Abstract

Usability is a hip word today in our society of Information Technology, but what does it really mean? In this report we want to emphasise key words like usability, design, users and their participation in the design process. This thesis involves a human centred way to think in software developing.

We have evaluated the usability in a system. The result from our evaluation and the methods we used are described in this thesis. Our contribution is motivated design suggestions.

We hope you will enjoy your reading!
Introduction

People, Computer and Work, MDA\textsuperscript{1}, is an educational program at Blekinge Institute of Technology. The program integrates the subjects Computer Science and Work Science throughout the education. The aim of the MDA program is to give the students certain skills in perceiving how people interact with Information Technology, and with one another, in order to design IT-solutions to fit in with and support daily work practice.

We applied some of the methods we have learned during our education and the result is this Bachelor thesis of 20 academic points.

We will tell you about the different systems involved in our project and our work process during this time. We describe our fieldwork, the methods and the result of them. We also emphasise questions like; What is usability? Why involve users in a design process? Further on we will mention a commonly acknowledged user classification. We will discuss the subject What is a good system? in the connection with our design suggestions. Finally we will tell you about our problems during and a summary of this report.

In this report the company, ABB Body-in-White AB is mentioned as BiW. The new system BiW will implement is TopTier Enterprise Unification Platform and we will use the acronym EUP. The employees at BiW call EUP TopTier. When we are mentioning the systems we mean both Scala and EUP at the company. Scala is an economy resource-planning program. Otherwise we call them by its name. The new system is a general system adapted to the requirements at BiW. We make differences between the general EUP and EUP. With the general EUP we mean the general system developed by TopTier to different companies. EUP is with adaptations to BiW, which means it’s suited to BiW. These two levels are important to have in mind while reading our report to understand the possibilities in each level.

History

Our focus in this project is to evaluate the usability of a system and make suggestions to adapt the Graphical User Interface, GUI\textsuperscript{2}. The system is in fact new to the company but the content in the database is going to be reused. Therefore we wanted to understand the old system and study it to get a general view both of the old and the new system.

The general EUP has a predetermined design done by TopTier, the company who developed the system. We thought our mission to evaluate the system included all GUI design for example the placing of buttons and colours. But soon we were informed it was only the design of the information that was changeable. When the ad-

\textsuperscript{1} The Swedish acronym of our educational program is MDA, which means Människor Datateknik Arbetsliv in Swedish

\textsuperscript{2} See Glossary p 53
ministrator at BiW adapted EUP to the company he structured the design of the information. The adaptation of the EUP also contains user profiles for the employees.

At the time we didn’t know the GUI design was predetermined as much as it was, our focus was to personalise the new system, to suit the personal profiles to specific classified user groups. This leads to studies about user classification because we wanted to find user groups. Our suggestion of dividing the employees at BiW was dependent on which information structure, the users require depending on their work assignments. EUP contains user profiles to make the use more effective. When we were exploring the opportunities of the profiles for employees we looked at different user groups. The division we had in mind was based on if the employees used the Scala system for in- or output data and the use of the information in the system Scala.

The result of a Future Workshop made us realise it was not a good idea to base the information on departments. We had interesting discussions about this with the participators and we got important aspects for our further work. This breakdown situation made us think on new lines entirely. We dropped our user classification and the personal profiles and focused on making a good adaptation to BiW with motivated design suggestions.
The Aim with our Project

At our educational program we have learned the importance of involving the future users in the system development. During our education we have practised different methods and techniques to involve the users in the development. Applying these methods in this project was important for us.

Our aim with this Bachelor thesis was to study the usability part in a new system at BiW from an MDA perspective. With an MDA perspective we mean the future system shall be suited to the future user which is important for it to support the user.

We will look at the employees’ work practice to understand their needs in the new system, because we want EUP to support as many as possibly in their work. We used different Participatory Design and ethnographic methods. To get a system, with usability, we need to understand the employees’ needs in their work. One way of accomplishing this is to perform field studies and apply Participatory Design methods. BiW bought a complete system and the design questions are related to the information structure and the adaptations of the GUI.

“Our mission is to evaluate the usability of the system and make a suggestion for the interface adapta-


Our Personal Goal with the Project

Our goal in the project is on one hand to develop the practical experience of how the MDA work can be practised in real life. We also wanted to work against a “sharp” project at a company and reflect upon the theories we have learned through our education.

The Result

ABB Body-in-White AB is going to get this report, which contains description of our project, our mission at BiW, the results and references to methods to analyse the system with users participating. We are giving BiW a compilation of the different methods, a list of references and discussions about what’s important to think of in involving the users in the developing of software. As a part of our project we are going to present the Bachelor thesis at BiW.

---

1 See Glossary p 53
2 See Glossary p 53

5 BiW’s specification, Translated from Swedish
Background

With this part of the thesis we describe the company ABB Body-in-White and the work process at BiW. We will also tell you about the ongoing project, The TopTier Report and Analyse System Project.

Finally we will tell you why BiW bought a new system, the general EUP, and why they needed to change the structure of the Scala system. You can read more about the systems in chapter The Systems.

ABB Body-in-White AB

In 1997 ABB bought a part of Volvo and they started a joint-venture company, ABB Body-in-White AB. A joint-venture company is owned by two concerns in this case ABB and Volvo. BiW, which means unpainted coach, is an industrial concern.

The company develops and serves complete press-automation systems and control systems in the vehicle industry. The process at the company includes pre-study, simulation, process and construction, installation, put into operation at the company. Other assignments are to supply education, service and financing. BiW’s way of working is in projects. The business vision is to create the future production systems in the automotive industry.

The company buys the manufacturing abroad and the primary concerns are assembling and installation. In Ölölström they are 240 employees but in the concern of BiW they are 287 employees. BiW has five product domains; Assembly, Gun Weld Technique, Press Automation, Control Systems and Customer Service.

The Assembly product domain at BiW offers complete assembly lines as turnkey installations. The assembly systems are based on standardised and/or requirement-tailored solutions, and they are designed to meet all production volumes for cars and trucks. BiW also offers separate, individual equipment units for installation in existing assembly lines, for example hemming units. Full service competence together with extensive, genuine process knowledge and a project management organisation that will be working to find the best and most efficient solutions.

In the Gun Weld Technique product domain there is Lightweight Welding Guns (LWG) which makes the work easier in both manual welding and robotic welding. LWG reduce cycle times and increase precision. BiW also offers accessories for the Lightweight Welding Guns. For example electrodes Tip Dressers which are available in electrically driven or pneumatically driven models, for manual use or together with function packages. The gun welds are pneumatic or servodriven. They can be manual or robot operated and with or without integrated transformer.

The Press Automation product domain offers greater capacity and lower unit price and even better quality and more flexible and reliable operation. BiW offers a complete, turnkey installation based on the extremely flexible DOPPIN System solutions. The DOPPIN System in-
cludes equipment for conventional press lines from the smallest to the largest presses as well as equipment for transfer presses.

The Control Systems product domain offers tooling for line-automation equipment and robots. Tooling is a vital part of line-automation equipment. Precise orientation, secure retention and low weight are criteria essential to efficient, trouble-free operations. ABB has a wide range of tooling specifically adapted to robots.

The Customer Service product domain optimises the production processes of each customer. The department for Process Service offers continuous process improvements, complete training programs, feedback of operating and process characteristics, and spare parts management.

As one of our supervisor said: “We make all the tools that’s makes all the parts you can see on a Volvo!” 6

The Work process

The company is divided into different assignment areas; Leading, Control and Business development, Personal, Economy and IT, Market, Project Leadership, Preparation, Construction and Development, Supply of Material, Assembling and Installation and Service.

Leading, Control and Business development is one of the assignment areas. The purpose is to satisfy the customers’ expectations and to work continuously to improve the company’s processes and products. It’s also includes questions about strategies.

The Personal assignment area is responsible for guarantee the competence at the company through development in competence and recruiting.

The Economy and IT assignment area is working with economic project control, internal and external account and bookkeeping. The area is also responsible of the company’s work with IT questions.

The Market assignment area works with the entire contact process from an inquiry to an accepted contract and possibly customer complaints.

The Project Leadership assignment area guarantee resource planning, start projects, follow through the planned activity of project and finish the projects.

The Preparation assignment area purpose is to technical and economic make the customers wishes clear through order preparation, make an offer calculation, process preparation and layouts. Preparation also works to write technical descriptions, simulation and programming of robots.

The Construction and Development assignment area shall through development, construction, product keeping and customer suiting create processes, method, technical and products, which fills the customers requires and expectations. The area also creates routines of quality insurance of development, mechanical construction, construction of control technique and standardisation.

---

6 Notes from Supervisor meeting 010226, Translated from Swedish
The Supply of Material assignment area is responsible for the choice of suppliers, offer inquiry, agreements, orders, goods and store keeping and work with complaints.

The Assembling and Installation assignment area is responsible for manufacturing, assembling and test the products. Included installations, try out the product and start the product at the customers’ place.

The Service assignment area is to improve the customers’ production processes and make the customers of the company satisfied and works with guarantees.

All these employees in the different assignment are going to use EUP.

**The TopTier Report and Analyse System**

In January 2001 a project started at BiW, *The TopTier Report and Analyse System Project*. The purpose is an integrated system to be used for output of data and maybe also in the future, in an Intranet. With output data we mean values you will look at on the screen or print out as reports. EUP is going to be used by all employees at BiW, read examples of the future users work assignments in chapter *The Future Users in EUP*.

The process of selecting this system began when the project team picked out a test group. The test group consisted of eight persons from different sections at the company. The selection of the test group depended on the employees’ organisational position since the project team wanted employees with different purposes. The test group together evaluated three different systems and finally chooses the system they considered suitable. The majority of the participators in the test group considered *The TopTier Enterprise Unification Platform, The general EUP*, most suitable and it’s the reason why BiW bought the system.

The general EUP will be adapted to BiW depended on which information BiW chose to view in EUP. The manager of the project is going to classify the employees based on if they only need to print reports or also to analyse values in EUP.

**Why Change the Systems?**

The main reason to buy a new system, according to BiW, was to fill the need of compiling reports and analysing values. The employees were not able to get the reports they wanted with the old system. This forced them to make own data processing outside Scala. For example some of the employees have Excel files, in which they fill in the data the different systems at BiW today don’t support. The problem is the system Scala can not ensembles the values in the database Scala uses in form of reports, which was the reason of making for example Excel files.

“To get a general view in the projects we keep files in Excel and Access to control incorrect values and make selections”. (Log Future Workshop, Translated from Swedish)

---

7 Field note #69 and #74, Translated from Swedish
Keeping the information outside Scala limits the spread of information at the company. For example Excel files are not stored in the database Scala uses and keeping a data file in the hard disk of the computer hinders other employees to take part of the information. This lack of information flow is a big problem at BiW and even though the company has an open information policy, EUP is going to increase the information flow, since the general EUP have the ability to for example read Excel files. EUP is able to enoble figures in the database Scala uses to a greater extent. BiW hopes this improve the insurance of quality.

An important requirement in the new system is the search function because today there are limited possibilities to search among the information in Scala. Another aspect is the licence costs, which will be reduced with EUP. The Scala users who only use the system for output will only use EUP to perform their work. We want to emphasise; it’s the structure of the information, which will be changed. BiW wants to separate in and output with the different system. Many of the Scala users will use EUP as complement in their future work.

