ..., and redundancy ...” (Beaudouin-Lafon & Mackay, 2007, p. 1024).

### 3.3.3. Mixed-fidelity

In the beginning the experts thought that a prototype only had one level of fidelity. But for the prototype to fulfill its purpose and answer specific questions one level of fidelity is not enough (McCurdy et al., 2006; Arnowitz et al., 2007). It is important to better characterize the space of fidelity.

Arnowitz et al. (2007) state that:

*By deliberately making some elements high fidelity, the audience is better able to focus on the higher fidelity items, giving them an*
They proceed to present six independent categories of a prototype’s content: information design, interaction design and navigation model, visual design, editorial content, brand expression, and system performance and behavior. The first category, information design, refers to how the users navigate between the screens. The second category, interaction design and navigation model, includes the layout, graphic elements, font selection, and color scheme (Dondis, 1973 in Arnowitz et al., 2007). The fourth category, editorial content, is the text. It includes both what text is used and how, for example labels, headers, and the expression in them. The fifth category, brand expression, refers to the impression of the brand the prototype gives the user. The last category, system performance and behavior, refers to the technical impression the user gets from the prototype. (Arnowitz et al., 2007)

Instead of Arnowitz et al.’s categories, McCurdy et al. (2006) split the fidelity into five orthogonal dimensions: level of visual refinement, breadth of functionality, depth of functionality, richness of interactivity, and richness of data model. In the first dimension, level of visual refinement, it is good to have high fidelity when one needs feedback on the visual design but low fidelity when one needs feedback on the other dimensions. In the second dimension, breadth of functionality, high fidelity is similar to horizontal prototyping. In the same way, high fidelity in the third dimension, depth of functionality, is similar to vertical prototyping. The fourth dimension, richness of interactivity refers to the same attribute as Arnowitz et al. (2007) called style, i.e. whether the user can interact with the prototype or not. The fifth dimension, richness of data model, refers to how much of the actual data the prototype will support. McCurdy et al. (2006) state that:

*By using these dimensions --- it is possible to create mixed-fidelity prototypes that more precisely apply prototyping resources in support of specific end goals.* (McCurdy et al., 2006, p. 1235)

### 3.3.4. Parallel prototyping

Recent studies performed by Dow et al. (2009) show that iterative prototyping produces better design. This is supported by Sharp et al. (2007), who state: “the more iterations, the better the final product will be” (Sharp et al., 2007, p. 530). Dow et al. (2010) also show that creating and evaluating multiple prototypes in parallel results in design of higher quality, a better exploration of the design space, and increased self-efficacy. Parallel prototypes not only increase the result, they also increase the feedback (Dow et al., 2011). Perhaps people are more comfortable with giving critique on multiple prototypes.

### 3.4. Evaluation

Evaluations can be done for different reasons, for example to improve the design or choose between two concepts (Goodwin, 2009). An evaluation that is performed to gain feedback on the usability of the design is called a usability study (Sharp et al., 2007). According to Goodwin (2009) there are different types of evaluations, both depending on when in the design process it is performed, and on the approach, i.e. how and with whom the evaluation is performed. Depending on when in the design process one performs the evaluation it can either be formative or summative. A formative evaluation is done during a project to correct the direction of the
product while summative is done in the end of a project to refine the design of the product. Both can be used when comparing several concepts in a comparative evaluation.

3.4.1. What to evaluate
The purpose of the evaluation should decide what to evaluate, but to be sure that the evaluation covers all relevant areas of the design solution it may be useful to use the model by Arvola & Artman (2007). Their model includes the following five elements: design concept, function and content, structure, interaction, and presentation. Design concept is the design idea of the product, i.e. the purpose of it and how the product is supposed to be used. The function and content corresponds to both the functions in the product and the information content. The structure is how the functions and content are arranged in the design. The interaction is how the user interacts with the design. The presentation includes both the style and the layout of the product.

3.4.2. Usability
Usability measures how well a product fulfills its purpose. It does not exist in an absolute sense, and it can only be defined and measured in reference to the context (Brooke, 1996). According to ISO 9241-11:1998 usability is defined as the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a certain context of use. The ISO standard defines the measurements of usability as:

- effectiveness – the accuracy and completeness with which users achieve specified goals
- efficiency – the resources expended in relation to the accuracy and completeness with which users achieve goals
- satisfaction – the freedom from discomfort, and positive attitudes towards the use of the product

It is therefore important to define the purpose of the product, who the intended users are and what tasks the system is supposed to support (Brooke, 1996).

Recent studies performed by Karapanos et al. (2009) show that the user’s experience of a product changes over time. It is other qualities that make a user want to use the product over time than those that make the user like the product from the beginning.

3.4.3. System usability scale
In addition to the general approaches to evaluate the usability of a system, it can also be useful to have a “quick-and-dirty” way of measuring it (Brooke, 1996). It is also useful to be able to compare the results between different products. One such tool is the System usability scale, SUS. It consists of a Likert scale with ten standardized statements about the system’s usability, which the user will grade from 1-5 (or 1-7) where 1 means strongly disagree and 5 means strongly agree. Half of the statements are formulated so that a 5 corresponds to high usability and the other half so that a 5 corresponds to low usability. The ten statements were selected from a pool of 50 statements, where the ten with the most extreme responses and with high intercorrelation were selected. The ten statements are to be found in Appendix A.

The user should be asked to grade the statements before any following discussion takes place. The user is asked to grade them quickly with little reflection and if the user can’t decide he or she should choose three as grade on a five point scale. (Brooke, 1996)
The grading of SUS corresponds to a score, with a range of 0-100. It is calculated by first calculating the value of each statement, where the value depends on the grade of the statement. Each statement corresponds to a value from 0-4, which depends on the grade (1-5). The value of statement 1, 3, 5, 7, and 9, is the grade (1-5) minus 1 and the value of 2, 4, 6, 8, and 10 is 5 minus the grade (1-5). For example if statement 1 gains a grade of 2 its value is 2-1 = 1 and if statement 2 gains a grade of 2 its value is 5-2 = 3. The values are then summarized and the sum is multiplied with 2.5. The score is a measurement of the products usability (Brooke, 1996). However, it is not meaningful to view the result of each statement separately.

SUS gives a reliable measurement of the usability and correlates well with other measurements of usability (Brooke, 1996; Bangor et al., 2008). As a value to compare the score with, Bangor et al. (2008) states:

*This means that products which are at least passable have SUS scores above 70, with better products scoring in the high 70s to upper 80s. Truly superior products score better than 90.* (Bangor et al., 2008, p. 592)

The statement can be interpreted as though the usability of a product with score above 70 is acceptable, while 78-89 is good and 90-100 is excellent.
4. Method

The following chapter presents the method of the study. The study was divided into a design part and an evaluation part. The design part included several iterations of low-fidelity prototyping and resulted in two final prototypes. These two prototypes were then evaluated with users during the evaluation part. The evaluation included a quantitative and a qualitative part.

The task from Infor, which was presented in the introduction, was broken down to six subtasks. The first subtask was to interview a selection of users to gain an initial understanding of how they would use the dashboard, what type of information they find relevant, and what types of users would be helped in their work by a dashboard. The second subtask was to select the most suitable information to present in a first version of the dashboard. The third subtask was to explore the design space, using two approaches: one list-based design and one object-based. The fourth subtask was to create two prototypes of the final design solutions and implement them in html5. The last subtask was to evaluate the prototypes with users to validate if the prototypes fit their needs, which one they prefer, and how to improve them.

The initial interviews and the selection of which information to present was made outside of the thesis, and the prototypes were created in a prototyping tool instead of implemented in html5. Therefore only the design and the evaluation of the prototypes are included in the report.

In the design process there were several iterations with question-sessions, sketching, and feedback from internal stakeholders. It resulted in the creation of two interactive digital prototypes created in a prototyping tool, presenting the same information in two different design solutions. In the evaluation process there was a qualitative part, in the form of interviews, and a quantitative part, in the form of the System usability scale.

4.1. Type of study

There has been little research in the area of dashboard design. The study is therefore explorative (Patel & Davidsson, 2003). The evaluation is a comparison between two prototypes, which makes the study comparative. The study is primarily inductive since dashboard design is an unexplored area and the study draws conclusions from the empirical data (Bryman, 2002; Patel & Davidsson, 2003). The study is based on qualitative and quantitative data from the evaluation. This study is therefore explorative, comparative, with a primarily inductive approach, and based on both qualitative and quantitative data.

4.2. The design process

According to Sharp et al. (2007) and Dow et al. (2009) a design process should be iterative. To explore the design space it is useful to have prototyping iterations with sketching and feedback from internal stakeholders (Arnowitz et al., 2007).

During the sketching process, low-fidelity prototypes should be used to be able to do several iterations during the exploration of the two concepts. Low-fidelity prototypes are more suitable in the early stages of the design process, since they are quick and easy to create (Rettig, 1994).

Two prototypes were created both to be able to explore the design space better (Dow et al., 2010), and to be able to compare the two concepts and evaluate which one users prefer. One of the prototypes is based on lists and explored how the dashboard can present information in an
efficient way that resembles the other tools, which the users are familiar with. The other prototype is based on objects and explored how the dashboard can present information of the system in a way that is closer to how the system is operating, and was inspired by tablet applications.

To get useful feedback for the prototypes during the evaluation it was important to decide what type of prototypes to create, and which features to include. To decide what kind of prototypes to create it was useful to consider the purpose of them (Houde & Hill, 1997), and decide which questions they should answer (Arnowitz, et al., 2007). It was useful to describe them according to Arnowitz et al.’s (2007) attributes, to gain a deeper understanding. Especially the last attribute fidelity was important to take a closer look at. For the prototype to more precisely answer the relevant questions it should have mixed fidelity. To decide which parts of the prototype that should have high fidelity, and which parts that should have low fidelity it was useful to divide the fidelity according to Arnowitz et al.’s (2007) six areas or McCurdy et al.’s (2006) five dimensions.

It was useful to have some design guidelines to follow during the design of the prototypes. Design guidelines for interaction design are based on an understanding of human perception and gestalt theory. The following design guidelines are not specific for dashboard design, but general for interaction design. The chosen guidelines are however more applicable to dashboards, according to Few (2013).

In dashboard design less is actually more, “eloquence of communication through simplicity” as stated by Few (2013, p. 93). It is easier to communicate clearly when avoiding distracting decorations. Tufte (2001) calls this the data-ink ratio, the amount of data-ink divided by the amount of total ink used to print the graphics, where data-ink is the ink used to present data.

*Maximize the data-ink ratio, within reason. Every bit of ink on a graphic requires a reason. And nearly always that reason should be that the ink presents new information.* (Tufte, 2001, p. 96)

Even if Tufte primarily talks about data-ink ratio when presenting quantitative data, Few (2013) applies this on other information as well. To maximize the data-ink ratio Few uses the following strategy: reduce the non-data pixels, and then enhance the data pixels. This is done by first removing all unnecessary non-data pixels, and de-emphasizing the remaining ones. Then all unnecessary data pixels are removed, and the most important ones are highlighted.

1. Reduce the non-data pixels
   A. Eliminate all unnecessary non-data pixels.
   B. De-emphasize and regularize the non-data pixels that remain.

2. Enhance the data pixels
   A. Eliminate all unnecessary data pixels.
   B. Highlight the most important data pixels that remain.

**Figure 4.1:** The steps of useful dashboard design, by Few (2013, p. 98).

According to Few (2013) there are some useful guidelines when designing dashboards. The layout and the information structure are important. It should be clear which data are important, in what sequence the data should be viewed, which data are related, and which data are
relevant to compare. Data that always are important should be placed in the upper left corner or in the middle. Non-data information, such as navigation and selection of what data to show, are preferably located in the bottom-right corner. Data where the importance changes dynamically can be highlighted in different ways, which are described later. Instructional or descriptive text can be placed as pop-ups, so they are only present when necessary.

