Communication in a Design Team

-Creating meaning in a design team through boundary objects

Master Thesis
Sara Lindstedt
2005-06-08

LIU-KOGVET-D--05/11--SE

University of Linköping
Department of Computer and Information Science
Cognitive Science Programme
Abstract

This thesis explores object based design team communication. It is assumed that boundary objects in design teams serve as an important communication aid and are considered to have a crucial role in the conducting of multidisciplinary teamwork. Objects, such as design specifications and prototypes, can for example help bridging knowledge gaps between the different interests involved and offer guidance and support in their design work. The aim was to identify the possible problems that might occur connected to the different objects used in a design team. In order to study this, an ethnographical inspired study was carried out. Overall results were that the design team needed to extend their use of objects, in order to be fully supported by them in their work. Further, the existing objects needed to be changed or used differently. A “communication resource hub” was suggested, where all the new and old objects could be gathered. In this “resource hub” there should be room for different models that could be applied as a support for deciding on the right objects for the right purpose.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Aim</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Method</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>Sources of Theory</td>
<td>3</td>
</tr>
<tr>
<td>1.5</td>
<td>The Reader</td>
<td>3</td>
</tr>
<tr>
<td>1.6</td>
<td>Definitions</td>
<td>3</td>
</tr>
<tr>
<td>1.7</td>
<td>Report Overview</td>
<td>4</td>
</tr>
<tr>
<td>1.8</td>
<td>Acknowledgements</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>The design team setting</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1</td>
<td>The multidisciplinary setting</td>
<td>11</td>
</tr>
<tr>
<td>2.1.2</td>
<td>User and client</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>The design work</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1</td>
<td>General models for design work</td>
<td>17</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Criticism of the normative design methods</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Design Documentation</td>
<td>20</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Design Rationale</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Objects in team design</td>
<td>22</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Well-known communication objects</td>
<td>23</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Workgroup collaboration support</td>
<td>33</td>
</tr>
<tr>
<td>2.4.3</td>
<td>The function of boundary objects</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Case Study</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>Ethnography</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Contextual Inquiry</td>
<td>38</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Affinity Diagrams and Work Modelling</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>THE DESIGN TEAM STUDY</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Ethnographical inspired studies</td>
<td>43</td>
</tr>
</tbody>
</table>
# Table of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Waterfall model</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>The spiral lifecycle midlevel</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>The star lifecycle model</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Example detail of a functional design</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>Example of a detailed design</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Example of User’s Model</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>Flowchart document example</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>Wireframe document example</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Detail of a conceptual sketch</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Detail of a storyboard</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>A model of what prototypes prototype</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>Post-hoc worknotes</td>
<td>34</td>
</tr>
<tr>
<td>13</td>
<td>The affinity diagram made for this study</td>
<td>46</td>
</tr>
<tr>
<td>14</td>
<td>The design team’s adjusted version of RUP</td>
<td>49</td>
</tr>
<tr>
<td>15</td>
<td>Example of the design team functional documentation</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>Example of the design team graphical documentation</td>
<td>53</td>
</tr>
<tr>
<td>17</td>
<td>The star lifecycle model</td>
<td>63</td>
</tr>
<tr>
<td>18</td>
<td>A template for the design team’s functional design document</td>
<td>70</td>
</tr>
<tr>
<td>19</td>
<td>Two ways of structuring the functional design document</td>
<td>72</td>
</tr>
<tr>
<td>20</td>
<td>Example of a well-documented graphical design</td>
<td>77</td>
</tr>
<tr>
<td>21</td>
<td>The interface programmer and the system developer working with templates</td>
<td>79</td>
</tr>
<tr>
<td>22</td>
<td>Post hoc worknotes</td>
<td>87</td>
</tr>
</tbody>
</table>
1 Introduction

This chapter introduces the background to the area of design team communication, which is in focus of this study. The aim of the thesis is presented and the method used to conduct the study is introduced. Some definitions of terms used throughout the thesis are given.

1.1 Background

Effective communication among the members of a software development design team is crucial to its success. The design teams working on a project often consist of members from different practices such as programmers and designers. The multidisciplinary composition of the teams is the key to why complex design solutions can be realised but it also means that individuals from different backgrounds and perspectives have to communicate with each other and be able to explain ideas. This is a big challenge for the team members, which often causes misunderstandings and other interruptions to the progress of the projects.

Numerous studies have been carried out in order to gain understanding about how communication can be improved and supported throughout design team work (Badke et al. 2002). Some of these theories are focused around the objects used in design teams. This thesis is centred on two of these object-based theories: the ones of design rationale and boundary objects.