Many “human errors” in the input in the Scala makes the system untrustworthy. The failure depends on the employees who don’t know about the “human errors” think Scala is the problem. If one employee put in a wrong value it causes a lot of damage in the rest of the system. One of the participators in our Future Workshop expressed the problem like this:

"Human errors for example enter wrong values in the system is a combination of deficiency in Scala and lack of knowledge among the employees." (Log Future Workshop, Translated from Swedish)

This problem will remain because the entering of new values will still be done in Scala. The fact that people make mistakes, “human errors”, will always continue. EUP allows a data overview, which makes it easier to find errors in the input by grouping the data in Scala in new ways in EUP. “Just finding the errors is worth a lot.” It’s an economic concern for BiW to find the errors before they influence far too much of the organisation.

A benefit with EUP according to a user is the possibilities of finding failures: “With EUP its possible to take notice of the errors before we could only imagine the errors.”

---

8 Field note 010426, Translated from Swedish
9 Field note 010426, Translated from Swedish
What is usability?

Usability has become a desirable goal in software development. As the phrase goes a program should have high usability and be user-friendly.

Usability addresses the relationship between a tool and its user. In order for a tool to be effective, it must allow the intended users to accomplish their tasks in the best way possible. The same principle applies to computers, web-sites, and other software. In order for these systems to work, their users must be able to employ them effectively.¹⁰

Usability builds on the older idea of user friendliness, which was a term used to describe whether or not a system was easy for the user to work with. ISO¹¹ defines usability as: “...efficiency and satisfaction with which specified users can achieve specified goals in particular environments...”¹² The efficiency refers to how well the user carry out his or her task while using the system. Satisfaction means how the user experiences the use of the system.¹³

What makes software usable depends on a number of factors. For example how well the functionality fits the user's needs, how well the flow through the software fits the user's tasks, and how well the response of the software fits the user's expectations.¹⁴ We can learn to become better GUI designers by follow design guidelines. You can read about our design suggestions based on guidelines and our field study in a later chapter in this thesis. Not even the most well informed designer could only create a highly usable system without a process involving the actual users.

As Nielsen points out usability is the quality of a system easy to learn, easy to use, easy to remember, error tolerant, and subjectively pleasing.¹⁵ With Nielsen’s principles he wants to show the meaning of usability and we think this is a good picture of what usability is to us. Nielsen’s usability shows in picture 1.

¹⁰ Diamond Bullet Design, 2001, Internet
¹¹ See Glossary p 53
¹² ISO DIS 92-41-11, Faulkner, 2000, p 114
¹³ ISO 9241 part II
¹⁴ Diamond Bullet Design, 2001, Internet
¹⁵ Nielsen, 1993
The system should be easy to learn and support the user to perform his work with the system. The system should be efficient for the user, when the user has learned the system, to generate a high level of productivity. It should be easy to remember the system, the casual user should be able to return to the system after a period of not having used it, without have to learn everything all over again. The error rate in the system should be low. The users should make few errors during the use of the system, and if they do make errors they should easily recover from them. Further, catastrophic errors should not occur. The system should be pleasant to use, when users are satisfied they will like it. When a system support all these principles the system has high usability.

A system with usability should be easy to use and learn. With easy to use we don’t mean the designers shall blunt the intellect of the users but instead take advantage of the human natural settings. Through flexibility the system shall support novice users and expert users and all the users between these two categories. You can read more about the different users in chapter User Classification.

We think a system should not generate a lot of questions to the user while he is working. Consider this: if you are writing a document in a program and suddenly a dialogue window appears with a question you have to first answer before you can continue the writing. If you are a fast typewriter you do not look at the screen while writing and you will not see the dialogue window. Your writing will continue but on the screen the marker is stuck in the dialog window. The program is not supporting your writing process. As Nielsen said about “user-friendly”: “… user don’t need machines to be friendly to them, they just need machines that will not stand in their way when they try to get their work done.”16 We think this quotation speaks for itself and tells what we think about user friendliness.

Sarah Kuhn claims a well-designed technology take advantage of the human strengths such as skill, judgement and capacity for learning to create a robust and flexible system.17 Why shouldn’t we? We build the systems for the humans. We have to take advantage of what

---

16 Nielsen, 1993, p 25  
17 Kuhn, 1996, p 283
the computers are good at and what the people are good at when we are designing systems.
Why Users in the Design Process?

To engage the future users in the design process is important since their work practice is going to be supported by the new system. It was natural for us in our project to involve the users since our education advocates the human centre approach. Only the future users are capable to give us the necessary information, which can help us create a good system. “...an understanding of user’s work should be the starting point for defining enabling technology and direct future technology.”¹⁸ The future users are experts in their work and sometimes they can give us solutions about a problem that arise during the design process. At our Future Workshop we discussed design suggestions with the future users and we noticed they put the problems into other dimensions to us. A suggestion was to put the Scala modules in EUP, which included hiding modules they don’t use. The participants considered this not good because it was important for them to have all modules available. This was an important aspect to us, which we took in consideration. You can read more about this in chapter Our design Suggestions.

An employee who is affected by a decision or an action shall have a chance to influence it; participation in the design process is one main point in work place democracy.¹⁹ The employee’s work place and work routines will be affected and it should be natural to engage the employee in the design questions. At BiW a majority of the employees we met showed great interest of helping us get the system usable and at our Future Workshop the participators all contributed to the different design suggestions with enthusiasm.

The designer should develop a system making use of the employees’ knowledge of work. The quality of the system increases if the users participate from the beginning in the design process. The importance of suiting the technology to people is today commonly acknowledged.

Involving users in the design process will reduce the stress symptom that depends on the increased knowledge of the computerising. The result of this may for example lower the costs for sick-listed. The users don’t have to learn the system from scratch because they learn during the whole design process which is also time saving.

Systems, which do not fit the people, will always exist as long as companies and development teams ignore the importance of involving the user in the design process. Technology should be our friend in the creation of a better life, and complement human capacities and assist those who need extra support.²⁰ It’s obvious to design technology for the people and not design from the functions of the machine.

¹⁸ Holtzblatt, Jones, 1993, p 179
¹⁹ Bödker, 1991, 1993
²⁰ Norman, 1994
The Users as Participators

Participatory Design\textsuperscript{21} emphasises the importance of end-user participation in all phases of the design process.\textsuperscript{22} The end-user is the expert in his or her domain consequently it’s important to bring users into the software development process. The reason is it’s in the end-users’ domain the system will be in use. The end-users as participator lead to an initiated design process of the new system: at the same time as the future user influences the future system the software developer learns to understand the users’ work. Being engaged in the users work practice is one way to get knowledge about the work. For example like we did in this project, observing the user while working. In Participatory Design there are several methods, which can help to understand the users work practice. You can read more about these methods in the chapter Field Studies.

The designers have an important role in the user involvement. As Holzblatt and Jones point out its important designers let the users understand they are the experts in the work process.\textsuperscript{23} During our field study at BiW we met future users at different department to get a general picture. Even if a secretary claims she has nothing to bring us, she might give us a hint about the system.

It’s important to the users to know their suggestions are essential. If the designers see the importance of the future users as participators in the development, and if the users feel they can affect the design process, it will change the whole co-operation. It’s also important to show the users the individual benefits with the new product or system, which motivate them to participate in the design process even more.\textsuperscript{24}

The future users will feel more comfortable if the designers work with them in their work place. For example a clerk feels better if the designer observes him in his own office, because it’s his home ground. In the designers point of view it’s more natural to meet the users at the place where the system will be used. To meet the future users at their work place was obvious to us. We could not get the same material to analyse if we had met them on another place. Participatory design methods involve the users and there are many different methods a development team can use to involve the users and

\textsuperscript{21} See Glossary p 53
\textsuperscript{22} Kuhn, 1996
\textsuperscript{23} Holzblatt, Jones, 1993
\textsuperscript{24} Grudin, 1993
The Systems

One major application the employees at BiW use today is Scala, an economy resource-planning program. The new system is as mentioned called the general EUP. Both of these systems, Scala and EUP, are going to be used in parallel. Scala will be used for input of new values and EUP will take care of the output in form of for example reports. With input we mean new values, changed values or values being entered for storage and with output data we mean, as we said earlier, data you will look at on the screen or print out reports. In picture 3 you can look at the system connections:

Today Scala and EUP use the same database, but EUP also connected to other databases. In the future EUP will use more databases and the reason is BiW wants to get a comprehensive system.  

"The Portal will in the final installation (after phase 1) will contain several different..."
data sources. PDM system... Scala, SQL – Some local systems in Access will be implemented with a SQL 7 Server, OLAP Cubes, Lotus Notes Applications... Lotus Note Database..., Documentum..., WWW – Product catalogue and QA homepages, Application ...

Scala

Scala is an economy resource-planning program, ERP\textsuperscript{27}, which is catalogue structured. The different catalogues are called modules and at BiW they are named Bookkeeping/Accounting, Customer, Supplier, Order/Invoicing, Purchase Order, Material and Production Control, Stock Control, Statistic, Project Administration and System Maintenance. These modules are also divided into a number of sub modules.

The employees at BiW use Scala to enter new values in their work. Employees in the Storage department enter all incoming products into Scala, which makes it possible to for example the Industrial Logistics department to know what products need to be ordered. The Finance department uses Scala to put together bills from different projects. You can say Scala is a connecting place to all departments at BiW. The quotation, which is a user observation, describes one user, which works with Scala:

"The user is about to create a purchase order in Scala. She clicks with the mouse and opens the module "Purchase Order" and its sub module "Purchase Order Suggestion". A dialogue window shows at the screen, the framework in Scala is all the time showed in the upper Toolbar. She uses the mouse all the time except when she has to write something in the fields. In the dialogue window she has to select alternatives. This selection decides what appears at the Purchase Order. It's possible to get a list, which contains all the articles. – It's possible to get a list, which contains all the articles. – If I make a selection it speeds up the search.

To confirm all the selections in the dialogue window she presses enter at the keyboard. (A help dialogue window is constantly shown in Scala. It tells you what to do next.) When enter is pressed a new dialogue window shows "Create Purchase Order Suggestion". All the selections she made earlier are showed again and she reads all the alternatives to make sure she got the values she needs in the Purchase Order. She presses enter.

Dialogue window: Print on the screen? She selects YES. A report appears on the screen. (A third part product called JetForm shapes the report. JetForm uses control codes to put the information in Scala on the right place at a report-template.) " (Field note 010508, Translated from Swedish)

\textsuperscript{26} Petersen, H-H, Implementation of TopTier at ABB Body-in-White version 1.2, 010430, BiW's own material
\textsuperscript{27} See Glossary p 53
Picture 4 shows the left side of the window in Scala. The right side, see Picture 2, shows the sub modules in the module, which is marked. In the program Scala there are more modules besides these but the enterprise version at BiW consist of these modules.

**TopTier Enterprise Unification Platform**

The general EUP is the new system and the program gets all information from the same database as the system Scala. The users of EUP will look at views of information. The views are different relations between tables and fields in the database.

The general EUP is web-based and uses a web-browser, Microsoft Internet Explorer, which means it's based on Internet technology. In the future is it possible to integrate an Intranet.