When adding data to a dashboard it is important to supply sufficient context to make the data understandable, but too many details can disturb the overview. The data should be presented in the most suitable way, for example a graph. It is preferable to use the same kind of graph if it is appropriate. Colors should only be added if they add meaning to the data, for example to emphasize or mark that the data are related to other data. Using different hue of the same color for indicating different importance is better than different colors for those who are colorblind. Lastly, the dashboard does not need to be unpleasant to look at just because there are no unnecessary decorations. (Few, 2013)

4.3. The evaluation process
The evaluation was formative and performed in the end of the design process as a comparison between the two design solutions. The focus was the usability of the information and the information presentation.

4.3.1. Evaluation approaches
There are several evaluation approaches. Two basic approaches are user studies, evaluations done with users (Sharp et al., 2007), and inspection methods, evaluations done by experts (Nielsen, 1994). The two approaches complement each other; some of the usability problems found by users are not found by the experts and vice versa (Nielsen, 1994).

User studies can be done with users in a test environment, which is called usability testing, or as a field study in the users’ natural environment (Sharp et al., 2007). In usability testing the test environment implies that the user will not be interrupted during the test and the user’s performance is measured and often quantified. Field studies are used to understand the user’s use of the product. They often include data gathering techniques like interviews and observations.

Two inspection methods are heuristic evaluation and walk-throughs (Nielsen, 1994). In the first usability experts evaluates the interface according to guidelines and standards. In walk-throughs the expert is testing different user scenarios and evaluates if the user would be able to perform it.

4.3.2. The chosen approach
The purpose of the evaluation in this study was to gain knowledge about if the users find the dashboard useful, which information is relevant to them, which design concept they prefer, what types of users they believe will have use of the dashboard, and how to improve the design. The chosen approach was therefore a usability study conducted with users. The reasons to do it with users instead of a usability expert were that there has been no thorough study of the users and the users are a heterogeneous group. In addition the ERP system has much available information. Furthermore, the evaluation was a comparison between two design concepts, not on the details of the design. It could however be useful to do an evaluation with a usability
expert in addition to the users, to gain more feedback on the details, later on in the design process when it is validated and the prototype includes the look-and-feel more explicitly.

4.3.3. Evaluation method
To get feedback on each prototype, what the users liked and disliked, one prototype at a time was presented and evaluated. If they had seen both directly there was a risk that they had decided which one they like before they gave feedback on them, which would have affected their answers.

For the result to be reliable half of the respondents evaluated the list prototype first and the other half the object prototype first. In order to do this the number of respondents needed to be at least eight, preferably ten. If the respondents had been more than ten the evaluations would have taken too much time to perform and the data too long to process.

Since the respondents were few, the users heterogeneous, and the ERP system has much information available, it was not only important to get knowledge about which prototype the users prefer but also why they prefer it. In addition it was important to get knowledge about if some information is missing, and why that information is relevant to add. A questionnaire, which is a tool to collect primarily quantitative data according to Bryman (2002), would not have answered these questions. Quantitative data often supports deductive studies while this study is primarily inductive. Interviews on the other hand are suitable to get answers to the questions and are a tool to collect qualitative data, which often supports inductive studies. Observations were not an option because of the geographic distances and limitations in time.

The interview questions were chosen given the purpose of the evaluation. They were semi-structured with both open questions, to get all possible feedback, and more specific questions, to be sure that the respondent reflected on the relevant details in the prototype. The interview started with open questions and ended with more specific questions, which is considered to be motivational for the respondent (Patel & Davidsson, 2003). To make sure that all relevant parts of the design solution were evaluated, the model by Arvola & Artman (2007) was used when the questions were created.

4.3.4. Triangulation
A strategy that leads to a more reliable result is triangulation (Sharp et al., 2007). Triangulation means that the study either relies on more than one technique to gather the data or more than one approach for analyzing it. The evaluation process included both a quantitative part, consisting of the System usability scale, and a qualitative part, consisting of a semi-structured interview.

The reason to use SUS was that it is quick and easy to use, and there is a value to compare the score with. SUS answered what the user thinks about the prototype’s usability. Since it was possible to compare the value between the prototypes it also answered the question of which prototype the user finds more usable. The interview gave feedback on what the user likes and dislikes in the prototypes, if the information is relevant, how they can be improved, which one they prefer and why.

4.3.5. Selection of respondents
In both quantitative and qualitative studies a selection of respondents is made. There are two types of selection: probability selection and non-probability selection. In probability selection the
respondents are chosen at random, which is assumed to make the selection representative for the whole population and thereby make the result generalizable. Probability selection is mostly used for quantitative research.

In non-probability selection the respondents are chosen in another way. There are three main types of non-probability selection: selection by convenience, snowball selection (also called chain selection) and quota selection. The first one means that the scientist selects people that are available at that time. This is an accepted and legitimate way to get a preliminary result or to make a pilot study to test how the survey is perceived before testing it on a probability selection. The second type, snowball selection, is a kind of a convenience selection. The selection of respondents is extended with people the respondents know, which are also available. This type is mostly used for qualitative studies. The third type, quota selection, is mostly used in commercial studies. The objective is to get a selection that reflects the demographic distribution in the population. This is done by deciding which categories are interesting and a number of respondents for each. Selection of respondents for qualitative studies is often a combination of selection by convenience and snowball selection. (Bryman, 2002)

In addition to the probability selection and the listed main types on non-probability selection there is an alternative strategy called theoretical sampling. According to Glaser & Strauss (1967, in Bryman, 2002) theoretical sampling is a better choice in qualitative research. Theoretical sampling aims at finding categories and their characteristics. It is an iterative approach where several consecutive interviews or observations are used both when formulating a category and for validating the categories’ importance. (Bryman, 2002)

The respondents for the evaluation in this study were selected by a mix of theoretical sampling and convenience with some elements of snowball selection (Bryman, 2002), because of the time and resource constraints of the study. The respondents were selected according to the following three criteria. The first one was that someone in the M3 Technology department knew the customer. This was set to make it easy to get in touch with the customers and to make it more likely that they would take time from their work to participate. Then the type of selection had elements of a snowball selection (Bryman, 2002), if the interviewed users knew someone they believed would have relevant feedback. The second criterion was that the customer needed to be interested in participating in the evaluation. This was set to get only relevant customers, which is important when the number of participants is small. The third criterion was that since the customers are a heterogeneous group it was useful to try to get representatives from as many groups as possible. Therefore the evaluation involved both internal and external customers.

The internal customers were employees from the departments of service, support, sales and demo environment. From services it was only technical consultants, but from support it was both technical and functional analysts. The respondents were both domestic and from abroad since Infor is a global company and people in different countries have different cultures. As a result of the criteria stated above only Europeans were selected. Because of the third criterion the selection was a kind of theoretical sampling (Bryman, 2002), in which the respondents represent different categories of users.
5. Execution

In this chapter the execution of the study is presented in detail. The chapter describes how the design process was performed, the resulting prototypes, and last how the evaluation was conducted.

To know what type of prototype to create, how to design the prototypes, and how to conduct the evaluation, applicable theory about software prototyping, design of software information dashboards, and software evaluation was selected. In the meantime the initial interviews were conducted in parallel. From the initial interviews (not the interviews conducted during the evaluation) the most suitable information was selected to be presented in the prototypes. After the design space was explored in several iterations of sketching, with feedback from and question sessions with internal stakeholders, the final prototypes were created in a prototyping tool called Indigo Studio. When creating the prototypes much consideration was done when deciding what type of prototype to create.

Then the preparation of the evaluations started, which included contacting suitable respondents, and designing the interview questions. The evaluations consisted of two sessions per respondent. The first session was only a demonstration of the first prototype. The second session included evaluation of the first, demonstration of the second, and evaluation of the second. The evaluation consisted of a quantitative part, in which the respondent was grading the ten statements in SUS, and a qualitative part, in which the respondent answered some semi-structured interview questions.

During the initial interviews it emerged that many of the respondents thought the idea of a dashboard was interesting, but that it was a bit difficult for them to decide precisely what information they were in need of. They thought it would be easier to give feedback on a prototype. By creating the prototypes, it was easier to get feedback on the information and to validate the design concepts.

5.1. The properties of the dashboard

From the task description and the initial interviews the following requirements emerged. The dashboard should be designed for the web. It should be possible to implement in html5 for the possibility that future versions will be available in a tablet or smart phone. The dashboard should allow the user to get a quick overview of the installed system, i.e. what products are installed on which host and connected to what space. For each product it should present additional information such as what version is installed, what fixes are applied, if there are updates available, which products it is connected to, if the product is up and running, offline or stopped, and links to the management page, the property page and the log file in the Grid’s web user interface. It should allow the user to customize it and to start/stop the products.

For the dashboard to fulfill the requirements and be useful for the users it needs to have certain properties. It needs to be updated frequently to check if the products are up and running or stopped and when new products or versions are available or installed. It needs to be interactive, for the user to be able to switch between different screens, to start/stop applications, to be able to get linked to the Grid’s web user interface, and to customize the view. Because of these links the dashboard will not be a pure dashboard; it will also be somewhat of a portal (Few, 2013), with short-cuts to useful pages in the Grid. Some of the users should have administrator
privileges with the right to start/stop the applications and some are ordinary users, only able to view the information.

The users will use the dashboard every day or at least a couple of times a week, so it is more important that they find the dashboard useful when they have had time to get familiar with it rather than like it the first time they use it. It is good to have tool tips for new users, to make it easier to get started.

5.2. The design process
During the exploration of the design space, the sketching resulted in digital paper prototypes, and the internal stakeholders consisted of my mentor, who develops the LCM, the component owner of the LCM, the manager and the director of the development team responsible for the LCM, three people responsible for the Grid, including the architect of the Grid and the director of the department responsible for the Grid, a usability analyst and interaction designer, and the supervisor of the study.

Some of the issues during the sketching and the creation of the final prototypes were the same for both prototypes. Two of these issues during sketching were how much information the dashboard should present and in which views the dashboard should present it. For the first issue the requirements were not precise enough to decide how much information to present. A dashboard should only present relevant information but since different information is relevant to different users it is better to present more information and add the possibility to hide non-relevant information. The downside of presenting much information in a dashboard is that each user must decide which information they are interested in and hide the other. There is a risk that they keep the default setting, which make the dashboard seem cluttered.

For the second issue, the applications can be viewed from a space perspective, with all applications for each space. However, not all applications belong to a space. In the LCM’s user interface there is, in addition to the spaces, the entry Other in which the License Server and the databases are listed. The applications can also be viewed from a host perspective with all applications for each host. This might be more interesting for more technically-oriented users, who also probably want to see more details while other users only want to see an overview of the system. Users often also want to compare the installation in two spaces. Therefore, the prototypes do include one view with the applications per space, one with applications and middleware that do not belong to a space, one view in which the user can compare two spaces, and one view with the hosts. In this way the prototypes follow the logic in the LCM. The users were able to give feedback about whether all views are relevant or not during the evaluation. Since the dashboard focuses on presenting an overview of the system, more effort was spent on creating the space view than the host view. By focusing on spaces in the dashboard the user is prompted to think in terms of spaces and it is consistent with the new LCM 10.1.

5.2.1. The prototypes
The resulting prototypes were created in a prototyping tool called Indigo Studio. I chose Indigo Studio since it is free and has the necessary functions. It includes support for storyboarding, sketching, and interactive wireframes.