The approach of design rationale concerns the developments of systems that in an automatic way can capture the design decisions made and document them. This is done in order to help teams keeping track of which design decisions are made and why. A problem with these systems has been that they may feel unnatural to use. (Twidale et al. 1993) Therefore the Design Rationale research is at present directed towards how the theories can be developed, with the aim to make it work in a more flexible way that suits the design process better.
The theory of boundary objects is focused on how communication can be improved by using different objects that the team members can talk around in their discussions. These objects have the function of bridging the knowledge gaps between the different roles and can help the team members communicate ideas despite their different backgrounds. The boundary objects that are commonly used within design teams are design specifications, guidelines and prototypes.

1.2 Aim

The aim of this thesis was to identify the problems connected with the use of boundary objects in a real design team. The boundary objects were used as an insight into the operations of the team, in order to gain understanding of; the needs of the different roles and the critical problem areas that need to be solved in order to alter, enhance and encourage communication in design teams. The study was carried out with regard to the following research questions:

- What function do boundary objects have in a real design team?
- When and how are the objects used?
- What are the problems that arise, both externally and internally, when using different objects?
- What can be done to support object based team communication, viewed from all roles’ perspectives?

The main research question was:

*How should different objects within different areas of communication in a real multidisciplinary design team be used, in order to support the team members creating meaning in their design work as efficiently as possible?*

By carrying out a study in a real design team, with regard to these questions, it was believed that useful information would be retrieved about how team communication can be improved. Even though communication between design team members is not always made through physical objects, the focus of the study was on the communication when objects were used.
1.3 Method

A case study was made of a real design team with the focus on the different boundary objects that were in use. The study was based on Beyer & Holtzblatt's (1998) ethnographical inspired method: Contextual Inquiry. The study carried out for this thesis is considered only a case study and the results yield for the design team in question can not be generalised to yield for design teams in general.

1.4 Sources of Theory

The main sources of theory for this essay were articles published in different well-respected magazines. In order to retrieve information on the commonly used objects in design team, internet sources had to be consulted, since they could offer a plethora of objects used in different settings, which was highly valued for the study. The sources used on the Internet were considered to be reliable and information was for example often gathered from university web sites.

1.5 The Reader

This thesis addresses readers with an interest for interaction design, usability and team communication. It is presupposed that the reader has basic knowledge about human-computer interaction, system development and methods for team design work.

1.6 Definitions

Before proceeding, some important terms need to be defined. A design team can often be divided between the roles working on the design of functionality; interaction and graphical appearance and the roles working on technically develop and implement the design. In this essay the roles working on the design will be referred to as the designers and the roles working with development of the code and implementation will be referred to
as the programmers. Together they form what will be refereed to as a design team.

By using the term objects I intend to cover all artifacts that can be used in a design team with the function of either guiding the team or serve as a communication means between roles when transferring design ideas. An object can also serve as visualising tool in the design process.

1.7 Report Overview

The thesis contains, apart from the Introduction, the following chapters:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Theory</td>
<td>This chapter sets the theoretical framework for the thesis by introducing the concepts of what a design team can be and what the theories of boundary objects and design rationale are. Some objects that are commonly used within design teams are presented.</td>
</tr>
<tr>
<td>3</td>
<td>Method</td>
<td>This chapter contains the theoretical background to the method that was used. The concept of interpretative case studies is explained. This is followed by a</td>
</tr>
</tbody>
</table>
motivation and description of why and how the method was applied in the study of the design team.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Study</td>
<td>This chapter gives an introduction to the design team that was studied and the methods and objects that were used by them.</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
<td>This chapter presents the results gained throughout the field study.</td>
</tr>
<tr>
<td>6</td>
<td>Analysis</td>
<td>This chapter presents the analysis made of the results. The analysis framework is presented and six categories of objects are introduced, from which the analysis was made.</td>
</tr>
<tr>
<td>7</td>
<td>Discussion and conclusions</td>
<td>The results are discussed in relation to the problem statement. The conclusions are presented and future studies proposed.</td>
</tr>
</tbody>
</table>

Tabell1: Table of contents.

1.8 Acknowledgements

I am greatly indebted to the following people:
INTRODUCTION

The subjects of the study - the team members of the design team - for introducing me to their work and patiently answering my questions.

Linda Lidman, my mentor at IDA, Linköping University, for giving me excellent guidance and valuable feedback throughout this work.

Kristin and Nicky: for much good advice and support throughout the whole essay.