The system works as a report and analysing system. An employee, who works to follow-up projects, could view values that belong to a specific project. The employee can get values for the project and compare them with similar projects they have done before. The compared values can be online or fixed data depended on how they will ennoble the figures in form of reports or OLAP cubes.

It’s possible to relate information in the general EUP with a powerful function, Drag and Relate. The user can view the result in the Workspace, see picture 5. Drag and Relate is really a user “drag and drop” action. This quotation illustrates a user printing a purchase order report in EUP, which is a similar assignment as in the previous quotation:

---

See Glossary p 53

Martina Holmquist, Ann-Marie Pettersson, MDA98
“A user does a Drag and Relate\textsuperscript{29} Marks a purchase order from Question in the Workspace in EUP and drags it to TTPuorderHead in the choice Scala Report in the Toolbar, because she wants a relation between these. She gets the wanted information in the Workspace. There are also buttons such as Excel, Print and Go to Find. She chooses print and the Workspace changes again and she selects a printer and pushes the button print.” (Field note 010506, Translated from Swedish)

The purpose with Drag and Relate is to relate two different objects from the database. The result of the Drag and Relate and the related objects’ different data shows in the Workspace. Picture 5 shows the main screen of a version of EUP:

In the general EUP the users are able to ennoble the data in OLAP cubes.\textsuperscript{30} The users can change the appearance of OLAP cubes themselves.\textsuperscript{31} In OLAP cubes you work against stored values, the values are dependent on already interpreted data. When you use OLAP cubes, you store a lot of data and get a quick overview of the values. You can choose to dump interval values for example every week if you like to. OLAP cubes can be used with online values too, but this is nothing BiW will use.

\textsuperscript{29} See Glossary p 53
\textsuperscript{30} See Glossary p 53
\textsuperscript{31} OLAP cubes, Appendix 2
With reports you work with online values, which means you always have the latest values at hand. There will be about fifteen different report-templates. The reports-templates gives the user possibilities to filter the content of the information by doing selections. For example if the user wants a report from the “module” TTMMainProject he has to make selections otherwise he gets all information about every main project in his report. The users are able to produce their own personal reports. This will increase the information flow because all the files outside Scala can be put in EUP. The quotation shows an employee at the Goods department:

“Before he delivers he has to make a consignment note. He opens the program Excel, we asked why and he told us all parameters were not available in Scala. Therefore he uses Excel based software from the specific package supplier.” (Field note #74, Translated from Swedish)

The general EUP got the possibility to read many different file-types for example Excel and Word and to ennable figures to a greater extend.

The top of the Toolbar shows a question mark, see 6.1 at picture 6, which is a link to Books Online. The function Books Online is an instruction book for the system where the user can get help.
The choices Time and ReportComponents in the Toolbar are not running in this moment. BiW has not decided if they are going to use them.

The other two parts in the GUI is the Information Bar, see picture 9, and the Workspace, picture 10.

The user can save this selection of values as a query in his personal folder in the choice Scala.

The choices Time and ReportComponents in the Toolbar are not running in this moment. BiW has not decided if they are going to use them.

The other two parts in the GUI is the Information Bar, see picture 9, and the Workspace, picture 10.

Picture 7 shows the choice Scala, in which the user has different sub choices. In the choice Scala the user views a query tool which allow the user to make SQL statements. A SQL statement is a Query, which contains relations between tables and fields in the tables.

The user can save this selection of values as a query in his personal folder in the choice Scala.

Picture 7 shows the choice Scala, in which the user has different sub choices. In the choice Scala the user views a query tool which allow the user to make SQL statements. A SQL statement is a Query, which contains relations between tables and fields in the tables.

The choices Time and ReportComponents in the Toolbar are not running in this moment. BiW has not decided if they are going to use them.

The other two parts in the GUI is the Information Bar, see picture 9, and the Workspace, picture 10.

Picture 7 shows the choice Scala, in which the user has different sub choices. In the choice Scala the user views a query tool which allow the user to make SQL statements. A SQL statement is a Query, which contains relations between tables and fields in the tables.

The user can save this selection of values as a query in his personal folder in the choice Scala.

The choices Time and ReportComponents in the Toolbar are not running in this moment. BiW has not decided if they are going to use them.

The other two parts in the GUI is the Information Bar, see picture 9, and the Workspace, picture 10.

The user can save this selection of values as a query in his personal folder in the choice Scala.
The News part contains information BiW wants to show the employees. It could be information about a new system that will be implemented at BiW.

The Information Bar, see picture 9, disappears and become a part of the Workspace while you are working. For example if the user search for an order number the result will be showed in the Workspace at the same time the Information Bar disappears. Only the Workspace and Toolbar are viewed.

Picture 10 shows the Workspace, which the user views the result of his work for example search results.

In the Workspace the user has Alerts and Questions available. Alerts are warnings about running data compared with a specified value. The user can use an Alert to be reminded when a project exceeds maximum costs. When the user is reminded by his Alert it shows in the Workspace, see picture 10. The administrator makes a number of Alerts and while making the Alerts he sets the necessary parameters for example start and end date. The user selects between the different Alerts, which the administrator made.

Questions are a summation of fixed values and they are not compared with any parameters. The Question shows for example a project costs. The administrator makes SQL statements and decides if they are a Question or an Alert. The user’s Query, as we mentioned earlier, could be made as a Question or an Alert. Then the user sends his Query’s SQL statement to the administrator, which have the authority to implement it to a Question or Alert into the GUI. Questions and Alerts are for a group of user and Queries are for one user only.

Beside the Search engine in the Information Bar there is a Find function in the general EUP, see picture 11. The Find function appears when the user marks a module in the choice Scala in the Toolbar. The Find function let the user search for specific data for example an order date.
The Administrator’s Possibilities in the general EUP

It’s possible to adapt the general EUP to BiW and the administrator has the possibility to make user profiles for the employees. The purpose is to make the employees everyday actions easier and make the system to fit the company. At BiW the administrator makes the adaptations of the user profiles, which mostly depends on the structure of the employees’ information. The adaptations are going to be based on what you need to do in the system and how much you will use it. It’s important that the users don’t have to make a user profile in the general EUP themselves. As Nielsen writes; “people are more motivated to start using things than to take the initial time to learn about them or to set up a lot of parameters”. The users have no time to explore the possibilities of changing the profile; instead they give priority to learn the new system. Composing a user profile should be a possibility for the users and not a demand. This is nothing we could explore because the TopTier Report and Analyse System Project are delayed and EUP is not used in full scale at this moment.

The general EUP has been adapted to the company and the information at the company. The project manager has selected the structure of the information and he selected the content in all parts of the GUI, at picture 5.

The reports in EUP will be designed according to the company’s needs and wishes from the employees. A special team will work with the design of reports needed in BiW to fill the need of new reports. The employees have to approve the reports and check if the values in the reports are correct.

The administrator selects the different data sources to EUP, both sources to look at values and to search among. They also decide which databases EUP will use for example Scala. The future EUP is going to use almost all databases at BiW. The administrator decides which views the users will see and the views are relations between tables and fields in the database.

It’s the administrator who decides rights and authorities for the users and the user groups in EUP. The authority includes for example what the users have available in EUP and what data they are able to search among.

The administrator determines the content and design of the Toolbar and the catalogues in the choices such as

Nielsen, 1998, Internet

Overview of the future EUP, Appendix 1
Scala, Scala Reports and MyPortal. The administrator also decides what the users can do with their profiles. The administrator makes a basic authority for the users; from this the user has possibilities to make a profile.

Other possibilities the administrator have are for example to create shortcuts to other programs such as Word. The shortcuts will be seen as an icon in the Toolbar. The administrator also decides which of the different Alerts and Questions the user can choose among.

The administrator also determines which web-sites the users can use to relate to Scala. For example the users can connect EUP to an external field in Dagens Industri, which makes it possible to compare with market values.

The administrator cannot change the program structure but he can programme some parts by himself.

The User’s Possibilities in EUP

With restrictions the users are able to make a user profile, based on the basic authority from the administrator. The way information is shown can be changed. The users use Personalise in the choice MyPortal to make the changes. The user can for example hide everything besides the Search engine. The users can change the front view at the Workspace part of the GUI, for example select how many columns he wants to show. The user can save his queries in a personal folder in the choice Scala in the Toolbar.

The users are able to choose among the Alerts and Questions which are based on the basic authority from the administrator, see picture 6. The quotation illustrates a user, which works with purchase orders who is going to adapt some parts of her profile:

"The user clicks with the mouse on the choice MyPortal and then Personalise in the Toolbar. The Information Bar changes, i.e. its colour, content and looks. Different choices appear. There are more choices now than in the usual Information Bar. She clicks at the choice Search and there she chooses among three different choices such as Web, documents and all. She chose all, clicks on Links and first nothing happen. She clicks again and this time at the name and a drop down menu appears and she ticks off some of the link she doesn’t want to view. She chose to adapt the Alerts too and she clicks at the choice Alert. A drop down menu shows and she choose to show Late purchase order and at last she clicks on Save." (Field note 010506, Translated from Swedish)

When the users do adaptations of their profiles the content in the Information Bar changes, as the quotation describes and as it shows in picture 12. A new choice for the user is Custom iViews, which could be for example a welcome picture in the Workspace.

---

34 Dagens Industri is the name on a Swedish newspaper, which in English is called The Industry of the Day.
Users can make specific Queries. For example if the user also wants to know which project the purchase order belongs to he perform another Drag and Relate relation and is able to look at further selections in detail. This Query can be saved in the users own folder in the choice Scala. The user can only save Queries in his personal folder. The administrator determines the fundamental functions, but if the employee wants to adapt the user profile even more he has to contacts the administrator.
**Field Study**

The ethnographic study gives us a collection of field material about the user and his work practice. The result of the study is dependent on careful analyses of our field data and experiences.

Participatory Design methods combined with an ethnographic study gains a comprehensive picture of the future users work practice. Different methods are described in detail below. Our fieldwork has involved four important main principles. These four main principles are according to Blomberg et al *Natural settings, Holism, Describing and Members Point of View.*

We studied the employees in their everyday settings, their natural settings. It’s a commitment to study the users’ activity in their everyday settings. We considered it important to evaluate EUP in the employees everyday context, to see if the system fit into the ensemble, which refers to another of Blomberg’s main principle: *holism.* The holism emphasis on natural settings derives in part from a belief that particular behaviours can only be understood in the everyday context in which they occur. The third main principle, *descriptive,* is based on the descriptive understanding of the users studied. Ethnographers describe how the users actually behave, not how they ought to behave. We developed a descriptive understanding of the users by observing the employees work practices and found out what the users said and what they actually did was not the same thing. A thing obvious to the users may not be obvious to the observer. The purpose with our ethnographic research methods was to get an insider’s view of the situation. The final principle is about understanding the world from the *Members point of view.* We have carried out our project guided by these four principles.

> “...ethnography is concerned with understanding other peoples behaviour in the context which it occurs and from the point-of-view of the people studied.” (Blomberg et al, 1993, p 128-129)

Through our ethnographic study we have got a view of the employees work situation. Some of the methods we have used are interviews and field notes.

We want to emphasise general guidelines to follow in the methods, but we consider it important to adapt them to the circumstances. When we have used the methods between the two of us and together with the users we have adapted them to the situation.