To decide which features to implement in the final prototypes I thought about the purpose of them. The purpose of creating the prototypes and evaluating them was to validate that the users
find the dashboard useful, i.e. the information in it, to understand who the primary users are, and to validate the concepts; if the users found any of them useful, which one they prefer, and why. The purpose was also to get some feedback on some of the details such as the presented information, the terminology, the symbols, the customization possibilities, the navigation, and if the prototype would work with various M3 solutions. Therefore the prototypes were created somewhere in between low-fidelity and high-fidelity (McCurdy et al., 2006; Arnowitz et al., 2007).

The prototypes were not created with high-fidelity since the concept had not been validated yet (Arnowitz et al., 2007), and there was a risk that the user would have been given feedback on the look-and-feel instead of the concept (Rettig, 1994). According to Rettig (1994) there is also the risk that they would believe that the product was almost finished and therefore expect it to be released soon. This is essential for a development department; the user must not get the assumption that the prototype is due to be released. It is not even certain that the study eventually will end up as a finished product. Since the prototypes were made in a prototyping tool, not fully implemented and with fake information it was deemed to be little risk that the respondents would believe it was a nearly finished product.

For the user to be able to test the prototypes, what information they present and how they behave, the prototypes needed to show examples of this. Since it would have taken much time and effort to implement a fully functioning prototype, even in a prototyping tool, and the purpose of the prototype should guide how much is suitable (Houde & Hill, 1997), I chose to only present one or a few examples of each information and function. The downside was that it made the prototypes more difficult to navigate for the user and it required some imagination from the user to evaluate the prototype.

The purpose of the prototype should also guide how much detail to add in it. Since Infor has guidelines for their user interfaces it was not relevant to evaluate the colors, fonts, size of text etcetera. I did consider Infor’s guidelines and implemented those that were in line with the guidelines for dashboard design and that were mainstream, which made the prototypes look complete enough for the respondents to be able to validate the concept and the added details. From Infor’s guidelines I used a couple of shades of gray to emphasize different information and to make it less tiresome and easier to process than a screen, in which all text and lines are in black on a white background. I also chose to use a couple of different sizes of the font for the same reason. Lastly I also used a few of the symbols that were applicable in the prototype, like the one for settings and to close a window. Most of the symbols were taken from the LCM, so the user would be familiar with it. If the prototype would have been created later in the design process it would have been more relevant to implement all of Infor’s guidelines to make the prototype look more like the finished product, when it is primarily the details and look-and-feel the designer needs feedback on.

Another issue was how much example data the prototype should display. Adding much example data could have disturbed the user if it found any inconsistency in it or if the user is used to solutions that look different. On the other hand it is important that the actual data fits in the prototype’s user interface, and missing data might also disturb the user. I have therefore added the names of the applications presented earlier, and all applications that can be handled by the Grid are handled by the Grid. I have also added examples of the space information for a space of type Production and information about the Grid and M3 Business Engine Environment (M3BE) since
both the Grid and the M3BE are important products and contain much information. I have also added a database of each type and five hosts. I believe it is a reasonable assumption that most customers do not use more than 5-10 hosts.

5.2.2. The prototypes’ attributes
The prototypes’ attributes and characteristics were described according to Arnowitz et al.’s (2007) eight attributes:

1. Audience: The audience is external.

2. Stage: The stage is midterm. Perhaps it would have been better to evaluate the concept and basic assumptions on the customers before creating the interactive prototypes, but because of the limitations this was not an option.

3. Speed: The speed is in between diligent and rapid. The prototypes include some interaction and details but not any styling.

4. Longevity: The longevity is medium. The prototypes, at least one of them, will not be thrown away but kept for further development after the evaluation and as support during the implementation, but since they are not implemented they will not be kept after the implementation.

5. Expression: The expression is somewhat in between conceptual and experiential. It expresses more than the concept but it is not fully functional, styled or implemented and so the user will not be able to test the user experience.

6. Style: The style is narrative. The prototype only presents examples of what the user will be able to do with the dashboard, and how it will look and behave.

7. Medium: The medium is digital.

8. Fidelity: The fidelity is mixed.

The fidelity can be more closely specified according to Arnowitz et al.’s (2007) six areas:

- Information design: High – the design and structure of the information is important for a dashboard (Few, 2013). It is important to get feedback from the customers, if it is understandable and accessible.

- Interaction design and navigation model: Medium – the prototype presents examples of what the user can do with the dashboard and how to navigate, as in a narrative way. It would have been more important to implement all the interaction and navigation if the prototypes were made late in the process, to present the look-and-feel for the user. Now it takes too much time and effort compared to the possible feedback.

- Visual design: Medium – the visual design is important for a dashboard, to let the user easily access the information. The parts that are important for the dashboard has been implemented, such as using different shades of gray instead of black for the text and the boxes/lines, as few lines as possible, only colors in areas where it is relevant such as red if something is wrong, that the symbols are logical, etcetera. The prototypes are not styled (fonts, color-scheme, etcetera) since it is less important early in the process and it is quite restricted by Infor’s guidelines. The styling should not affect the focus of the
tester during the evaluation. It is either way not that important for a dashboard of an ERP system to be styled, it is more important that it shows correct information in an accessible way.

- Editorial content: Medium – it is important that the dashboard uses terms and expressions that the user recognizes and understands. Especially the object prototype contains many abbreviations, which need to make sense. However, it is a dashboard that will be used several times a week or even daily so the user will get used to the terms, but it will be less pleasant to use if the user gets annoyed. It would also be useful to get feedback on help texts and notifications, but this is not in focus in the prototypes. Some information is only presented a few examples of, for instance help texts and additional information to let the user understand what the final product would look like, how it would act and how it could be used. Hopefully, this makes the respondent give more feedback on what information the dashboard presents. It would have been more important to present all information if the prototypes were made later in the process.
- Brand expression: Low – some of the new brand expression is a minimalistic design. This is useful for dashboards, which I have implemented to some extent.
- System performance and behavior: Low – since it is only a non-implemented prototype of the design.

The fidelity specified according to McCurdy et al.’s (2006) five dimensions:

- Level of visual refinement: Medium
- Breadth of functionality: Medium – The prototypes present examples of all functionality so that the customer will understand the full usefulness of the dashboard.
- Depth of functionality: Low – the prototypes only present the user interface.
- Richness of interactivity: Medium – there are examples of all functionality but it is not fully implemented so the user can not try all possibilities.
- Richness of data model: Medium – The prototypes present enough data for the user to get a feeling for how it will look, how it will act and what it can be used for, but not more. More data only takes time from the design and might even disturb the tester during the evaluation if the data is inconsistent or if the tester is not used to the new M3 13.1.

5.2.3. Summary
The prototypes are digital with focus on the concept, a limited amount of details, and a minimum of styling. They are wireframes and the dashboard design guidelines are implemented together with some of Infor’s user interface guidelines.

5.3. The resulting prototypes
The result of the design process was two interactive prototypes, created in Indigo Studio, presenting the same information with two different design solutions. First there is a general description of the content and the functions, which is the same for both prototypes. Then the model by Arvola & Artman (2007) is used to present the design solutions in detail, both the general features that are the same in both prototypes and those that are specific for each prototype.

Figure 5.1 presents an image of the list prototype and 5.2 presents an image of the object prototype. In Appendix B there are more images.
Figure 1: Image of the list prototype.
Figure 5.2: Image of the object prototype.
The content of the prototype is split into four views: Spaces, Other, Compare spaces, and Hosts. To navigate between the views there is a menu in the top-left corner, displayed when hovering with the mouse pointer. The name of the view is also presented in the top-left corner. The settings icon is in the top-right corner.

The view **Spaces** presents the products sorted by space. There are four spaces presented as tabs: Production of type production, Test and Education of type test, and Development of type development. The preselected space is Production. Only parts of Production are interactive in the prototypes. To know which parts the user can click the gray area on the left side of the prototype labeled Highlight. This makes all interactive parts yellow. The user can also switch to space Test, but nothing is interactive in it. The tabs are ordered by type: production, test, development. The order should be interchangeable, but this is not possible in the prototypes. The default sorting of the products should be as in the LCM, were the products handled by the Grid are inside the Grid, and the order inside and outside of the Grid is alphabetic. It should also be possible to sort the products according to alerts and personal choice. All unnecessary information should be possible to be hidden, as well as the tooltips that might be annoying after a while.

The view **Others** presents the products and middleware that do not belong to a space. This is done to resemble the logic in the LCM’s user interface. The view **Compare spaces** presents all the products in two spaces at a time, together with the versions. The view **Hosts** presents all the hosts, together with the products and what space it belong to for each host as well as some information about the host.

To be able to verify that the object prototype would fit in a tablet screen, I adjusted the font size to be large enough for a user to be able to interact with it on an 11 inch screen. Unfortunately the font size looks excessively large when presented on a larger screen.

To describe the design solutions the model by Arvola & Artman (2007) is useful. Below are the five elements described one by one. The design concept, and the function & content are the same for both prototypes. The structure, interaction and presentation are partly the same and partly different and therefore described both in general and for each prototype.

### 5.3.1. The design concept
The purpose of the dashboard is to provide relevant information from several sources in a structured and simplified way for both internal and external customers. The purpose of the prototype is to gain knowledge about if the customers find a dashboard useful, both in terms of the information it presents as well as how it is presented and how the prototype can be improved.

### 5.3.2. The function & content
The information presented in the dashboard is:

- The view **Spaces** presents the spaces, with name, type, a list of the hosts and the databases that it uses, and an aggregation of the status of the space.
- The products for each space, with name, installation name, product name, version, status, if there are any patches available, the number of warnings and errors in the log file, whether the product is shared with another space, a link to the management page, the log file, and the property page in the Grid’s web user interface, what hosts it runs on, and which database and other products it is connected to. There is also space for
additional, product specific information. Only M3BEEnvironment and the Grid have additional information presented as these two are the most prominent products.

- For the hosts the prototypes present the external address, the internal address, the type, the version, and the status.
- For the databases the prototypes present the name of the database and the schema, the type, and the version.
- The view Hosts presents the products for each host, together with the space the products belong to.

The functions are:

- The user is able to start/stop the products.
- The user is able to open a new browser window with the Grid’s web user interface by clicking a link to the management page, the log file, or the property page.
- The user is able to remove available updates from the list, and reload them again.
- The user is able to clear the warnings and errors for each product.

5.3.3. The structure
The user can move from each view to any of the other by using the menu in the top left corner. In Spaces the user can also move from each tab to any of the other tabs.

List prototype
In each view the structure is very flat. For example in Spaces, the user can select any product in the selected space or go to another space, no matter which product is selected.

Object prototype
The structure in the object prototype is a little deeper. To view more information the user must click on the corresponding icon, and the new information pops up in a new window. The windows should be movable, if they hide any of the other information, but this is not possible in the prototype.

5.3.4. The interaction
The user switches between the views by using a menu up in the left corner. The menu is only displayed when hovering with the mouse for it not to disturb the view. Few (2013) suggests that the navigation is not up in the left corner but since most users are used to have the menu up in the left corner this was a nice compromise. The user switches between the tabs by clicking another tab. When clicking the status of a product the user can chose to start/stop the product or open the management page in a new browser window. When the user double clicks the name of a product a new browser window opens with the management page. When the user clicks on a warning or error digit the user gets the option to clear the counter.

List prototype
In Spaces the columns in the application table can be hidden by clicking the header and chose which column to hide. They can be reopened the same way. The tables can be collapsed by double clicking the header of the table. The list can be sorted with the different columns by clicking the column’s header. The order of the products can be changed by drag-and-drop. The new order can be saved with the save button below the table. Drag-and-drop is however not implemented in the prototype. The order can also be changed by opening settings and choose one of the predefined orders: alphabetic, alerts or personal. Alphabetic changes the order of the
products to pure alphabetic, with no regard to whether the products are handled by the Grid or not. Alerts shows the products with alerts at the top, and Personal switches to the last saved order. In settings the user can also turn off the tooltips and hide all tables except the application table.

**Object prototype**
In Spaces the boxes can be moved by drag-and-drop and the order saved with the button in the bottom-right corner. Drag-and-drop is however not implemented in the prototypes. The container for the products handled by the Grid can be expanded/collapsed and resized by click-and-drag. The order can also be changed by opening settings and change it to one of the predefined sorting orders: groups, alerts, or personal. Groups sorts the products into some kind of logical groups, Alerts sorts the products with alerts to the left and the rest to the right and Personal sorts the products as the last saved sorting order. In settings the tooltips can be turned off and the optional information such as status and version can be hidden.

5.3.5. **The presentation**
The icon for status is borrowed from the LCM’s user interface. The icon for settings is borrowed from Infor’s user interface guidelines. The exclamation mark for new updates placed next to the version is chosen by me.

**List prototype**
In the list prototype the products in the selected space is listed in a table together with the version, the status, if there are any updates available, and if there are any warnings or errors in the log file. To the right of the table, information of the selected space is presented. At the top there are two rectangles, one for general information and one for additional information. Below the rectangles there are two tables, one listing all the hosts the space use and one listing all the connected databases. When clicking a product, its name gets bold, the background behind the name turns gray, and the information to the right changes to information about the product. Below the host table a table with the connected products is displayed and below the database table a table with available updates is displayed. Below the table with the updates there is a rectangle with room for a description of a selected update.

The icon for open/close an entry in the table is borrowed from the LCM’s user interface. The arrow that shows next to the header of a column when sorting the table by that column is borrowed from Infor’s user interface guidelines.

**Object prototype**
In the object prototype the products are presented as boxes to symbolize objects. Inside the box there is an abbreviation of the name of the product, the version, the status, a digit presenting the warnings or errors, if there is any update available, and a plus-sign for additional information. If clicking the plus-sign the additional information pops up in a new window. Inside the additional information window there is information about the associated product, a symbol for hosts and for databases, and links to the Grid’s web user interface. The hosts and database symbol opens up a new window with a list of the hosts or the databases.

The icons for database, schema, and host are borrowed from the LCM’s user interface. The icons close, information, and shared are borrowed from Infor’s user interface guidelines. The icons
chosen by me are a plus-sign for additional information and an arrow to expand/collapse the container of the products handled by the Grid.

5.4. The evaluation process
Ten respondents were selected, presented below. Half of them got the possibility to get familiar with the list prototype and half the object prototype before the evaluation. The evaluation consisted of both the System usability scale and semi-structured interview questions about the prototypes usability. The evaluations were performed with telephone and WebEx because of the geographical distance. WebEx is a conference tool that lets the participants share desktops with each other.

5.4.1. The selected respondents
The selection of respondents resulted in four employees from Services, three from Support, one from Demo Environment, one from Sales, and one external customer, ten in total. The four from Services have the roles as Technical Project Managers, System Consultants, or Subject Matter Experts. From Support one respondent was selected who work in Technical Support. The respondent from Technical Support recommended two Functional Analysts from Support as well. Of the other eight respondents, that were not Functional Analysts, six had participated in the initial interviews, and the last two were contemplated but were too busy to participate. One more from Services participated in the initial interviews but was too busy to participate in the evaluation.

5.4.2. How the evaluations were conducted
The respondents were contacted by e-mail and told that the interview would be recorded and that the recordings will be destroyed when the report is finished. They were also told that they will be anonymous in the report so that external people will not know who has participated in the interviews. Internal employees might still be able to figure it out. The emails are added in Appendix A.4. The selected citations in the Result chapter were confirmed by the respondents.

To be sure that half of the respondents viewed the list prototype first and half the object prototype, every other respondent was demonstrated with the list prototype and the others the object prototype, in chronological order according to the time of the first session of the evaluation with each respondent. For the Functional Analysts in Support I made sure one of them viewed the list prototype first and one the object prototype first. For the rest, three from Services and the one from Technical Support viewed the list prototype first and the rest, one from Service, the one from Demo Environment, the one from Sales, and the external customer, viewed the object prototype first.

For the respondent to be able to get familiar with one of the prototypes before they evaluated them, but still evaluate the first one before they viewed the second one, the evaluation was split into two sessions. On the first session, which lasted for about 30 minutes, the purpose of the dashboard and some background information about the prototypes were presented. Then the first prototype was demonstrated and the respondent was able to try it out in their web browser. Between the first and the second session the respondent was asked to test the prototype a few times. They were also asked to consider the information presented in it, and reflect on how the dashboard can help them in their work.
In the second session of the evaluation, which lasted for about 60 minutes, the respondents were asked to first answer a few questions about themselves. Then they graded the ten statements in the System usability scale for the first prototype and answered the interview questions about the first prototype. After that they got a demonstration of the second prototype, graded the same ten statements for the second prototype and answered the same interview questions.

60 minutes was quite short to be able to evaluate the first prototype and both demonstrate and evaluate the second. However, the demonstration and the interview questions for the second prototype were shorter than the first, which was natural since both prototypes presented the same information. Especially the questions about what information was missing, where to find it in the Grid’s user interface, and how it is used in their work took much time.

5.4.3. Background questions
The interviews started with a few background questions about the respondents, to get some basic knowledge about them. The questions was about what department they work in, how long they have used the ERP system, how old they are, and how familiar they are with the LCM’s and the Grid’s user interface. Questions about how familiar they are with the LCM and the Grid are interesting since it describes if the user is familiar with their user interfaces, and if they find the information useful in their work. The questions can be found in Appendix A.1.

5.4.4. System usability scale
For the quantitative part of the evaluation, the respondents were sent an email with the statements so that they could read the statements but they graded them verbally over the phone. SUS can be graded on a 5 or 7 point scale. The version with a 5 point scale is the one explained in Brooke (1996), and was used in this study. The email with the statements can be found in Appendix A.2.

5.4.5. The interviews
The interview questions were chosen based on the purpose of the evaluation. To make sure that all relevant parts of the design solution were evaluated, the model by Arvola & Artman (2007) was used. The design concept, i.e. the purpose and the intended use of the dashboard, were relevant to validate, if the different users found the dashboard useful or not. The function and content element resulted in questions about the information and the functions. The structure resulted in questions about the different views. The interaction resulted in questions about the navigation between the views, how to access additional information, the possibilities to use drag-and-drop in the dashboard (which was not implemented in the prototypes), and the possibilities to customize the view. The last element, presentation, resulted in questions about the layout of the information, the terminology, and the icons. A question about the possibility to add two roles for the dashboard, one admin and one viewer was added, since it is interesting if all users should be able to view all information and use the functions. In the end of the questions about the second prototype the user was asked which prototype he or she preferred and why. The questions were reviewed by both the supervisor of the study and my mentor. The interview questions can be found in Appendix A.3.
6. Result

This chapter presents the result of the evaluation, i.e. both the quantitative results from SUS, and the quantitative and qualitative results from the interviews. From the quantitative results a few more measures are calculated, with some basic data analysis. First the quantitative results are presented, then the values from the analysis of the results. This is divided into two parts, one related to the first research question and one to the second. Last the qualitative results from the interviews are presented.

6.1. The quantitative results

The quantitative results from the grading of the statements in SUS and the answers to the last question “Which prototype do you prefer?” in the interviews are presented in the two tables below. One table presents the results for the group of respondents that got familiar with the list prototype, group 1, and the second presents the results for the group that got familiar with the object prototype, group 2. The first column in the tables displays what team the respondents work in, the second and third presents the SUS score for the first and the second prototype, the fourth is which one they preferred according to the score, and the fifth is which they preferred according to the interviews. Most respondents felt strongly for one of the prototypes, but one respondent commented that the preferred prototype just won by an inch and one wanted both. The comments are presented in the sixth column. In the last row in each table the average score is calculated.

During the grading of the statements in SUS some respondents graded some statements between two points. In these cases I used the middle value. If the respondent answered 1-2, I used the value 1.5. This should not affect the reliability of the result since the scale is linear. If the example above was the grade of statement 2 the value for grade 1 would be 5-1 = 4, for grade 2 would be 5-2 = 3, and the value for grade 1.5 would be 5-1.5 = 3.5. The value contribute to the total score as value * 2.5, which in the three cases above would be 4*2.5 = 10, 3*2.5 = 7.5 3.5*2.5 = 8.75. The contribution for the grade 1.5 is right in the middle of the contribution for the grade 1 and the grade 2.

The first table presents the results from the group of five respondents that viewed the list prototype first and got a chance to get familiar with it.

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Score SUS</th>
<th>Preferred</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>List</td>
<td>Object</td>
<td>SUS</td>
</tr>
<tr>
<td>FunctionalSupport2</td>
<td>83</td>
<td>43</td>
<td>List</td>
</tr>
<tr>
<td>Service2</td>
<td>75</td>
<td>50</td>
<td>List</td>
</tr>
<tr>
<td>Service1</td>
<td>86</td>
<td>58</td>
<td>List</td>
</tr>
<tr>
<td>TechnicalSupport</td>
<td>83</td>
<td>65</td>
<td>List</td>
</tr>
<tr>
<td>Service4</td>
<td>85</td>
<td>45</td>
<td>List</td>
</tr>
<tr>
<td>Average</td>
<td>82</td>
<td>52</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: The result from respondents in group 1.

The second table presents the results for the second group of five respondents that viewed the object prototype first and got a chance to get familiar with it.
According to Bangor (2008) it is not meaningful in general to view the result of each statement separately. I believe, however, that the grading of the first statement “I think that I would like to use this system frequently” gives a hint about whether the respondent would use the dashboard or not. Many of the respondents have also commented while grading the statement why they chose the grade, which makes the statement important. I have therefore chosen to present the grading for that particular statement in the tables below.

### Grading of statement 1

<table>
<thead>
<tr>
<th>List</th>
<th>Object</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionalSupport2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Service2</td>
<td>4</td>
<td>1-2</td>
</tr>
<tr>
<td>Service1</td>
<td>3 (5)</td>
<td>3</td>
</tr>
<tr>
<td>TechnicalSupport</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>FunctionalSupport1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>DemoEnvironment</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Service3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ExternalCustomer</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Sales</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 6.3: The grading of statement 1 for all respondents.

Except for the Functional Support analysts, seven of eight respondents agree or strongly agree that they would use the list prototype frequently. The eighth respondent would also strongly agree if the dashboard got improved with a few small changes.

### 6.2. Analysis of the quantitative result

From the quantitative result presented in the tables in the beginning of the chapter, some more quantitative values connected to the research questions have been calculated. First the values connected to question 1 are presented in two tables, one for the values from SUS and one for the values from the interviews. Then the values connected to question 2 are presented analogously.

**Question 1 – Do the users prefer a list-based or an object-based dashboard?**

The table below presents the calculated values, based on the result from SUS.

<table>
<thead>
<tr>
<th>SUS</th>
<th>Absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many preferred the list prototype?</td>
<td>7 of 10</td>
</tr>
<tr>
<td>How many preferred the object prototype?</td>
<td>3 of 10</td>
</tr>
<tr>
<td>How many thought the list prototype had low usability (&lt; 70)?</td>
<td>1 of 10</td>
</tr>
</tbody>
</table>
6. Result

| How many thought the object prototype had low usability (< 70)? | 6 of 10 |
| What was the total average score for the list prototype? | 80 |
| What was the total average score for the object prototype? | 64 |

Table 6.4: The values connected to the first research question, calculated from the result of SUS.

The table below presents the calculated values, based on the result from the interviews.