The involved employees wanted to be anonymous, therefore we decided to not write anything which could reveal their identity. It has been hard for the thesis ethnographic trustworthiness, but the only ethical possibility is to follow the restrictions from the employees and BiW.

---

35 Blomberg et al, 1993, p 125
36 Blomberg et al, 1993
You can follow a structure in the further chapters. The structure is to first describe the theory of the methods and then our applications of the methods in this project.

**Learn the Systems**

Learn the systems is not a method its a name we used for it. To learn the systems was a huge assignment for us in this project. We think it’s important to not forget this in an evaluation, because often the evaluator is an outsider and do not know the system. As Faulkner mention:

“In order to make software products usable we need to know what the task is that the user will be performing with them.” (Faulkner, 2000, p 57)

We needed the knowledge about the systems to understand how the employees worked. Through getting knowledge about the systems we could understand our mission at BiW. We learned to use EUP and Scala by “learning-by-doing” 37. We learned experientially by exploring the systems without any instructions. We tried, failed and tried again. At the same time as we learned about Scala and EUP, we also needed to see the possibilities in the systems. During our field study we learned a lot from the employees when they described their work.

---

37 Lave & Wenger, 1991, The authors mean learning by doing is to learn experientially, through trying out on your own hands.

**Informal Interview**

An interview is, according to Ely, a conversation between two persons with the purpose of gathering information. Ely divides interviews into two different kinds, which are formal and informal interviews. The formal interview is planned and structured while the informal is more ad hoc 38. According to Ely every interview has got its own structure; the difference is in how the structure is arranged. In some interviews the structure is predetermined and in other cases the structure of the interview is shaped in course of events.

We have used informal interviews, which means the structure of the interview is not arranged and the interview takes place at the employee’s work place. An informal interview to us is when we ask the employee to describe something and the dialogue has continued without prepared questions. The structure of informal interviews has been an introduction of ourselves and our work at BiW, then the employee showed and told us about his work and what parts he used in Scala and EUP.

We both took field notes during the interviews that made it possible for us to early document our experiences. The field notes have been useful to reflect upon in our process of writing. “... ‘field notes’, are quick notes or whispering in a tape recorder of details and dialogues which are guideline for a more specific description.” 40

---

38 See Glossary p 53
39 Ely,1993, Translated from Swedish
40 Ely, 1993, p 78, Translated from Swedish
At the start of our project it was only the super users who used EUP. Therefore it was important for us to come in contact with them. We wanted to get to know both the system and the users who are going to be supported by EUP. The first meeting we wanted to be short, to get a brief introduction of the employees and then come back later. The purpose was to form an opinion about the employee. Questions we thought about were: Is he interested in what we are doing and does he have the time for us?

With about fifteen employees we used informal interviews, but soon we realised this was time consuming and we couldn’t meet all the employees in this way. Therefore we made a questionnaire and sent it out to the employees. We are going to describe this method in chapter Questionnaire. With the questionnaires as basis we formed an opinion of which employee who was interested for us in our work and whom we could observe. After this we didn’t make any appointments with the users. Sometimes the employees were too busy to meet us, which was frustrating. But we didn’t want them to be prepared, because we thought it would be better to see the spontaneous work without any preparations.

**Observation**

Observation is to understand human activity in the everyday setting in which it occurs. It’s well-known that what people say and what they do are not the same. This is a main reason for observations of ongoing activity. As Blomberg et al said:

> “The distinction between what people say and what they do is related to the distinction between ideal and manifest behaviour. Ideal behaviour is what every ‘good’ member of the community should do, whereas manifest behaviour is what people actually do.” (Blomberg et al, 1993. P 130)

There are many ways to conduct observations. One way is to be the fly on the wall, meaning the fieldworker tries to be as unobtrusive as possible. The observed person do not notice the fieldworker which leads to he acts naturally.

Another way to observe is to be a participant observer, which means the fieldworker becomes a full participant in the activity. The observed person and the fieldworker work together in the activity studied.

During observations the fieldworker gathers material to be analysed. As Ely points out: "The most important way of gathering facts are through looking and listening.”

During our observations, one of us was active in a conversation and the other was observing. This was easier to accept for the employee than if we both observed without any conversation. At the same time we observed the user we also learned more about the systems and the employees work. It could be like this for the active person in the observation: (We = W and the employee = E)

---

41 Blomberg et al, 1993
42 Blomberg et al, 1993
43 Ely, 1993, p 49, Translated from Swedish
“First we did an introduction about our self and the purpose with the interview. Then:
W: What are you working with?
E: I use the program Smart team. I do manufacturing lists and use the module Material and Production Control and some of its subparts such as adaptation.
W: Ok
E: I may have to find out an article number then I use the sub module Ask Article.
W: Can you tell more about your work and show us?
E: I also do pre-calculations, which is done in the modules Material and Production Control. I do a registering of a structure, put in new articles and units codes.
W: How do you do that?
E: I put in new facts with the help of an old copy of this work. I also make a release to the production in Word. He switches program, which lays as an icon at the start-menu, with the mouse and shows us the specific file. It’s an order to the workshop and then I have to do one more order to the external customer.
W: Do you use the system to add data or to print it?
E: I don’t print a lot of reports.” (Field note #44, Translated from Swedish)

Fägerborg mentions it: “To ask, listen and watch—at the same time try it out by yourself — were the most important learning methods.”44 We have performed all these events during our project.

The quotation shows an extract from a observer log. You can compare the two different quotations and notice the details in the last one. An employee (E) who works with purchase orders showed and told us (W):

“E: I do orders for spare parts and put them into the system. I check the order if it’s delivered in time and look after the deliveries. (She marked the module Material and Production Control and go through some subparts such as Adaptations and Questions.)
E: I ask for article and check the delivery date, so ordered article will be here in time. (She turns around)
E: If an article is not in the system Scala will just take next number. That’s why I use Material and Production Control and Ask Question about the article there. (She shows a message that pops up and it said the article isn’t registered.)
W: Which other modules in Scala do you use?
E: I use Purchase Order and Registered and Registered set.
W: Disadvantages in Scala?
E: There is no follow-up tool. Scala is hard worked and I use the Excel files because it’s a long way to do the same thing in Scala” (Field note #69, Translated from Swedish)

We want to emphasise the difference between work practice and work process. To study work practice is to notice details in the work that the observed is not aware of and the work process includes only the flow of the work.

We noticed a problem while comparing the questionnaires to what the users actually said when we met them. For example in the questionnaire the employees wrote which modules in Scala they work with. When we observed them at the field it appeared they actually did not use all the modules they mentioned in the questionnaire. Since we used other methods as complements to the questionnaires we became aware of this problem. Run-

44 Fägerborg, 1997, p 19, Translated from Swedish
fors points out this as self-contradictions depending on if the person answered “naturally” or wrote what he thought he should answer.\textsuperscript{45}

As we mentioned earlier one of us wrote and observed while the other was active in the conversation. Looking while the employees worked were useful in the case we could notice things they didn’t tell us. We could observe the details the employees left out when they described their work practice. As the quotation show when an employee works with calculations:

“\textit{I register project calculations and one calculation has a lot of activity codes. (He goes trough his paper and search for something. He shows a paper with all activity codes and their different meanings.) He calculates each unit... Writes in a unit code (The costs for it shows automatic.) First step is to select a subproject and do a calculation. (He skims through a binder to find a project number, but he didn’t tell us about it).}” (Field note #51, Translated from Swedish)

While observing the employees sometimes showed us things not relevant to us in our work. This was an aspect hard to deal with. We experienced it difficult to explain the lack of interest to the employee. But soon we realised even if it was not relevant at the moment it may be important information to us in our future analysis.

We observed about thirty employees and many of the employees were helpful to us and described more than once when we didn’t understand. Some of the observations were very short and other lasted for an hour. One employee printed screen dumps at the same time he talked about what he did. The employee wanted us to have the dumps as a support to understand our notes we have taken during the observations. If we didn’t understand we always could come back to him.\textsuperscript{46} This support by the users made us feel safe in our work.

\section*{Questionnaire}

Questionnaires, formularies, consists of questions about a specific topic. In most cases the questionnaires have a statistic purpose. Questionnaires are for gathering and structuring a large collection of information. Faulkner describes the method:

\textit{“One of the ways in which the usability engineer can gather information about current work practices or opinions on the system is by the use of the questionnaire.”} (Faulkner, 2000, p 35)

Our purpose with the questionnaires was to come in contact with many different users. The assignment was more complex than we imagined.

By sending out 185 questionnaires to the employees we came in contact with different users and could get a picture of which systems they used. We asked about Scala, its modules, EUP and how much the employees used the systems. The purpose with this method was not to

\textsuperscript{45} Runfors, 1992, Translated from Swedish

\textsuperscript{46} Field note #51, Translated from Swedish
meet every employee who returned the questionnaire, but try to identify the employees who were interested in our study and to get information about them.

About the complex of problems of making a good questionnaire, Runfors points out the importance of not giving the person whom will answer the questionnaire no space for misunderstandings. The questions must be obvious. The person who fills in the questionnaire will not know anything about the background of the questionnaires.

Before we sent out any questionnaires we had to get an approval from the personal manager. We got the advice to delete the name part in the questionnaires to make the employees feel more protected and anonymous. Instead we numbered the different questionnaires and wrote the name on a list. We were the only who knew the employees who answered. It was easier for us to work with single numbers instead of confidential information about the employees. For example Hans Hansson (fictitious name) has number 4 at our list. Another piece of advice was to have an even scale instead of an odd scale in the answer alternatives. With an odd scale we have a tendency to choose a middle alternative. With an even scale you can’t choose a middle alternative; you have to reflect on if you belong in the beginning or in the end of the scale. For example one question was:

“Which of the programs do you use? (Number the programs with 1-4, 1 is very little and 4 is very much)"

We called EUP TopTier in the questionnaires because it was an established name at BiW.

The questionnaires were sent out in two sets. The first set consisted of 87 questionnaires and about 50 percent of the receivers returned the questionnaires. We started with a compilation of these answers and met some of these users on the field, but soon we realized it was not enough for our continued work. Many of the employees weren’t interested in our work or they did not have the time and so on. In the second set of questionnaires, we had a different view of what we wanted to know. We made the questionnaire more specific concerning what the employees used in the different modules in Scala. For example the employees should answer which sub modules in the Scala module Purchase Order they used. We also wanted to know if they printed values or put in new ones. An example in the questionnaires:

“Do you use the program to:
Print reports? Yes No
Registered new values? Yes No
Both? Yes No”

We thought it was interesting to know this because of the dividing of in- and output between Scala and EUP.

The employees were given five days to answer the questionnaire and send it back to our internal post box. First we considered sending the questionnaire by e-mail.
but decided to send it by internal post because it’s harder to forget a physical letter. E-mail is easy to “forget” to answer, especially with a large attached questionnaire. For example if an employee got the questionnaire by e-mail and doesn’t have time to fill it in, he probably will forget it later.

The questionnaire was a good method for us in our project. We easily formed an opinion of the different employees and we found the users who were interested in our project. The questionnaires were a basis in our work process. We got a picture of how different employees worked in different systems. Our purpose was to get an overview of the user and to find the users who would be the most interesting in our further work. We wanted to get together with users, both those who already used EUP and users with no experience at all of the system. The questionnaire made it easier for us in many ways; for example in the questionnaire we asked the employees, if they were interested in meeting us. From now on we knew which employee who wanted to meet us and who wouldn’t. The compilation of the answers was difficult to do. There was a large number of information to structure.