<table>
<thead>
<tr>
<th>Interviews</th>
<th>Absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many preferred the list prototype?</td>
<td>7 of 10</td>
</tr>
<tr>
<td>How many preferred the object prototype?</td>
<td>3 of 10</td>
</tr>
<tr>
<td>How many disliked the list prototype?</td>
<td>2 of 10</td>
</tr>
<tr>
<td>How many disliked the object prototype?</td>
<td>6 of 10</td>
</tr>
</tbody>
</table>

Table 6.5: The values connected to the first research question, calculated from the result of the interviews.

The calculated values connected to the second research question are presented below.

**Question 2 – How does the possibility to get acquainted with the prototype before the evaluation affect the result?**

The table below presents the calculated values, based on the result from SUS.

<table>
<thead>
<tr>
<th>SUS</th>
<th>Absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many preferred the prototype they got familiar with?</td>
<td>8 of 10</td>
</tr>
<tr>
<td>How many preferred the list prototype even though they did not get familiar with it?</td>
<td>2 of 5</td>
</tr>
<tr>
<td>How many preferred the object prototype even though they did not get familiar with it?</td>
<td>0</td>
</tr>
<tr>
<td>How many thought the list prototype had low usability (&lt; 70) of those who got familiar with it?</td>
<td>0 of 5</td>
</tr>
<tr>
<td>How many thought the object prototype had low usability (&lt;70) of those who got familiar with it?</td>
<td>1 of 5</td>
</tr>
</tbody>
</table>

Table 6.6: The values connected to the second research question, calculated from the result of SUS.

The table below presents the calculated values, based on the result from the interviews.

<table>
<thead>
<tr>
<th>Interviews</th>
<th>Absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many preferred the prototype they got familiar with?</td>
<td>8 of 10</td>
</tr>
<tr>
<td>How many preferred the list prototype even though they did not get familiar with it?</td>
<td>2 of 5</td>
</tr>
<tr>
<td>How many preferred the object prototype even though they did not get familiar with it?</td>
<td>0</td>
</tr>
<tr>
<td>How many disliked the list prototype of those who got familiar with it?</td>
<td>0 of 5</td>
</tr>
<tr>
<td>How many disliked the object prototype of those who got familiar with it?</td>
<td>1 of 5</td>
</tr>
</tbody>
</table>

Table 6.7: The values connected to the second research question, calculated from the result of the interviews.

### 6.3. The qualitative result

The qualitative result from the interviews is presented below. It is divided into the following parts: why some preferred the list prototype and some the object prototype, the respondents’ general thoughts of the dashboard, if any result differs for the different user groups, and lastly the suggestions for improvements divided into the same categories as the interview questions. Some of the citations supporting the results are translated from Swedish.
6.3.1. Why some prefer the list prototype
Those who prefer the list prototype give the following reasons: it is intuitive and easy to read off, it displays all the relevant information at the same time, without being cluttered, and many users are more used to view the products in a list than as objects.

Those who prefer the list prototype think that the object prototype is perceived as more cluttered and complex, less intuitive, that the boxes and the information is jumping around, and that it is cumbersome to have to click much to be able to view all information needed. Some think that it might be acceptable as a compromise when shown on a tablet, which has a smaller screen than a desktop computer. Worth noting is that only one of the respondents who got a chance to get familiar with the object prototype disliked it.

6.3.2. Why some prefer the object prototype
Those who instead prefer the object prototype think that it displays less information and is clearer, it only displays the information the user is interested in right now. The object prototype is better to symbolize a system, the objects symbolize the programs running next to each other. Pop-up windows are better than moving away to another view, as in the Grid’s user interface, but to many pop-ups will make it cluttered. Worth noting is that only respondents who had the chance to get familiar with the object prototype liked it.

Those who prefer the object prototype think that the list prototype looks too much like the LCM, it is less interesting. It displays more information than needed, and becomes cluttered with many rectangles and tables when it tries to present all information at once; one does not need to view all the information at the same time.

6.3.3. The use of a dashboard
All the respondents are positive to a dashboard as a complement to the existing tools, and believe that they would use it frequently in their work. The respondents like the overview of the system, that they can view all the essential information at the same time.

Citation from Service3, about the object prototype:

It is an easy way to view almost all the things you have in the installation with a simple quick view. You see everything in a quick view, ... the status, if there are some warnings and so on. It is really nice to see if everything is up and running or if you have problems. And if you want to go deeper, you have some tools that get you deeper, but not too deep. You don't need to be a system administrator to check the things you are showing in the dashboard. I think it is quite easy for everyone to understand what is inside the dashboard ... If you go to the LCM, the LCM is more complex ... When you are going deeper into the LCM, you need to be system engineer ... to know all the things you have inside. With the dashboard you don't need that. So I think it is really nice.

Citation from Service4, about the list prototype, translated from Swedish:

It is often that I log on at new customers. The first thing I do is of course to open the LCM and start looking around to see what they
have installed and of what versions, how it fits together, what hosts, and so on, to get an idea [of the system]. I think that this dashboard can help me with this. Similarly during the installation, that you have one place where you quickly can get an overview of what has been done and what versions that are installed and where.

Citation from ExternalCustomer, about the list prototype, translated from Swedish:

From what I have seen I answer “strongly agree” [on statement 1 for the object prototype], of course it is [just] a mockup ... I think what has been done is good. Then, as I said, a first version is always a first version, you will always find things that can be even better. But I definitely think that this is something that you will use.

Citation from TechnicalSupport, about the object prototype, translated from Swedish:

Technical Support analysts are almost all used to work with LCM themselves. Then you have the functional users [functional support analysts], and it is getting more and more important that you are able to handle the LCM and the information there in. ... My thought was that this [the dashboard] is very useful for those that are not that technically skilled. What you have shown me so far is in principle, almost to the point, what I was expecting.

For many of the users it is important not only to be able to view an overview of the system, it is also important to be able to drill down in the information, and that it supports the daily monitoring of the system.

Citation from DemoEnvironment, about the object prototype:

[regarding statement 1 for the object prototype] If I could drill in [get deeper information] and actually do something [i.e. be able to go to the error-log as opposed to have to go to the LCM to be able to apply a fix], it would be something I would use a lot. ... But if it is more a case of showing you things but you still have to ... get into the LCM at the same time, then I would probably use it for an overview, look for issues, look for the red squares, look for errors, but then you would go to the LCM to actually do something about it.

Citation from Service1, about the list prototype, translated from Swedish:

[regarding statement 1 for the list prototype] That you can start/stop, that you can check the logs, that you can have warnings if there are errors, or if a certificate is going to expire ..., then I strongly agree [with statement 1 about the list prototype]. If any of the system functions do not work, if you can’t lock or put a subsystem offline, or kill a job, or something like that, from the dashboard by going to the Application pages ..., then I believe it is less usable. Then it is more that you go in
and look for information, but if you need to do something you use the LCM.

Most of the requested drilling, i.e. get more detailed information, is supported by the links to the Grid’s user interface.

Most of the respondents think that the information, which the prototypes present, is almost enough; if adding more information there is a risk that the purpose of the dashboard is lost.

Citation from Service1, about the list prototype, translated from Swedish:

... it needs to be quite sparse, so that it won’t end up like the Grid, were you drown. So, do not cram in any more.

Since the users use the dashboard for different reasons it also needs to present information that not all users consider relevant.

Citation from Service1, about the list prototype, translated from Swedish:

... all [users] that use it [the dashboard], use it for different reasons, and are therefore in need of different information, ...

They also believe that it is enough to be able to view the information, not being able to manipulate it.

Citation from Service4, about the list prototype, translated from Swedish:

... this is a dashboard, you are not supposed to be able to do a bunch of stuff with it, you are only supposed to ... absorb information, you should not be able to manipulate it in this interface, ...

There are different opinions if the additional views, i.e. Compare spaces and Hosts, are useful or not. Some consider Compare spaces very useful, especially when upgrading the system, some do not as long as it does not display more detailed information, primarily about M3BE. Some consider the host-view very useful, to present all the hosts the system use, and what products run on a specific host, some do not see the point in it.

There are also different opinions about whether or not the customization of the dashboard is necessary and if the proposed possibilities are enough or not. Some consider it important to be able to customize it for the user’s different needs, some think that it is fine as it is, that it just makes it more complex and difficult to implement. Some would like to be able to customize the different spaces and/or products differently. For a customer it is not interesting to see if there is an update available in the space Production, especially not every day, but it is interesting to see if there is a critical notification. In the space Test, however, it is interesting to see if there is an update available, that they can apply and test before it is relevant to apply it in Production.

Many of the respondents believe it is useful to use the same logic as in the LCM. They also like to be able to view the available updates and to be able to clear the error counter per product, instead of clearing the counter for all products.
Some of the respondents would like some more details, primarily about M3BE such as applied fixes, and some would like for the dashboard to present the customer’s whole system, including all third party products. Many of them also believe it would be useful to be able to print or export the overview, to be able to add their own additional information or note about some product, and to connect the available updates and perhaps the logs to the new Knowledge Base, which the Support team is creating. They also had a few comments on the symbols and the possibility to clear the error-counter. The suggested improvements are presented in detail later in the chapter.

It is not only interesting what the respondents have expressed during the interviews; it is also worth noting some of the things they have not expressed. None of the respondents dislike the dashboard. None of the respondents have supported the idea to have the dashboard in a tablet or smart phone. Many of them are however unfamiliar with tablets. The respondents seem to be satisfied with the level of monitoring the prototypes support, with a few minor changes presented later in this chapter.

6.3.4. The different user groups
The respondents believe that the dashboard will be useful for different user groups. The suggested groups are those of the external customers’ employees that are responsible for the system, and for internal employees at Services, Support, DemoEnvironment, and Sales. All these groups are represented by the respondents.

The different user groups use the dashboard for different reasons, and may therefore have different needs of information. Each type of user group and their needs is presented below.

Services
The dashboard is useful for several roles in the Service team, such as Technical Project Managers, Systems and Business consultants, Technical experts, and Technicians/Developers. Technical Project Managers, Systems consultants, and Technical experts are represented by the respondents. They think that the dashboard is useful both in the implementation phase, during the installation of the system and the troubleshooting, and to be able to get a quick understanding of the customer’s installation later when configuring or upgrading the system. During the troubleshooting it is useful to be able view the log files and to go to the Management page to be able to execute, kill, or monitor individual jobs and subsystems in M3BE.

All from Services liked the list prototype, since they are used to view the products in a list, and that it displays all the relevant information in the same view at one click. One of the respondents wanted both prototypes and preferred the object prototype, since the respondent does not need to see all the information at once.

Technical Support
Many of the analysts at technical support are able to handle the LCM, and often need to see more information than the dashboard presents. According to the respondent from Technical-Support it is primarily the Functional Support analysts that would have use of the dashboard, and perhaps a few of the Technical Support analysts. However, TechnicalSupport states that the dashboard gives a nice overview of the system and graded the first statement in SUS with a 4.
**Functional Support**

The functional support analysts do not use the LCM that much. They mostly use the Management page (formerly the Server view) for M3BE in the Grid’s user interface to view information. They use the Management page for functions such as starting the debug, find class to show the fixes applied on a program, to view the dump log and the error log, see if the subsystems are running, and to view the properties. The overview of the system is nice for the understanding of the system but do not help them that much in their work. They need deeper information about M3BE, which can be found in M3BE’s Management page. With the information the dashboard present in the prototypes, it would be sufficient to use it as an entry point to the Management page, according to FunctionalSupport2. If some of the information they currently store in an MS SharePoint page was to be added as additional information in the dashboard, it would be nice to use the dashboard in their work.