Questionnaires are not a typical method at the MDA program, but it was a good way for us to get started in this project with many users.

**Scenario**

A scenario, according to Löwgren and Stolterman, describes a course of events in a specific situation. The stories should be personal and detailed to make it possible to handle all kinds of imaginable situations and because it should be easy to recognise the situation. Bödker said: “Scenarios can be used to draw attention to the present practice of the users and to the aspects of this that will be changed due to the introduction of a new artefact.”

We differentiate between scenarios between the two of us and with the users, because we have used scenarios in both cases. Scenarios as a method were useful to us when we got to know the systems. We had a dialogue at the same time as we “worked” with the systems. It could be like this: Imagine you are a clerk in the Finance department and you would like to see the budget of a specific project. With a scenario as starting point we explored and learned about the systems.

When we met the users it was often more easily for us to understand the work situation if the employee played a scenario for us. One example is when we met an employee on the field and he simulated one part of his work to us while we were observing. This was instructive to us and we got an insight into his routines. Scenarios helped us to get an insight into how the different systems are used in practice, which gave us an understanding for the employees in their work practice.

The project manager made scenarios and told us the differences between how the systems worked today and

---

48 Löwgren, Stolterman, 1998, Translated from Swedish
49 Bödker, 1991, p 144
50 Field note #09
how it will work in the future EUP. The quotation shows a scenario of the future EUP:

“In the future in EUP when an employee gets a phone call from a customer who wants to get a drawing. The employee enters the purchase number in the EUP system, he drag and relates it and gets the drawing on the screen, enters the print button and the employee gets the drawing in the printer.” (Field note 010313, Translated from Swedish)

Scenarios have been a good method for us to draw parallels between something familiar and something new.

**Brainstorming**

Brainstorming is a method to help you generate new ideas and thoughts. According to Löwgren and Stolterman there are many ways to practice this method. Most important is to let all the participants have opportunities to freely speak out without being criticised. A guideline in this method is to be positive to each other’s ideas. A clear question or a problem serves as a starting-point to produce as many ideas as possible. When the brainstorming is over you have to estimate the suggestions, develop or delete.\(^5\)

We applied this method to ourselves, not with the users, but it’s a good method when a problem arises during the design process. We used this method because it was a way for us to find new perspectives of our work, for example when we got breakdown situations. A breakdown situation is according to Bödker:

“…breakdown situation, situations in which some unarticulated conflict occurs between the assumed conditions for the operations on the one hand, and the actual conditions on the other; between the human reflection of the material conditions, and the actual conditions.” (Bödker, 1991, p 27)

Our most serious breakdown situation arose when we realised we had to change our focus. Our questions were; What’s our focus now? How do we go further in our work? We started to give each other different suggestions and through critique we tried to find a solution to our problem.

During a brainstorming you shouldn’t criticise other member’s ideas but when we applied the method we criticised each other because we wanted to limit ourselves. Brainstorming can generate too many ideas or solutions to a problem and this in itself could be a problem.

The result of our most important brainstorm was to leave the idea about the user classification and the personal profiles, as we mentioned earlier, and instead look at the information structure and suggest an adaptation of the GUI through guidelines.

**Future Workshop**

The purpose of the method Future Workshop is according to Greenbaum to start from a common problematic

---

\(^5\) Löwgren, Stolterman, 1998, Translated from Swedish
situation generating visions about the future and discuss how these visions can be realised.\textsuperscript{52}

Future workshops are divided into three phases, the \textit{critique}, the \textit{fantasy} and the \textit{implementation phase}. The critique phase is basically according to Bödker et al a structured brainstorming focusing on current problems at work. The purpose is to list existing problems in the specific work situation and then encircles the main problems. The purpose of the fantasy phase is to let the participants solve problems without any critique. In the implementation phase the participants discuss how they ought to solve the problems.\textsuperscript{53}

We had a Future Workshop with employees at BiW and we tried out the phases in this method. We choose to not involve management levels because we thought it would restrain the participants from freely speaking out. Including ourselves we were six participators and all the users were from same department at BiW. We tape-recorded the workshop because we as designers would participate and be active in the Future Workshop. The participants allowed us to tape-record the workshop. We placed pencils, papers, tape-recorder and screen dumps as mock-ups on the table. Our purpose was to let the participants use the material during the Future Workshop.

At the beginning of the Future Workshop we told the participants about us, our project and also about the disposition of the Future Workshop. We started with the critique phase and discussed Scala and its advantages and disadvantages. The participants discussed problems for example errors in update of new values, “human errors” and the lack of reports.

With the general EUP in mind we applied the fantasy phase and implementation phase as one phase. The reason was most of the general EUP is already implemented except for example reports and other information structure. We both think the discussions in this Future Workshop were useful for us. We got a lot of material to analyse and many new aspects of our design suggestions to reflect upon. One important aspect to us was the confirmation of information. One example of this is the use of external files outside Scala. “\textit{I don’t write in Scala instead I use my own supplier-program in Excel.}”\textsuperscript{54}

The most important result of the Future Workshop was a conclusion about the Scala modules. Our opinion was to hide the modules in Scala the user don’t use. We discussed this in the Future Workshop and it appeared the users did not like this idea. If you are aware of how to do a Drag and Relate between two modules you have to access the modules necessary to get a result. More about this can be found in chapter \textit{Our Design Suggestions}.

\section*{Mock-up}

A mock-up is a simple prototype in paper or cardboard representing the future system. The purpose is to give the user a simulated picture of the intended system. Ehn and

\begin{small}
\textsuperscript{52} Greenbaum, 1992
\textsuperscript{53} Bödker, et al, 1993
\textsuperscript{54} Field note #69, Translated from Swedish
\end{small}
Kyng argue why mock-ups are good. They mention concepts like hands-on-experience, understandable, cheap and fun. Hands-on-experience implies you are able to simulate, show and touch the system without having the real system at hand. The mock-up is understandable and cheap because it’s made of simple materials like paper or cardboard. The last argument is “fun”, because it’s fun to work with mock-ups!

Mock-ups are a good way for the development team and the users to get a common picture of the future system. Mock-ups are good since the users are able to see the system or a product not finished “for real”. As we experienced in earlier projects a computerised prototype gives the wrong signals to the users. The future users may think the system is finished. Perhaps the user is afraid to suggest modifications when he sees a computerised prototype. A mock-up in paper is familiar to the user and it may feel easier for the future user to express design suggestions. Holzblatt and Jones point out the benefits with paper prototypes as easy to make and work with and they support the ongoing changes and challenges in the design process. Mock-ups allow the users to fantasise about how a product may look like.

Making mock-ups to realise our ideas in real life has been an instructive way to follow up our design suggestions. We have worked with mock-ups when we reflected upon different design solutions in EUP both with the

users and when the two of us were discussing design suggestions. When we looked at the employees different needs and wishes we sketched at the same time on a note pad. It was a way to show each other different ideas.

Using mock-ups in our Future Workshop was a useful method for us to get the users involved in the project because they could feel they had something to give us. We used screen dumps, at Scala and EUP, as mock-ups to engage the employees and to use as a basis for discussions. The screen dumps as mock-ups were a good basis for all the participants in the workshop and it was a good way to start thinking. It was easy to show the participants details at the dumps at the same time as we discussed.

**Process of Writing**

We have based the name of this method on our writing because writing is a huge assignment in a ethnographic work.

Our process of writing was to be in different places and write separately. After a day or two we switched documents, reflected upon each other’s work and continued the writing. We worked with the same version of the report, and renamed it careful in number order. This iterative process suited us well. It’s easier to get your own version of things and to put your own words and ideas on paper. It has been important for us to regularly get together and discuss the report. It has been a lot of extra work for us depending on this report is in English. We were not used to work with a foreign language

---

55 Ehn, Kyng, 1991
56 Holzblatt, Jones, 1993
in a report like this, which was a new instructive experience for us.

We wanted to write in terms, which would make it possible for someone unfamiliar with the circumstances to understand what we are writing about. We as writers knew about the company, the “typical workflow” and other details we take for granted. The fact this thesis is in, for us, a foreign language has made this even more difficult.

We chose to write during all the phases in the project, which was a good method for us because we had to reflect upon the situation and gradually analyse the material from the fieldwork. We experienced it sometimes as time consuming; it demands much writing and a lot of information to handle.

At the end of our project we found it important to work together since the time was running out. Working separately is a good but time consuming method.

**Comments about the Methods**

The methods we have described can be used in an ethnographic field study. To make good systems it’s important to really know what you should look for and learn more about in the user work practice.

To understand what happens at the work place is a basic concept in developing good and usable software. You must obtain knowledge about the users and guidelines to develop a system to fit the users in different situations.

Using Participatory Design and Ethnographic methods is a way to let the future users participate in the design process. We used the methods iterative through the entire project.

Another method is to use video taping when observing the user. We have experienced the method as good in earlier projects. Video taping gives information to look at closely. By watching the material over and over again makes it possible to discover details in the users work practice.

Jordan and Henderson means “…videotaping /.../produces data much closer to the event itself than other kinds of re-presentation.” We considered this method not right in our circumstances, because of the secrecy at the company and the ethic morals to the employees. Video taping is also a time consuming method. One aspect would have been to video tape a selected group of employees but we found our other methods good enough for us.

When evaluating materials from the fieldwork it’s important to think of the ethic morals. In a thesis the fieldworker has to balance the employees anonymity and the trustworthiness of a thesis.

---

57 Jordan, Henderson, 1994, p 13
User Classification

The IT management group will classify the users of EUP dependent on a level of authority as basis of what’s relevant for the employees in their work. The purpose is to make profiles, which contains relevant information in their work. The employees will be divided into super, advanced and basic users, which you can see in the categories below.

We want to emphasise the importance of the user classification. There are many experts in this area and therefore we have chosen to not get involved in the discussion but mention it as a reminder. The experts in usability have common guidelines in classification. For example Hoberg divides the users into five levels such as novice, new-beginner, competent, good and expert. We are going to mention the novice, expert and the users in the middle of these two, which we consider most important. You have to find these three user groups before you divide them into smaller groups. The different users are described shortly in the following chapters; Novice Users, Middle Users and Expert Users.

The Novice Users

Novice users are those who are afraid of using computers. They don’t know what to do and they are afraid to break something. They have little or no experience at all of computers.

We think novice users needs to easy find and understand good help system. Functionality should be limited and the system should always volunteer activities for novice users to choose from. It’s easier to a novice user to select alternatives from a given set of choices and it’s not necessary to show the novice users all parts of the system at once, because they can be interrupted in their work process.

Novice users need guiding through the process of getting to know the system. As Suchman said: “For the novice engaged in a procedural task, the guiding inquiry is some form of question ‘What next?’” The question is a request of next step in his work with the system.

You can find recently employed at BiW in this category since they have no or little experience of the company’s system. “I’m new here. I have just been here for a year and I haven’t worked in Scala.” A year is a long time to be a recently employed, but it’s not the time that decides if you are a novice user or not, it’s the knowledge about the system.

---

58 Hoberg, 1998
59 Faulkner, 2000
60 Suchman, 1987, p 132
61 Field note #53, Translated from Swedish
**The Middle Users**

We think the middle users are both novice and expert users, they need what both of this categories need, expectantly good manuals.

When Faulkner talks about this group of users she calls them intermittent users.\(^{62}\) With intermittent users she means users who use the system occasionally in periods of time. They do remember broad aspects of the system but not the details, it’s therefore important they are supported in their search for help.

The Middle User group is often the biggest of these three, but it can be divided into smaller groups. This group is often the most interested because they are a major group and the knowledge is varying.

You can find the basic users at BiW in this category, they who are in need of the reports in EUP, "I will use TopTier mostly to standard reports."\(^{63}\) Some of the employees already knew about BiW’s dividing with EUP.

**The Expert Users**

Expert users are users who are good at the programs they are using. They are users who touch all the buttons and really enjoy the use of computers.

It’s not possible to be expert in everything and therefore experts also need help system, often to consult with. For example the experts get more advanced assignments to do and have to learn more about the system. They become more and more advanced users.

Expert users need to be provided with accelerators through the system. In a GUI this might be shortcuts and defaults. According to Grand it’s important to provide shortcuts to routine tasks for experienced users.\(^{64}\) They who regularly use a specific GUI should be able to perform routine tasks as quickly as possible, through for example short commands at the keyboard.

You can find the advanced users at BiW in this category because they are in need of analysing values by for example using OLAP cubes.

**The Super Users**

The super users at BiW are seven employees who have used the general EUP from the start of the introducing of the TopTier Report and Analyse System Project. All except of two of the super users were involved in the test group when the general EUP were selected.

The name of the group, super users, is established in the line of business, but we have chosen to not take part of this in our discussion. We have divided the super users as a part of the experts because also expert can be divided into minor groups and should be in consideration.

The responsibility of the super users are according to BiW to help the users, in the super users domain of competence, with problems in EUP and to pass on questions and problems to the project team. They do also take part

---

\(^{62}\) Faulkner, 2000

\(^{63}\) Field note #41, Translated from Swedish

\(^{64}\) Grand, 1999
in the development of EUP, which means they should support the project team with information, education and maintenance.

**Conclusions about User Classification**

We have noticed three levels of expertise at BiW; novice, middle and expert no matter if they are basic, advanced or super users. Novice, middle and expert users are depended on their skills and knowledge of the systems. Everybody in the different levels should be represented in the systems.

A good system is flexible if it supports all these categories of skills and knowledge. It's very important to have these different users in mind during the design process. If a system only fit to the novice users then all the other users will find the system uninteresting. Other users can't learn more about the system or develop their skills. A system for experts only is a system, which is difficult to learn for the new users and they may give up before they have learned the system. The systems for the middle users may seem as a good system but is not obvious. The design process balance between all these users, by doing this makes the system flexible.

As we said before there are many experts in user classification. We want to recommend Xristine Faulkner and Christer Hoberg as further reading in the domain of user classification, see chapter Literature.
What is a Good System?

Is it possible to develop a system to suit everybody? It’s not conceivable, but it can fit as many as possibly. The GUI is the user’s picture of the system, it’s through the GUI the user interact with the system and therefore is it important to have a well-designed GUI. Bödker describes a user interface, our GUI, as followed:

“The user interface can be defined as the software and hardware supporting the human operation of the computer application in a specific type of use activity, constituting some of the material conditions for triggering specific operations in a specific use situation.” (Bödker, 1991, p 140)

Most important is to fit the GUI to its purpose and for the users. An IT artefact must always be judged in the situation it will be used in. For example Winograd claims the importance of observing an IT-artefact in its right context otherwise its meaning will be twisted. The user is influenced by his earlier experiences from the IT-artefact. Many developers and experts in the domain of usability agree IT-artefacts have to be developed in the specific situation they will be used in.

Our Design Suggestions

The general EUP has a predetermined design done by the company who developed the system, TopTier. The administrator is able to adapt the design of the information in the general EUP. In the EUP there are possibilities to adapt user profiles for the employees.

Our usability guidelines and design suggestions are based on the literature we studied and our field study at the company.

Language

According to Nielsen a good system should speak the users language to be familiar, he claims the language should not be in a foreign language. The foreign language in this case is English. We asked two users, user 1 and user 2, if they had made adaptations in their user profiles. Neither of them had done it before, but they started to look for the function:

“User 1 started to click in the Toolbar with the mouse. User 2: ‘I think it’s Personalise.’ User 2: ‘No I am sure it’s not Personalise.’ User 1 searches and clicks with the mouse to find the function. /.../ At last we tell them that the function for adaptation is named Personalise.”(Field note 010516, Translated from Swedish)

The quotation shows how hard it could be to understand the meaning of foreign language words. Another reason

---

65 See Glossary p 53
66 Winograd, 1996
67 Nielsen, Internet
to not use foreign words is the tendency to get long words when translating them. For example the long term TTAccountTransaction leads to shortening, in this case TTAccountTransac, see picture 14. An employee expressed: “Everything is shortenings!” This is a problem the designers have to consider in the decision taken in the information structure.

We reacted upon words in EUP, which are both Swedish and English. Because of EUP is in process of change, the Swedish-English words may not remain. One example is delproject in the choice Scala in the Toolbar in EUP, which means sub project. Del is the Swedish word for part and project is an English word. Another example the employees experienced hard to understand in English was the term business part, which means motpart in Swedish. Mot in Swedish means against and part is part in both languages. This language mixes could make the EUP inconsistent. We discussed this issue with our supervisor at BiW and the answer we got was “there is no word for it in English the employees understand.”

Another reason for the language mixes in EUP is that Scala and EUP uses the same database. The database is in Swedish and because of this it occurs conflicts when the values are viewed in an English GUI, EUP.

We examined the language mixes with the employees. The majority found it easier to use the Swedish terms and language mixed words instead of name the terms in English. “It’s difficult to understand the words in TopTi-

er. What does Backorder means?” We consider this aspect right because we think the system should use the users’ language. With the users’ language we mean words, phrases and concepts, which are familiar to the user. If words the employees use today change and are replaced with new one, the employees will not recognise them. As Nielsen claims the user should not have to reflect upon if different words and situations means the same thing. We think it’s good to use phrases the employees already use because they don’t have to relearn new words. Our suggestion is to use Swedish words as extensively as possible, otherwise use phrases the employees are used to, for example language mixed words they know about. By using words familiar to the users, make the system efficient to use.

These conclusions bring out the importance to suit guidelines into the existing situation. The complex of problems with the mixed language in EUP is an example of when it’s not possibly to follow the guidelines without adapting them to the situation. The guideline is contradicted by the situation. The guideline imply “don’t use a foreign language” but the situation demands the administrator to use a language which is familiar to the user otherwise a lot of time will be spend on learning and understanding new terms.

68 Field note 010516, Translated from Swedish
69 Notes from, Supervisor meeting 2001-02-26, Translated from Swedish
70 Field note 010516, Translated from Swedish
71 Nielsen, 1993
*The Choice Scala in the Toolbar*

EUP is supposed to work not only as a report and analysing system but also as “information central” to the employees. With “information central” we mean a way to quickly spread information among the employees. Employees should be able to look at information he needs in his work even if it’s may not be an ordinary work assignment. It’s important to think of two groups of users, those who don’t use Scala at all, but are going to use EUP as “information central” and those who uses Scala today. It’s important to create a good adaptation to both the groups. A flexible system fit both of these groups, which will support the group of users not using Scala. By making the system familiar to the group of users who know the system Scala this leads to a good adaptation. The users of Scala are also the major group and we want to let this big group of employees feel familiar in the new system by retain some parts from Scala. The users who for example are recently employed have to learn Scala since it’s BiW’s business system. Company has, according to the law, to have a business system.

With familiarity we mean the user should recognise him self even if he had not used the system before. It’s according to Grand of great importance to use elements in the GUI familiar to the user. GUI, which uses widgets and conventions already familiar to new users, are easier to grasp because the users will possess knowledge and expectations consistently with their use.\(^72\)

\(^{72}\) Grand, 1999
Our suggestion is to replace the content in the choice Scala in the Toolbar in EUP with the Scala system modules, see picture 13. The modules are familiar to the employees and they know where to find the information. Our idea is to let the users have all modules in the system Scala available through EUP.

The user will recognise himself in EUP if the system Scala in some parts is preserved. Since the system Scala, as we said earlier, has a limited search function it would be easier to find information among the system Scala modules through EUP.

Picture 14 shows the original content in the choice Scala. In picture 13 and 14 you notice the difference between our suggestion and the original design.

We think the “modules” in picture 14 makes it harder to find relevant information, because there is not the same information structure as it’s in the system Scala. According to us the information structure is scattered, consequently the “modules” seem to be more than they actually are. The design differs from what the Scala users are used to, see picture 4. With scattered information we mean for example projects are in several “main modules”.

"Picture 13: Our suggestion of the choice Scala in the Toolbar"

"Picture 14: The original Scala choices"
By replacing the modules in the system Scala in the choice Scala in the Toolbar, see picture 13, it will be easier to learn EUP for the users who are used to the system Scala. It will also be easier to remember because of the similarities between the systems.

It could be seen as information overflow to view all modules in the choice Scala in the Toolbar in EUP, as we suggested, but according to the users it’s important to have all of the modules in the system Scala available.

“I want to know that I have all the needed modules available in my work, that is user-friendly to me. I don’t want to discover while working that I don’t have a specific module available. It will interrupt my work…” (Log Future Workshop, Translated from Swedish)

To work efficiently, according the employees, means to have all the necessary modules available.

Another reason to have all modules available is the possibility to use the Drag and Relate function in full scale. For example if an employee at the Purchase department wants to know a value from the store room and only employees at the store room got the module, he can’t get the information necessary. The employee from the Purchase Department can’t do a Drag and Relate between these modules if he not have the modules available.

General Functions

General functions are similar functions in different systems for example Enter at the keyboard to confirm and the key F1 for help and so on. By keeping general functions in a new system makes it flexible to use. General functions are functions the user already knows about, which make the functions familiar to the user. We observed an example of using Enter to confirm a search. The employee works with purchase orders:

“The user searches for a specific pipe in the search engine in the Information Bar. Then she presses Enter at the keyboard and nothing happens. She clicks with the mouse at the button search instead.” (Field note 010506, Translated from Swedish)

The user didn’t know the function was not available at the moment and therefore the system didn’t support the command Enter, which it normally does.

The general EUP is web-based which help the user to feel familiar with the system and the web-browser have the usual functions the user are used to, for example back and forward functions.

This can also make the user confused. We experienced the help function in the general EUP hard to find. The employee who works with purchase orders:

“The user writes pipe in the search field and choose to search all. /.../she didn’t get any results and starts to look for manuals to get further in the work. She presses the key F1 and the manual appears. She starts to scroll with the mouse. Soon she closes the help window. – It was not the correct help function. (Web-browser) She starts to search among the different menus in the web-browsers toolbar. I can’t find any help!” (Field note 010506, Translated from Swedish)
The key F1 is established as help in most systems. If the user presses F1 in the general EUP the system shows the web-browser’s help function. The help function in the general EUP is a small question mark in the Toolbar, see picture 15.