Citation from FunctionalSupport2, about the list prototype, translated from Swedish:

> If it [the additional information for M3BE] would be supplemented [with information from the share point page] I would like to find the information easily in this product instead of digging down in the grid and the server view [the Management page] and in the property file, and so on.

It would be useful for them to also be able to see what kind of error the error-digit represents, if it is from the application or from the dump-log. If the error is in the dump-log it would be useful to see which program it belong to and what user-id that is responsible for it.

**Demo Environment**

For the Demo Environment team the type of hosts and databases are not relevant since they use the same type of virtual machines. The additional information, such as the ports and URLs, the link to the properties, and the available updates is much more important. The dashboard is probably also useful for the other Environment teams that handle the other internal environment, such as test and development.

**Sales**

Sales would like a less complex interface, which would be easier to present to less technically skilled new customers that want to get a basic understanding of the system. Sales prefer the object prototype since it presents a clear, clean view of the system. It would be even better if all extra information was moved from the first view to the additional information about the products, such as the version and available updates, and just keep the status and maybe the errors/warnings. The look of the object prototype resembles the look of Windows8 and tablet applications and the respondent from Sales think it is more modern than the list prototype. The respondent considers the list prototype only as a new version of the Grid’s user interface, not a complement to the existing tools.

Many external customers would like to go as far as having a traffic light for the monitoring of the system.

**6.4. Suggested improvements**

The respondents suggested some improvements for the prototypes, listed below. The suggested improvements that apply to both prototypes are divided into the same categories as the
interview questions. The suggested improvements that only apply to one of the prototypes are listed after the other categories.

**Information:**
1. Add details about M3BE such as applied functional fixes.
2. The version number of the product displays which fixes that have been applied to the product but it can be difficult for less technically skilled people to know if there has been a cumulative fix/update (update 2 includes update 1) applied. This could perhaps be solved by adding an explaining text in combination with the version number.
3. In the prototypes the M3BEEEnvironment available updates has been crossed out, since it was not relevant to present available functional fixes, which are available all the time and many are not interesting for the customer to apply. However, M3BE Foundation that is included in M3BEEEnvironment has updates and perhaps also Tools & Runtime has updates, which should be displayed in the dashboard. For the functional fixes, the dashboard could show when there is a feature pack available, which includes several functional fixes at once.
4. For support and customers to check if one of the latest applied updates has caused a problem, it could be useful to add a list of the five latest applied updates, with a time stamp when they were applied, to the list of available updates.
5. Additional information: Information that might be relevant to add to the Grid: Add more ports, perhaps one page with all standard ports the Grid use and a second with all ports. For M3BE: the API-port, the ports for the views and for net extension data. MAK-port is not relevant for many customers, could perhaps be removed. M3BE has dump logs that could be useful to reach from the dashboard. For MEC: the MEC-port.
6. Several of the respondents have explained that many customers consider their third party products, which they use together with M3, as belonging to the system. They would therefore like to be able to view the whole system in the dashboard, either by manually adding them so that just the some basic information is presented in the dashboard or that the dashboard itself fetches information about products that the Grid knows about, such as Vertex or ADC.
7. Compare spaces: Add which feature packs and functional fixes that have been applied to M3BE. Add the heap size to the applications.
8. Host view: Add how many processors are currently running, and how many jobs, on each host.

**Functions:**
1. It should be difficult to accidentally stop an application, it is easy to slip in the menu when one aims to go to the Management page instead of stopping the application. One solution is to remove the link to the Management page from the menu so the user only opens the menu when it aims to start/stop the application. Another is to add a notification, asking if one really aims to stop it, and make “No” the default answer.
2. Support has started to add information about problems to a Knowledge Base. Today it does not support available updates, but it is supposed to be added soon. When this exists it could be useful to connect the available updates to the Knowledge Base, by adding a link that opens the Knowledge Base with the right fix identifier added.
3. Many of the respondents want to be able to print or export an overview of the system, as PDF or text or as XML so it can be imported to Excel. It is possible in the LCM, but it
could be useful to be able to print an overview with fewer details than in the LCM. For example an overview of each space, with the Grid, what hosts it uses + Java version of the LCM service, and which products it handles + version.

4. Add a notification when certificates are going to expire, preferably with an email sent to a selected address.

5. Add the function that the dashboard sends an email to a selected address when something is wrong.

6. When the LCM service on a host is down, it could be useful to be able to ping the server to check if the server is up and running.

7. Add a list of the latest errors/warnings with a short description so that the user immediately can know if there is something they need to respond to right away. It might take some time to open a new browser window with the log file in the Grid’s user interface, 30 seconds is too long just to have a quick check. It would also be useful to be able to remove the errors/warnings one by one, and keep those that need a more thorough check. If the Knowledge Base support errors and warnings it could be useful to connect them by adding a link that opens the Knowledge Base in a new browser window.

8. Be able to add your own additional information for each product. Since different users are in use of different ports and URLs etcetera, it would be useful to have some default information that is most people use but also to add the possibility for the user to add his or hers own important information. It could also be useful to add small notes for each product, for example about how a problem was solved. If the information is long, it would not fit in the dashboard but perhaps it could pop-up in a new window or the user could add the link to where the file is stored.

9. There are more dependencies between the products than there is information about in the LCM. Some respondents would therefore like to be able to add more products to the list of connected products, products that will affect it or depends on it.

Views:

1. Spaces: In the database table for the space it could be useful to add information about to which product the database is connected.

2. Compare spaces: Add an icon that signals if the version differs for a product.

3. Hosts: Add the Java version for the LCM-service at each host in the host view. The Java version is useful to know when upgrading the LCM, which has some prerequisites.

Presentation:

1. Clickable information should have another color than pure information, such as blue links. One way is to change the color when hovering with the mouse.

2. Not everybody likes a white background color, or feel comfortable to read from it. It would be useful to be able to change the background and foreground color or change between some color themes.

3. Terminology: Change props to properties.

4. Icons: The share-icon does not mean shared, it means share, as on Facebook or Twitter. Exclamation mark is not suitable for available updates, but for errors, which one needs to address directly. Some would like the exclamation mark for the aggregated errors or warnings, since they don’t find it interesting to know the number, other likes to know the number and prefer the digits. One could also use a red warning triangle for errors.
5. Customizations: The user should be able to reset the customization to default. It could also be useful to customize the Spaces and/or products differently. Available updates is useful for Test and Development but not for Production for end-users. Critical notifications are however important for Production. Some products produce too many errors and warnings that it is not relevant to view.

Interaction:
1. It would be useful to be able to click alt+tab when switching between the spaces.

Other:
1. Where will the dashboard be installed? One suggestion is that it is uploaded to the LCM’s admin page and is available from the LCM server’s http-server.
2. The dashboard should be possible to open inside the LCM as with the Grid’s user interface, even if it is a web-page, since the LCM is the hub in the management of M3.

Suggestions that only apply to the list prototype
1. Add the aggregated status for the space, so that one can see it for all spaces all the time, as in the object prototype.
2. If the list of hosts is too long so that they can’t be viewed in the window, there might be useful to be able to click and open a larger pop-up window.
3. If the list of applications is long it might be useful to be able to filter or search in the list.
4. One should avoid needing to scroll in lists if possible. This can be a problem for the list of applications, which might get long for some customers with many products installed.
5. For those not interested in hosts and databases it would be useful to add the possibility to be able to add a table with the five last logs instead of host and database tables.

Suggestions that only apply to the object prototype
1. Add tool tips for the abbreviations. It could be useful to be able to change the tool tip for the abbreviations, since many old customers have their own naming standard.
2. The highlighting of the connected products is too weak.
7. Discussion

In this chapter the result of the evaluation is discussed. The discussion first answers the research questions: which prototype the respondents preferred and if the possibility to get familiar with the prototype affected the result. Next it discusses if the respondents would use any of the prototypes in their work, and how the usage would differ between the different user groups. After that it discusses if the respondents thought the prototypes had an acceptable level of usability and if the prototypes fulfilled their purpose. Then the most important improvements to implement in the prototypes are discussed along with to the reliability and the generalizability of the result. Lastly relevant future work is discussed.

7.1. Do the users prefer a list-based or an object-based dashboard?

Both SUS and the interviews give the same answer, seven out of ten respondents preferred the list prototype. Of these seven one liked the object prototype too. Of the three that preferred the object prototype one liked the list prototype too.

One of the two primary reasons that the respondents prefer the list prototype seems to be that they are more used to view the products in a list, that it resembles the LCM’s user interface. None of the respondents that did not get familiar with the object prototype liked it while four out of five of those who got familiar with it liked it, and three of these four preferred it. Three out of five of those who got familiar with the object prototype still liked the list prototype, and two of these three preferred the list prototype. Many of the respondents that preferred the list prototype has expressed that they are more used to view the products in a list, and that they are not used to tablets and tablets applications. On the contrary two of the three that preferred the object prototype expressed that it is because it has a new and fresh look, and all three think that it displays enough information about the system. One conclusion is that most users will like the list prototype.

The second primary reason is that the list prototype displays all relevant information at the same time. One of the three respondents that preferred the object prototype is from Sales and would only use the dashboard to present an overview of the system to new customers. The second respondent is from Functional Support, which according to a discussion later in this chapter might not be in the primary target group for the dashboard. The last respondent finds both prototypes useful and believes that only the object prototype will not do, even if the respondent would use the object prototype most of the time. A second conclusion is that the object prototype is not enough for some of the primary users; it does not display enough information at the same time.

Two of the respondents disliked the list prototype. One was Sales that thought the list prototype did not fulfill its purpose, that it was merely a nicer looking version of the Grid’s user interface. The other was a respondent from Functional Support who thought the object prototype was clearer, that the list prototype was perceived as cluttered, when it tried to present so much information at once. One way to solve it is to make it possible to make all tables collapsed by default for those users, so the user must actively expand the information it aims to display.

I still believe, however, that the object prototype is a good way to present an overview of the system. I also believe that it would have been more appreciated if the users would have been more used to tablets, which they probably will be in a couple of years. Likewise I believe that it would have looked more appealing if presented in a tablet or at least a bit zoomed out, so that
the font size would have looked more appropriate. During the evaluation the prototype, which was adjusted for an 11 inch screen, was presented for the respondents on a 14 inch laptop screen, which made the text look excessively large.

From a development perspective the object prototype might be trickier to implement since lists and tables are available in most graphical libraries. The list prototype’s design is also more tolerant if the customers’ installations differ much, i.e. what products they have installed and if they are handled by none, one, or several Grids.

It seems as if the users prefer a familiar look over viewing the products as objects, which according to Few (2013) is more natural. Familiarity seems to be a strong force.

According to the tables 6.4-6.7 the numbers are very alike between the result from SUS and the quantitative result from the interviews. What the numbers do not tell us is that they do not include the same respondents. For Sales and for DemoEnvironment the prototype with the highest score in SUS does not correspond to the preferred prototype stated in the interviews.

7.2. How does the possibility to get familiar with the prototype before the evaluation affect the result?

Eight out of ten respondents preferred the prototype they got familiar with. This result is however not as strong as it first may seem since seven out of ten respondents preferred the list prototype regardless of which prototype they got familiar with. So if all respondents would have gotten familiar with the object prototype, the number of how many that preferred the prototype they got familiar with would most likely have decreased. But the grading of the object prototype would probably have increased. The result is also a bit twisted since the respondents were used to the list-based concept before the evaluation.

Only respondents that had the possibility to get familiar with the object prototype liked it, and only one of the respondents that got familiar with the object prototype disliked it. Three of the four who liked the object prototype preferred it while the last one liked both but still preferred the list prototype since the respondent was more used to the list concept. It was probably easier to accept the list prototype quickly since it resembles the LCM. The respondents are used to view the applications in lists and tables, while the concept of using objects is new. Many of the respondents are not used to Windows8 or to tablet applications.