Our suggestion is to enlarge the question mark button and separate its colour even more from the background. We also want to complement the button with a help function showed as a “main module” in MyPortal in the Toolbar.

A flexible system makes it possible for all user levels to use also users with limited knowledge about the system. As an employee at the Finance department said:

“I have worked a lot in Scala and I know the system but there may be 250 others that don’t know it as much as I do.” (Log Future Workshop, Translated from Swedish)

A good system is flexible to fit even the users who aren’t familiar with Scala. They who never have worked in Scala should be able to use EUP effectively. A flexible system should support all the user groups we mentioned earlier, the novice users, the middle users and the expert users. According to Nielsen a good system has to be flexible for each category of users. Nielsen also writes: “You can collect usability metrics for both novice users and experienced users.”

We think the user should have the possibility to minimise, close and maximise all windows including the three parts in the general EUP and the dialogue windows. Today is this not completely supported by the general EUP. An user experience this:

“The user searches for a specific pipe in the search engine in the Information Bar. /…/ The result appears to a smaller window, like a dialogue window, and the search engine Excite shows. The user tried to maximise the window, but this is not possible in the general EUP. The user uses the scrollbars to look at the entire window.” (Field note 010506, Translated from Swedish)

Picture 17 shows the scrollbar the user has to use to navigate the entire window. This is not a good solution. The user should have the possibility to maximise the windows. It’s annoying for the user to scroll the window.

---

73 Nielsen, 2001, Internet
The different window functions, which were mentioned before, are a general function and the users expect to find these functions in all systems. At our Future Workshop the employees mentioned a disadvantage with EUP, the Information Bar, see picture 9, in the GUI disappears when you are working in the Workspace.

"The window is disposed in three parts, the Toolbar, Information Bar and the Workspace. When I am working in the Toolbar the Information Bar disappear. The information Bar and the Workspace become one part, to show the Information Bar again I have to click at

MyPortal” (Log Future Workshop, Translated from Swedish)

A participator at the Future Workshop suggested a multiple window manager to solve the problem with the different parts disappearing while working. "A multiple window manager should make the work easier. "74 This can be done by let the different parts (Information Bar, Toolbar and Workspace) exist like in picture 18. They could be viewed minimised at the screen in the taskbar properties at the bottom of the screen.

Our suggestion is to let the back function in the web-browser make the minimised windows show again. All the windows showed before the last action will show again if the user uses the back function. It’s important according to Nielsen to give room for the user to explore the possibilities of the system.75 By making actions reversible users can explore their work without being afraid of making errors. The general EUP is web-based and the web-browser supports back functions. You can easily move backwards in your work process. It’s positive if the user feels safe to freely explore the system

---

74 Log Future Workshop, Translated from Swedish
75 Nielsen, Internet
without has to ask for help while making wrong. The users find this as a good operation in the general EUP. The back function in the common Web-browser’s make the function established. You know where to find the back button and you know what is going to happen when using it.

The work would be easier and the user always has the different parts in EUP available since you know where to find them. It’s a familiarity way of hide functions or programs while the user are working.

When the design includes menus the entire area should respond on the action the user did, even though the user doesn’t clicks precisely on the menu name. The quotation shows the problem when the user adapts her profile:

"She clicks at the choice Search and there she chooses among three different choices such as Web, documents and all. She chose all, clicks on Links and first nothing happen. She clicks again and this time at the name and a drop down menu appears and she ticks of some of the link she doesn’t want to view." (Field note 010506, Translated from Swedish)

This is a general suggestion, involving all similar functions. Make the entire area around the function to respond to the actions.

**The Find / Search function**

The general EUP has two search engines, which could be confusing for the user. The Search engine in the Information Bar allows the user to search for documents and at the Internet, Excite. The other search engine, the Find function, allows the user to search for a specific data such as an article number. We don’t think it’s a good solution to separate the search engines, which make the system difficult to use.

The Find function is hard to find if the user doesn’t know where to look. The Find function appears when the user marks a module in the choice Scala in the Toolbar. This is not a good solution because search engines should be easy to find. Gather all the tools in the same place, which lets the user to easy find the different tools. Our suggestion is to have a shortcut to the Find function in the choice MyPortal. A flexible system supports novice users and expert users through for example accelerators such as shortcuts, and good help systems. Support, the expert users by for example making shortcuts to perform frequently used operations fast. To make everyday actions easier it’s important to tailor frequent actions. Accelerators are a good way for the expert to speed up the interaction with the system.77

During an interview we noticed problems in the Find function in the general EUP. The user entered a search word in one of the text-fields and when he should confirm it he became confused, because the button Find was placed wrong.78 Our suggestion is to place the button after the text-fields and in the right corner, see picture 19. To see the original design, see picture 11.

76 Nielsen, 1999
77 Nielsen, Internet
78 Field note 010313, Translated from Swedish
Placing the buttons at the top is a general design for the general EUP. You should place the functions, which operate on a specific object within the object itself. As an example place the buttons close to the operation. By keeping the functions in the object you will probably only see the functions you actually can use. When people look at the screen, they start at the top of the left corner and read fast to the lower right corner.

It’s helpful to have different fields to fill in when you want to make a search. If you have the order date of an order you can search for example the specific date of the order. The user is not committed to search on a specific field; the user has the possibility to exactly choose what to search for. When the user types the search term it’s important to write correctly to get a result. We observed an employee who tried to search for purchase orders from a specific firm, Hansens:

“The user selects the choice Scala in the Toolbar, TTPurOrderHead, and the Find function appears. He writes *hansens* in the supplier field. We asked the employee why he wrote the stars and he said: ‘It limits the search result.’ He presses Enter at the keyboard. He didn’t get any results. Then he writes *HANSENS* with upper-case letters. He still doesn’t get any results. Then the user writes the supplier code for Hansens instead. Now he got a result of the company” (Field note 010516, Translated from Swedish)

The user has the possibility to choose among the different search fields. If the user doesn’t get any results in one search field he has the possibility to search for another parameters. The quotation shows the difficulty to enter a parameter correctly, the user comments it: “It makes the search unsure!”

Reports

There will be about fifteen report-templates in EUP, which will be catalogue structured in the choice Scala Reports in the Toolbar. The information overflow will increase because files outside Scala could be put in EUP. The quotation shows how an employee who works with shipping goods perform his assignment today:

---

79 Agardh, 1998

80 Field note 010516, Translated from Swedish
“An employee at the Goods Department receives a package from a colleague, which he will deliver to a supplier. Before he delivers he has to make a consignment note. He opens the program Excel, we asked why and he told us all parameters were not available in Scala. Therefore he uses Excel based software from the specific package supplier. He enters 010320 in the Shipper date column, then BiW in the shipper column /.../ At last he saved the consignment note and printed it.” (Field note #74, Translated from Swedish)

The general EUP supports different file formats such as Excel, which leads to flexibility in the case the user can use the program he is used to in his work. Finally EUP will be familiar to the user because he doesn’t have to switch program to make reports.

EUP will improve the reports in the case the system supports more parameters. The reports in Scala don’t support all stored parameters in the database, which EUP handles. The general EUP can manage complex values and ennable the values into reports.

EUP will support an easier way to print reports. Compare Scala with EUP by looking at quotations at page 13 and 14. The users are able to schedule the printing of the reports. If the user wants to have the same report every week he could schedule it and the report will be printed in the interval the user choose, which also support an easier way to print reports. The user doesn’t need to remember printing the specific report. The printing actions, for example finding the specific report, are not necessary.

The reports are coded in different colours. For example the main projects is black, projects are grey and the sub projects light grey. The different projects are also coded by numbering in a scale from one to three. For example a main project number always starts with one and the final sum is marked with black. The example is based on the structure of the projects at BiW. We find it as a good initiative to code the reports. The reports give the user a good overview and it’s easy to find values that belong to a specific project.

System Failure

System-bugs and failures in the system should be minimised preferably without any system errors. The database EUP works against must reduce technical problems such as default while updating database tables. As a participant in our Future Workshop said about the database the system Scala uses: “One failure in one module means measuring in many modules, which means failures arise in all of them.” It’s not acceptable to the user to get wrong values during his work, because it makes the system unreliable and may also affect the reports. The result of the wrong updated values in the database causes problems in the organisation. As a future user said: “Failures in the input generate failures in the output in that case there is no change.” Failures in the system are a problem that has be considered. By rebuild

81 Log Future Workshop, Translated from Swedish
82 Log Future Workshop, Translated from Swedish
and rearrange the database the systems uses, the failures will be prevented. If the system failure, the user will be disturbed in his work that may lead to the user enters wrong values into the system.

“Human errors” will always exist but if there is a low failure rate in the system the users don’t failure during the use of the system. If the system generates warnings before something is going to happening, the “human errors” never need to arise.

A warning to the user can be a dialogue window, which asks the user if he is sure about something. For example; if you delete a file and the system asks: Are you sure you want to send file ‘temp.doc’ to the Recycle Bin? The system gives the user possibilities to choose yes or no. When the user failures the system should prevent the failure before the user carry out the failure completely.

Reflections about Design Suggestions

Designing a good system is much about common sense even if it’s difficult to see. For example if you put the numbers backward in a watch it will cause confusion and you have to think one more time before you can answer what time it’s. It’s a different way of thinking and even if it’s funny for a while it will soon become annoying. A designer have to have in mind what we are used to see.

The minor group should also be in consideration during the design process. For example support they who won’t be experiencing any colour. The text should have high contrast for example black text on white or pale yellow backgrounds and avoid grey backgrounds. It’s important to not forget to have the older users in mind and use large font sizes enough to be readable on standard monitors. We have not taken this in consideration, but we want to point out the importance if the system shall support all categories of users.

We consider the employees shall have all information available since BiW has an open information policy. We don’t think it’s good design to show too much information, but it’s important for the user to be able to search among all the information. All information includes the entire information at BiW, which will be connected to the future EUP. Replace information overflow with a good search function!

It’s difficult to get a good system, a system to fit the situation and the persons who are going to use it. Therefore it’s important to complement ethnographic field studies with usability guidelines and involve users in the entire design process.

---

83 Norman, 1988

84 Tognazzini, 2001, Internet
General Problems

Problems are a natural part in ongoing projects. One problem we experienced was the lack of information. Many of the employees at BiW didn’t know anything about the new system, the general EUP. This is a well-known problem at BiW, because the information flow does not satisfy the employees. The way we got notified about this problem was when we sent out the first set of questionnaires. Among the answers we got comments like: -What is TopTier, -I don’t know anything about this, -I have not seen any information.

Another problem was to find our focus as we mentioned earlier in chapter History.

Disagreements at BiW were another problem to us because what one person said was not true to another. When we checked it out we noticed it wasn’t correct. An example of this was a module in Scala, some said it was in use, other said it was not in use. It was hard to know what to believe and it ended in an observation with the users.

When the users didn’t have time for us or didn’t want to meet us, it became a vicious circle, we needed the material from fieldwork and could not go on with our report in other case. It happened we had to travel back home again without any material from the field, which was very frustrating. When we started our project it was the super user only who used EUP. Because of this limited EUP user group it was difficult for us to apply our methods at the beginning of the project. Since EUP not was in use, except of in the super user group, at the time we started our project we decided to focus on the Scala system. Scala will be used through the GUI of EUP as an output system, and by examine the Scala users we circle the future EUP users.