The only respondent that got familiar with the object prototype and that graded it lower than 70, which corresponds to not acceptable usability according to Bangor (2008), was one of the three that preferred it. The respondent that got familiar with the object prototype but still strongly preferred the list prototype graded it above 70. The two respondents who did not prefer it, of which one strongly preferred the list prototype, still graded the usability of the object prototype as acceptable. None of the respondents that did not get familiar with the object prototype graded it as acceptable.

The result of SUS for the respondents that got familiar with the object prototype is very even. One reason that these respondents graded the object prototype high can be that they liked the idea of a dashboard more than the design concept, while those who already had seen the list prototype rejected the object prototype and graded it very low. The respondents may also have been diplomatic because the interviewer had created the prototype. But when the respondent answered which prototype they preferred during the interview, the respondent had already
seen both prototypes and still four out of five liked the object prototype, and three even preferred it.

The result corresponds quite well to Karapanos et al.’s (2009) research about how user experience changes over time. If none of the users had got a chance to get familiar with the object prototype none had probably preferred it, and I am quite sure that if the respondents had been more used to tablets and some more had gotten familiar with the object prototype more had liked it and perhaps even preferred it.

The conclusion imposes some considerations when using parallel prototypes in a study. It is important to vary the order in which the respondents view and evaluate the prototypes, to get a reliable result.

7.3. Would the respondents like to use any of the prototypes in their work?
According to the grading of the first statement seven out of ten respondents agree or strongly agree to use the list prototype frequently. One of the three that was neutral (graded the statement with a three) would strongly agree if the dashboard was improved so that it indicated when a certificate was going to expire, preferably in addition to sending an email. The other two respondents that were neutral were Functional Support analysts. These are only interested in one product, M3BE, while the rest of the user groups are interested in the whole system. Functional Support analysts might therefore not be the primary target group.

According to the grading of the first statement also four out of five that got familiar with the object prototype agree or strongly agree to use it frequently. The fifth respondent is one of the Functional Support analysts.

The quantitative result speaks clearly that the dashboard would be used. The result is supported by the interviews. All respondents are positive to using the dashboard in their work. They find the overview of the system useful, and like to be able to view all the relevant information at once. Most respondents think that the level of the information is right, that too much detail does not comply with the purpose of it. They also believe that the purpose of the dashboard is to display information, not to change the information, and that different user groups will use the dashboard differently and therefore need it to display different information. Therefore it either needs to display more information than each user needs or it needs to be customizable. It is also important to be able to drill down in the information and that the dashboard supports some monitoring of the system. This is supported in the prototypes by the links to the Grid’s user interface, except for the need of an indication of when a certificate is going to expire.

In the interviews some of the respondents have also clearly expressed the need of a dashboard. However, it would also be interesting to hear how the customer would prioritize the dashboard relative to other features that are planned for the company’s products. Most users will say that they want a new feature that is presented to them. Prioritizing among possible features is another question.

7.4. The different user groups
External users and internal users from services and the environment teams will all have use of the dashboard. Some of the Technical Support analysts will probably also like it. Since the Functional Support analysts only are interested in one product they are maybe not the primary users of the dashboard, but according to the result of the evaluation they might still like to use it,
to get the overview of the system. Also Sales will have use of a cleaner and easier interface even if the respondent from Sales preferred the look of the object prototype, which was described by the respondent as more modern than the list prototype.

7.5. Do the users feel that any of the two prototypes has an acceptable usability?
Nine out of ten respondents thought that the list prototype had an acceptable level of usability, while only four out of ten thought that the object prototype had an acceptable usability, according to my interpretation of Bangor (2008). However, four out of five respondents that got a chance to get familiar with the object prototype thought it had an acceptable usability.

The average score for the list prototype from all respondents was 80, which corresponds to good usability according to my interpretation of Bangor (2008). For the object prototype it was only 64, but for those who got familiar with the object prototype it was 77, which is on the border between acceptable and good usability.

Many of the respondents have also expressed an appreciation of the prototypes’ design solutions during the interviews, which supports the result from SUS. They think that at least one of the prototypes present the information in a clear, clean way, and many think that at least the list prototype is intuitive and easy to use. It seems as if implementing Few’s (2013) guidelines worked out well.

7.6. Did the prototypes fulfill their purpose?
One of the reasons to use prototypes to evaluate the users’ needs was that some of the respondents thought, during the initial interviews, that it was difficult to fully understand the purpose of the dashboard, how it was supposed to be used, and who the target user group was. A contributing factor may be the relative lack of experience of the interviewer, in addition to the combined complexity of the system, the tools, and the user groups. As a consequence it was difficult for the users to answer what information they thought was relevant, and which user groups that would be helped by the dashboard. They thought it would be easier if they were presented with a prototype, which they could give feedback on. According to Sharp et al. (2007) the respondents can at least tell what they do not want if they are presented with a prototype. Prototypes can make a design idea concrete and make it possible for others to give feedback on it (Arnowitz et al., 2007; Verplank 1992 in Muñoz, 1992 p. 579). They can also be used to validate requirements and a design concept (Arnowitz et al., 2007).

During the evaluation I did not get that many complaints on the presented information. Most of the respondents seemed satisfied with the suggested one. They could more easily understand the purpose of the dashboard, how it would suit them in their work, that the suggested information conformed well with the purpose of it, and to identify relevant user groups.

The prototypes were only partly interactive, which made it a bit difficult for the respondents to test them but given the result I think the prototypes fulfilled their purpose well and therefore had the right level of details. The respondents thought that if they could test a fully functioning Beta version later they would be able to give more precise feedback, which corresponds well to the theory about prototypes.

Prototypes can also be used to explore an idea or a design concept (Arnowitz et al. 2007), and iterative prototyping produce better design (Dow et al., 2009; Sharp et al., 2007).
prototyping process made the possibilities much clearer for me and increased the level of the design, which corresponds well with the theory. Creating two prototypes instead of one helped me to explore two concepts instead of just one, and made it possible for the users to give feedback on both and to choose which one they preferred, which correspond to Dow et al.’s (2010; 2011) research.

The conclusion in section 7.2 implies that the respondents get attached to the first prototype they test. It is likely that also the designer who has created the prototype and put in much time and effort in it gets even more attached. It is therefore useful to use several prototypes from the beginning to not get stuck with the first prototype the designer creates, and instead explore several possibilities. This supports Dow et al.’s (2010) result.

7.7. Recommendation on suggested improvements to implement

Some of the suggested improvements are more suitable than others to implement. Improvements that are in line with the purpose of the dashboard, that are important for the use and or easy to implement should be added if the dashboard will be implemented.

Add the five latest applied updates together with a time stamp to the list of available updates, so the external customers and support analysts easier can analyze if a newly applied update might have caused the error. I also think it would be quite useful to view the five latest errors and warnings directly in the dashboard. It should also be possible to clear one error at a time. To make space for it in the list prototype, it might be useful to add two tabs for the right half of the space view in which the user can choose which tables to view in the primary tab and which ones to view in the secondary tab. In addition to this it would be useful to take advantage of the new Knowledge Base, and make it easy for the user to use it. It should not be too difficult to add a link to it for the available updates, and perhaps also for errors and warnings if it the Knowledge Base supports it.

Add the possibility for the users to add their own additional information for the products. It would also be useful for the users to add their own relations between the products in the connected products table. Another useful function is to add the possibility to print and export the data. Since expired certificates are very cumbersome to fix a warning should be implemented in the dashboard.

Remove the shortcut to the Management page from the status-menu and add a question if the user really want to start/stop an application, where the default answer should be No.

For the different views three small parts of information would be useful to add. For the space view the connected product in the database table would clarify which product the database is connected to. In Compare spaces an icon could signal if the versions differ for a product, and in Hosts view the Java version of the LCM-service should be added.

When it comes to the presentation of the information an interaction designer should give feedback on the colors, and Infor’s design guidelines should be implemented. The exclamation mark should only be used for errors, and the shared icon in the object prototype must be changed.
When the user customizes the interface there must be a possibility to reset them to default. An interaction designer should also decide how much information that should be displayed by default.

The aggregated status of the space should be presented in the tab for the list prototype as in the object prototype.

7.8. **Reliability & generalizability of the result**

The reliability of the study is limited since the number of respondents was small for a quantitative study and some of the heterogeneous user groups only was represented by one respondent. Also the criteria for the selection of the respondents can affect the reliability. Still the respondents were selected to represent the different user groups, not on whether they preferred list-based or object-based concepts. It would especially have been useful to have more answers from external customers to get a better understanding of their opinion.

The part of the result connected to the first research question is partly generalizable in the area of user interfaces for ERP systems. Buying an ERP system is both expensive and comprehensive. Also upgrading a functioning system to a new version takes some effort and is therefore done seldom. For ERP systems it is more important that the system is efficient and stable than that the user interface has a modern look-and-feel and is easy to learn. The users are often quite conservative and the result is therefore generalizable for this type of customer.

The part of the result connected to the second research question is more generalizable, even if the result was a little bit twisted because of the users’ familiarity with the list-based concept before the evaluation and the number of respondents was small. How the users’ experience of a prototype’s usability is affected by the possibility to get familiar with the prototype before the evaluation has little to do with what kind of prototype that is evaluated. The result is also supported by the research done by Karapanos et al. (2009).

7.9. **Future work**

Since the number of respondents was small it would have been interesting to perform an evaluation with the more respondents in each user group, to study the differences between them and to get a more reliable result. From an academic point of view it would be interesting to perform an evaluation where the respondents get more precise instructions on how much they are supposed to test the prototype they get familiar with before the evaluation and that both prototypes present concepts that none or at least few of the respondents are used to, so the result is less twisted.

The result of the evaluation will hopefully be used to implement a dashboard. If an implementation of the dashboard is successful, it would be interesting to study how later versions can be more customizable for different roles and if it could and should present information from more sources.

The result can also be used to improve the existing tools, primarily the LCM’s and the Grid’s user interfaces. One of the problems that have been noticed during the interviews is that some of the respondents have problems using the Grid’s user interface, that it both is hard to navigate in and that it presents too much information at once. The Grid’s user interface is under reconstruction which will solve much of these problems. The Grid’s user interface can however not give the user an overview of the system since it only represents one Grid, and the users have at least one Grid
in each space. The LCM on the other hand presents static information about most of the products. I believe that the users will find that the new LCM that was released in May 2013 gives a better structure of the products since it divides them into spaces. The LCM could of course support some more information by adding a few columns, for example available updates, some runtime information found in the Grid such as if there are any errors, but also the aggregated status of the spaces. But considering the purpose of the LCM, which is to make it easy to install and configure the system, it is not applicable to also make it present an overview of the system. Two different purposes should be supported by two different views, even if the dashboard of course should be possible to view inside the LCM as well as on a standalone homepage. The dashboard would be really useful, since it presents the most important information from both the LCM and the Grid in an accessible way.

It would be interesting to study if it is possible to present what products each product is depending on, in addition to the connected products that the LCM has information about, to increase the understanding of the system and as a tool for troubleshooting.
8. Conclusion

This chapter presents the conclusions of this study, which answer the research questions.

Question 1 – Do the users prefer a list-based or an object-based dashboard?
Seven out of ten respondents prefer the list prototype. This is partly because the respondents are more used to view the products in a list and unused to use tablets applications. If more respondents get a chance to get familiar with the object prototype they will probably like it. However, the object prototype does not present all the relevant information at the same time that several of the primary users are in need of, but instead presents the system in a cleaner way, which also is useful in other situations.

Question 2 – How does the possibility to get acquainted with the prototype before the evaluation affects the result?
The result of the study implies that the possibility clearly affects the result. None of the respondents that did not get familiar with the object prototype liked it while four out of five of those who got familiar with it liked it, and three even preferred it.

Five out of five that got familiar with the list prototype liked it, and two out of five that did not get familiar with it liked it too. There was not the same need to get familiar with the list prototype since it resembles the LCM’s interface, which most of the respondents are used to.
Appendix A. The evaluation methods

A.1. Questions for the background information about the respondent
   1. Department
   2. Role
   3. Years working with M3/Movex
   4. Age
   5. Usage of LCM (the static information)
      a. Do you feel comfortable using the LCM?
      b. Do you find the content in the LCM useful?
      c. Do you find the LCM user interface easy to use?
   6. Usage of Grid (runtime information)
      a. Do you feel comfortable using the Grid?
      b. Do you find the content in the Grid useful?
      c. Do you find the Grid user interface easy to use?

A.2. The statements in the System usability scale
   Grade the following ten statements from 1-5, where 5 means that you “Strongly agree” and 1 means that you “Strongly disagree”

   1. I think that I would like to use this system frequently
   2. I found the system unnecessarily complex
   3. I thought the system was easy to use
   4. I think that I would need the support of a technical person to be able to use this system
   5. I found the various functions in this system were well integrated
   6. I thought there was too much inconsistency in this system
   7. I would imagine that most people would learn to use this system very quickly
   8. I found the system very cumbersome to use
   9. I felt very confident using the system
   10. I needed to learn a lot of things before I could get going with this system

A.3. The interview questions about the prototype’s usability
   1. What do you think about the prototype?
   2. Do you find the information useful?
      a. Was all the information relevant?
      b. Was the level of the information suitable?
      c. Did you miss any information? General or application specific.
   3. Do you find the functions useful?
      a. Were all the functions relevant?
      b. Did you miss any function?
   4. Is there any information or function that should be hidden for a viewer compared to an admin?
   5. What do you think about the different screens/views?
      a. Spaces & other?
      b. Compare spaces?
      c. Hosts?
   6. What do you think about the presentation of the information?
Appendix A. The evaluation methods

7. What do you think about the interaction with the prototype?
   a. the navigation between the screens?
   b. how to access the additional or more specific information?
   c. using click-and-drag to change the order and layout?

8. Do you have anything more to add?
9. (Which prototype do you prefer?)

A.4. The information sent to the respondents

The first email below was sent to the respondents that had been interviewed during the initial interviews and therefore had a basic understanding of the purpose of the dashboard. The alternative first email was sent to those respondents that had not been interviewed. The second email was sent to all respondents.

First email:
Hi NN,
I write to you regarding my master thesis, where I am supposed to design a prototype for an M3 Information Dashboard.

From the interviews this spring I and [the component owner of the LCM] selected some information suitable to present in a first version of the dashboard. Then I designed two prototypes that present the information in two different design solutions. I now need help with evaluating these, both what information they present as well as how they present it.

The evaluation will be split into two parts, the first one will take about 30 minutes and the second will take place a couple of days later for about 60 minutes.

Would it be possible for you to participate in this evaluation?

Alternative first email:
Hi NN,
my name is Lovisa Gannholm. I am newly recruited to [the director of M3 Technology] team M3 Technology in Linköping, Sweden. I am doing my master thesis and have been asked by [the component owner of the LCM] to develop a prototype for a new product, an information dashboard. In general the purpose of a dashboard is to display information in an accessible way.

The idea of this dashboard is to show basic system information about M3Technology’s products for both internal and external customers. The information can come from e.g. the LCM client but also from the rest of the products.

The information presented is thus already existing but now in one place to be easily accessed by both internal and external customers. The information will be at basic level, more specific information is available elsewhere. For the information to be accessible the dashboard cannot present all existing information. That is why I need help from a few internal and external customers to decide which information that is most important.
I did some interviews this spring/summer with internal and external customers. From the interviews I and [the component owner of the LCM] selected some information suitable to present in a first version of the dashboard. Then I designed two prototypes that present the information in two different design solutions.

I now need help with evaluating these prototypes, both what information they present as well as how they present it. The evaluation will be split into two parts, the first one will take about 30 minutes and the second will take place a couple of days later for about 60 minutes.

Would it be possible for you to participate in this evaluation?

Second email:
Below I provide some information about the prototypes and the evaluation.

The prototypes present the same information but in two different ways. One is list/table based and the other is object based. The prototypes are created in a prototyping tool called Indigo Studio. These prototypes are not implemented, but are interactive - you can switch screen by clicking in different areas. The prototypes present examples of the most functions and the information that the design solutions are meant to have. These functions are possible to try out by clicking on the different buttons and icons in the prototypes.

The evaluation will be done by telephone and webex. The call will be recorded so that I can listen to it later, but will be deleted when the thesis is finished. Some important parts might however be transcribed and used in the report. The evaluation will be anonymous to outsiders since no names will be presented in the report. However the age, years working with M3 and the name of the department (for internal customers) will probably be included in the report.

The evaluation will start with a walk-through of the first prototype, where I explain and show the functions, by sharing my desktop. Then the participant will be able to try the prototype in his/hers browser, clicking around and viewing the different information. This while he/she thinks aloud, explaining what he/she experience and sharing the desktop. Between the first and the second occasion the participant is asked to try the prototype a couple of times and think about the information it presents, if it is relevant and if any information is missing.

At the second occasion the participant will be asked to consider ten statements about the usability of the dashboard, graded from "strongly agree" to "strongly disagree". Then the participant will be asked a couple of questions about the prototype, what was good and what was bad about it. After that the participant will look at the second prototype. The participant gets a walk-through and will then be able to try it for a couple of minutes, while thinking aloud. Finally the participant is asked to consider the same statements and questions as for the first prototype.

If you have any questions don't hesitate to ask!
Appendix B. The prototypes

B.1. Images of the list prototype

Figure 9.1: List prototype – Start view, the space Production is selected, the right half presents space specific information.
Figure 9.2: “Highlight” in the gray toolbox has been clicked, the interactive parts are highlighted.
Figure 9.3: The status (green arrow) of Tool2 has been clicked, the menu for the Grid’s status is displayed.
The Grid has been selected, the information to the right has changed to application specific information.

<table>
<thead>
<tr>
<th>Grid</th>
<th>Version</th>
<th>R/S</th>
<th>AU</th>
<th>E</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companion</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventAnalytics</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventHub</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GridDataBaseConnectivity</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InforSmartOffice</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONEnterpriseSearch</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONEnterpriseSearch3P</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAPSessionProvider</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaDataPublisher</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3BEEnvironment</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3BODProcessor</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3EnterpriseCollaborator</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3UAAdapter</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3BE_15.1</td>
<td>also in v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LicenseServer</td>
<td>v * * * *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StreamServe</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Additional Information**

Grid applications:

- Production: **!** 1 1

**Grid**

- **Application Information**
  - Installation name: PRD.Grid
  - Product name: Infor ION Grid 11.1.10
  - Version: 1.11.24
  - Links:
    - Management page
    - Logs
    - Properties

**Hosts**

- sestwrm3dev01 infor... win 2012.6.2
- sestwrm3dev02 infor... win 2012.6.2
- sestwrm3dev03 infor... win 2012.6.2
- sestwrm3dev04 infor... win 2012.6.2
- sestwrm3dev05 infor... win 2012.6.2

**Available Updates**

- Grid_11.124.1
- Grid_11.124.1

**Description of UI**

Lorem ipsum dolor sit amet.
Appendix B

The prototypes

Figure 9.5: M3BE has been selected, the information to the right has changed to application specific information.

### Production

<table>
<thead>
<tr>
<th>Applications</th>
<th>Version</th>
<th>R/S</th>
<th>AU</th>
<th>E</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>v 1.11.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Companion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventAnalytics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventHub</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GridDataBaseConnectivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InforSmartOffice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONEEnterpriseSearch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IONEEnterpriseSearch3P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAPSessionProvider</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaDataPublisher</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>M3BEEEnvironment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3BODProcessor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3EnterpriseCollaborator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3UIAdapter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3BE_15.1</td>
<td>also in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LicenseServer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StreamServe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### M3BEEEnvironment

#### Application Information
- **Installation name:** M3BE_15.1_PRD
- **Product name:** M3 BE 15.1 Environment
- **Version:** 10.1.2.0
- **Links:** Management page, Logs, Properties

#### Additional Information
- **Useful ports/URLs:** Port for MAK to BE URL to MUA
- **Foundation version:** 10.1.0.2.0
- **M3BE version:** 15.1
- **Feature pack vers.:** M3BE 15.1.1

#### Hosts
- sestwm3dev01.infor... win 2012.6.2
- sestwm3dev02.infor... win 2012.6.2

#### Connected Products
- LicenseServer
- MetaDataPublisher
- M3UIAdapter
- StreamServe

#### Databases
- **MSSQL**
  - MSSQLSERVER: MSSQL 2012 M3FDBPRD
  - 15.1

#### Available Updates
- Description of update
Figure 9.6: The settings icon has been clicked, the Settings menu is displayed.
Appendix B. The prototypes

Figure 9.7: The order of the applications has been changed to Alert.
Figure 9.8: The menu has been clicked, with the highlight on.
Figure 9.9: The second view, Other.
Figure 9.10: The third view, Compare spaces. Production and Test has been selected to compare.
Figure 9.11: The fourth view, Hosts. The first host has been selected, the information and the applications are displayed.
B.2. Images of the object prototype

Figure 9.12: Object prototype – The start view.
Figure 9.13: The Grid’s container has been expanded and presents all applications handled by the Grid.
Appendix B. The prototypes

Figure 9.14: “Highlight” in the gray toolbox has been clicked, the interactive parts are highlighted.
Figure 9.15: Production’s +-icon has been clicked, the additional info has been displayed, then the host-icon has been clicked, presenting the list of hosts, and last the info-icon has been clicked, presenting information about the one host.
Figure 9.16: The status (the green arrow) of the Grid has been clicked, the status menu is displayed.
Figure 9.17: The Grid’s exclamation mark has been clicked, the list of new updates is displayed, and the info-icon has been clicked, presenting information about the first update in the list.
Figure 9.18: M3BE has been clicked, it is highlighted and the connected applications has been highlighted with a thinner border.
Figure 9.19: Both the +-sign for BLS and for Tool2 has been clicked, both their additional info are displayed.
Figure 9.20: The settings-icon has been clicked, the settings menu is displayed.
Figure 9.21: The order of the objects has been changed to Groups.
Figure 9.22: The order of the objects has been changed to Alerts.
Figure 9.23: The menu has been clicked, with the highlight on.
Appendix B. The prototypes.
### Appendix B: The prototypes

#### Figure 9.25: The third view Compare spaces. Production and Test has been selected.

<table>
<thead>
<tr>
<th>Production</th>
<th>Test</th>
<th>Education</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRD 1</td>
<td>TST</td>
<td>TST</td>
<td>DEV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grid</th>
<th>BodP</th>
<th>IESM3</th>
<th>MEC</th>
<th>M3BE</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.11.24</td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>v.11.24</td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp</th>
<th>IES3P</th>
<th>MUA</th>
<th>LicServ</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
</tr>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GDBC</th>
<th>ISO</th>
<th>MSSQL</th>
<th>StrSer</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
</tr>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EventA</th>
<th>M3BE</th>
<th>EventH</th>
<th>MDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
</tr>
<tr>
<td>v.*<strong>.</strong></td>
<td>B</td>
<td>v.*<strong>.</strong></td>
<td>B</td>
</tr>
</tbody>
</table>
Figure 9.26: The fourth view Hosts. The first host has been expanded with the arrow, presenting the applications handled by the host.
References


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – 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/.

© Lovisa Gannholm