The fact that EUP was not in use except for some exceptional cases limited us in our work. It’s harder to evaluate a system which is not running.

A problem raised when some of the employees didn’t have the right opinion of what we were doing. They only showed us where the systems could be find and it didn’t work out well. We felt like controllers instead of persons who were interested in their work. To avoid this misunderstanding we started the interview to tell the employee in detail about our purpose. We thought it wasn’t necessary because the persons we met had filled in a questionnaire with detailed information about our work and ourselves.

It’s always easy to be wise after the event!
Summary

We have evaluated the usability in EUP and our contribution to the company, ABB Body-in-White AB, is our motivated design suggestions. We know the importance of field studies and Participatory Design methods and by taking these concepts in consideration we have done this report.

Study the work practice is an MDA aspect and implies how the employee actually performs his task. By making ethnographic field studies you can get an insight of how the employee uses the artefacts to perform his task. For example during an observation the observer notice the employee has to use the mouse to press at the search button in the system. The employee forgets it all the time and presses enter on the keyboard instead. The employee is interrupted in his work because he has to move his hand from the keyboard to the mouse. Even though the program did support Enter, as we mentioned, its details like this and to discover facts the employee doesn’t tell you we want to find during study work practice. In traditional usability literature you can read about work process, we want to emphasise the difference between work practice and work process. Work process or workflow describes the different operations during the employee’s work. For example to perform a task the employee writes a value in a field prints enter, goes to the printer to get the document and so on.

To get to our design results we have done an ethnographic field study and practised Participatory Design methods at the company. Our field study was about observing, discussing and interviewing employees at BiW to understand their work practice and their needs. We evaluated the systems together with the users to understand how they work and what they use the systems to.

A milestone in our work was our Future Workshop. Through our observations, discussions and interviews with the users, we developed design suggestions. In our Future Workshop we wanted to discuss these suggestions with the future users to get feedback about our ideas. The discussions in the Future Workshop generated new ideas to take in consideration and important feedback to us, the result was a push in the right direction in our continuous work. An important artefact to us in our Future Workshop was mock-ups. Screen dumps as mock-ups were a good basis for all the participators in our discussions. With the screen dumps all participants could show what they meant and it was easier to all to understand each other’s opinions. The result of our field study and our Participatory Design methods is a GUI adaptation in the form of design suggestions.

An important question to us in our work was: What is a good system? We don’t think it’s possible to develop a system that suits everyone. But it’s possible to develop a system to fit as many as possible. According to us is the most important rule of thumb to develop the software in the specific situation it will be used in. The EUP develop
operator develops a framework and suits the software to the specific company that buys the system.

There are general usability guidelines to follow when you develop a system. Even using them it’s hard to get a good system, a system with usability. To get a system to fit the situation and the person who will use the system it’s important, according to us, to complement ethnographic field studies and Participatory Design methods with usability guidelines.

Participatory Design methods, which mean engaging the future users in all phases of the design process. We have only used the methods in the last phases in the design. The end user is the expert in his or her domain consequently it is important to bring this knowledge into the software development process.

The users have to participate in the design process to get a good system and the designers have to take all guidelines in considerations. By a combination of these two and by putting the system in the situation it will be in use, the designers may success of designing a good, usable system.

A system, which has usability, is easy to use and learn. A system should be efficient to use and easy to remember, which lets the user return using the system after a period of not having used it. A system should have few failures and be subjectively pleasing.

A good system should be flexible by supporting all the different user groups, such as novices, experts and all the users in the middle group.

The language in EUP is supposed to be in English but there are a lot of Swedish terms. As we discussed earlier a user GUI has to fit its purpose and the users in the specific situation. The summary is, even if EUP not follow the guidelines in this case we have found out in our methods, it has to be a language mix if the users are going to be supported and not have to reflect upon English terms they don’t understand.

By keeping EUP familiar to the Scala users through the modules in the choice Scala in the Toolbar in EUP makes EUP more useably according to us. The majority of the employees at BiW are using Scala and will be affected by retain the modules in EUP. In this case you have to adapt the system to as many employees as possibly which are the Scala users.

The fact the general EUP uses a web-browser is good because the browser supports reversal functions. The users have the possibility to reverse their actions and they feel safe to freely explore the system without have to ask for help while making wrong.

From the user's perspective usability is important because it can be the difference between performing a task accurately and completely or not, and enjoying the process or being frustrated.

From the developer's perspective usability is important because it means the difference between success and failure of a system.

From a management point of view, software with poor usability can reduce the productivity of the users to a level of performance worse than without the system. In
all cases, lack of usability can cost time and effort, and can mean success or failure of a system.\textsuperscript{85}

Final Words

We hope we made a lasting impression on BiW in the case they adopt our suggestions.

Thanks to our supervisors at ABB Body-in-White, Mikael Lundkvist and Robert Samuelsson and all the employees at BiW who made this Bachelor thesis possible. Also thanks to Blekinge Technical Institute and our supervisors Bo-Krister Vesterlund and Olle Lindeberg.

*We hope you have enjoyed reading our report!* 

\textsuperscript{85} Diamond Bullet Design, 2001, Internet
-Glossary

Ad hoc
Ac hoc is to do something according to the circumstances. To a specific occasional or individual purpose.

Access
A database program in Microsoft Office.

Alerts
Alerts are warnings to a user that specific data items are performing outside an acceptable range or level. Users can alter the limits of the range in order to set up a more personalised scheme.

Artefact
An artificial object made by a human.

Drag and Relate™
TopTier’s (company that developed EUP) trademarked name for how application integration works for the user. Drag and Relate is really a user drag-and-drop action.

Database
A shared collection of data designed to meet the information needs of an organisation.

Enterprise Unification Portal

An enterprise portal that provides users with access to both structured and unstructured information. An EUP unifies the enterprise by allowing users to cross boundaries otherwise imposed by differences in applications and file formats.

Excel
A program in Microsoft Office which deals with for example calculations.

ERP
An Economy Resource-Planning program for example Scala.

Ethnography
A discipline that tries to understand the human behaviour from the point of view of those studied. The ethnographic study is performed in the natural settings of the workplace.

EUP
TopTier Enterprise Unification Platform is called EUP as acronym in this thesis.

GUI
Graphical User Interface, which provides its user with a way to interact with the technology.

Interface
The interface is a connecting link between any two things not only software-hardware. The interface is for example between man and machine or the interface between medicine and science

**Input system**
With input system we mean Scala, which is a system where you mainly add information.

**ISO**
The International Organisation for Standardisation, international bureaucratic for decree standards.

**Modules**
Scala is catalogue structured and based on modules for example bookkeeping/accounting, customer, supplier, and order/invoicing

**MDA**
Is an educational program: Människor, Datateknik, Arbetsliv: the Swedish acronym for Humans, Computer and Work.

**OLAP**
On-line Analytical Processing. By putting together data in an OLAP cube you have the possibilities to analyse in multiple dimensions. With an OLAP cube it’s possibly to look at data from different perspectives which makes it easier to analyse.

**Output system**
In this case is EUP an output system, we mean the information in the application merely can be seen.

**Participatory Design**
Participatory Design imply that user of the future system together with the software developer participate in the decision that influence the use and design of the system.

**Query**
Relations between tables and theirs field in a database. In this case SQL database.

**Question**
A sum of values which are not compared with any parameters. A question only shows values.

**Scala**
Scala is an economy resource-planing program that is catalogue structured.

**Sub module**
In a catalogue structure a sub module is a module in a main module.

**Super users**
The super users at BiW are seven employees and their responsibilities are to help other users, in the super user domain of competence, with problems in EUP and to pass on questions and problems to the project team.
Inventory

**Literature**

**Agardh, J et al.** A design journey in the unexplored land of usability and user-friendliness, 1998, Bachelor thesis, University of Karlskrona/Ronneby


**Ely, M.** Kvalitativ forskningsmetodik i praktiken – cirklar inom cirklar, 1993, Lund

**Faulkner, X.** Usability Engineering, 2000, Macmillian Press Ltd, London

**Fägerborg, E.** Den första tiden. Fältarbete och arbetssocialisation, 1997, Stockholm


**Grudin, J.** Obstacles to Participatory design in Large Product development Organizations in Schuler, D, Namioka, A, (red.) *Participatory design principles and practices*, 1993, Lawrence Erlbaum Associates, USA

**Hoberg, C.** Precision och improviastion om systemutvecklarens yrkeskunnande, 1998, Trydells Tryckeri, Stockholm
Holzblatt, K, Jones, S, Contextual Inquiry: A Participatory Technique for system Design in Schuler, D, Namioka, A, (red.) Participatory design principles and practices, 1993, Lawrence Erlbaum Associates, USA

Kuhn, S, Design for People at Work, in Winograd, T, Bringing Design to Software, 1996, USA

Lave, J, Wenger, E, Situated learning – Legitimate peripheral participation, 1991, Cambridge University Press, USA


Nielsen, J, Designing Web Usability, 1999, New Riders Publishing, USA

Nielsen, J, Usability Engineering, 1993, Academic Press, USA

Norman, D, The Design of Everyday Things, 1988 Canada

Norman, D, Things that make us smart, 1994, Canada

Runfors, A, Som man frågar får man svar, 1992, Stockholm

Suchman, L, Plans and situated Actions The problem of human machine communication, 1987, Cambridge University Press, Great Britain


Pictures
1. Jakob Nielsen’s concept of usability
2. Overview Scala
3. EUP (TopTier Portal)
4. Different modules and sub modules in Scala
5. Main screen
6. Toolbar
7. The choice Scala
8. The choice Scala Report
9. Information Bar
10. Workspace
11. Find
12. The Information Bar after the user chose Personalise
13. Our suggestion of the choice Scala in the Toolbar
14. The original Scala choices
15. View of the original question mark
16. Our Suggestion of the question mark
17. The search result in Excite
18. Our Suggestion of Taskbar properties
19. Our Suggestion of placing the button Find

Web-sites

Diamond Bullet Design, 2001,

Nielsen, 1998,
http://www.useit.com/alertbox/981004.html 2001-02-06

Nielsen,
www.useit.com/papers/heuristic/heuristic_list.html
2001-02-04

Nielsen, 2001,
http://www.useit.com/alertbox/20010211004.html 2001-02-26,

Tognazzini, 2001,
http://www.asktog.com/basics/firstPrinciples.html
2001-04-24, author of Tog on Software Design etc

http://www.useit.com 010520
http://usableweb.com 010520
http://www.creativegood.com 010520

Own material

Field note #9
Field note #28
Field note #36
Field note #39
Field note #41
Field note #44
Field note #45
Field note #51
Field note #53
Field note #66
Field note #69
Field note #71
Field note #74
Field note #77
Field note 010118

http://www.asktog.com 010520
http://www.usabilityfirst.com 010520

Martina Holmquist, Ann-Marie Pettersson, MDA98
Field note 010222
Field note 010226
Field note 010313
Field note 010315
Field note 010323
Field note 010403
Field note 010425
Field note 010426
Field note 010506
Field note 010508
Field note 010516
Log Future Workshop
Notes from Supervisor meeting 010226
Tape record 1.1
Questionnaires

Appendix

1. Overview of the future EUP
2. OLAP cubes