Special thanks to my family for hosting me and backing me up throughout my work.

Thank you all!
2 Theory

This chapter sets the theoretical framework for this thesis. First the concept of what a design team can be and the implications when working in a multidisciplinary setting is presented. In relation to this the theory of Professional vision (Goodwin, 1994) is introduced, which gives an explanation to why problems can occur when taking stands in different roles in the design work. Thereafter the external interests are introduced, such as the client and the user, to give a picture of what the communication with them means. Further, some of the normative design methods are described, in order to give an idea about how design work traditionally has been structured. Two models that describe the normative methods in a more general way are also presented. Thereafter follows a presentation of design documentation and some of the theoretical approaches to team communication and documentation. These are mainly focused around the theories of design rationale and boundary objects for efficient communication within design teams. The chapter is ended with a description of commonly used objects in team design.

2.1 The design team setting

Design team compositions can be very different depending on the company and the work setting that the team is active in. The roles involved - who they are and what they are called - is rarely exactly the same, from team to team. It is also important to point out that a role is not the same thing as a person, since it is possible for one person to posses several roles at the same time. Several people can also possess one role. It is common that the roles are selected and adjusted to fit each project. (Ottersten et al. 2002) The following are some of the roles that a design team can consist of. The different roles are referred to as “he” in this text, however, naturally both women and men can possess the roles. In order to give an understanding of whom the design team communicates with externally the concept of user and client is also introduced.
The Interaction Designer

The interaction designer is responsible for developing the user experience of the product, focusing on elements such as navigation, layout and user interactions by creating a structure for how the interaction between the product and the user should work. A central concern of the interaction designer is understanding the kind of activities people are engaging in when interacting with the product, since the appropriateness of different kinds of interfaces depends on what kind of activities need to be supported. He is also concerned with ensuring that the user understands the current state of the product. The interaction designer is responsible for preparing user interface specification documents, consisting of conceptual models, process flows, navigational maps, mock-ups and flow diagrams. Further, he is responsible for capturing and documenting design decisions, assumptions and rationale through an iterative process. It is also a part of his role to anticipate misunderstandings and mistakes, ensuring that both the user and the system can easily recover. (Baxley, 2002)

Usability Architect

The usability architect is responsible for leading the design process in a way so that the usability quality is ensured. By making a product usable it is generally meant easy to learn, effective to use and provide an enjoyable user experience. The work consists of planning, following up and reassuring the quality of all usability centred activities in the project. A further responsibility is to visualise the requirements and articulating the link between the expected utility and functionality of the product. The usability architect should have a close collaboration with the client, the project leader, the system architect¹ and the interaction designer. (Ottersten et al. 2002)

The Requirement Analyst

A requirement analyst acts as the liaison between the client and the design team, by leading workshops with and interviewing the client. He is also responsible for making the target group

¹ The system architect has the task of specifying what data will appear on each sequence/page of the application/product. His work is based on the specification built by the requirement analyst.
Analysis. His role is to describe the target groups and the organisation they are active in and expose possibilities and threats that might have implications for the product being designed. The requirement analyst is responsible for transferring the information about the target group to the rest of the team during the development process. (Alexandrou, 2004)

**The Graphic Designer**

The graphic designer is responsible for visualising the interface. By using graphical components and colour he completes the work made by the interaction designer by clarifying and thus deepening the information and the functionality. Knowledge about for example layout, colour and icons are important in the graphical designer’s work. (Ottersten et al. 2002) Web-based applications have brought graphic designers closer to the software development projects than they were before. Buttons, menus and other screen elements of a page are not standardised. Therefore the graphic designers can have a big influence of how a site should look. The graphic designers are for example often tasked with deciding how to best portray the values of a company to the use of colour and layout. (Alexandrou, 2004)

**The Programmer**

The programmer is responsible for building the products functionality. A web project may need just one or a number of different programmers, depending on the size of the project, the type of solutions and which programming languages are required to make the product work. The programmer can be involved during project planning to provide technical advice on the proposed product. During production, the programmer commonly works with the graphic designer. (Web team roles)

**The Project Leader**

The project leader is responsible for planning and managing all the human and technological elements of the projects, from concept to completion. These responsibilities include contact with the client, planning, budgeting and preparation. The project leader can also participate throughout the concept design and arranging of user research. He is also responsible for briefing and managing the rest of the design team; overseeing the content, creative and technical development; overseeing site
testing, release and evaluation; bringing the project to completion, within the for the project decided time frame and budget. (Web team roles)

All together these roles form a multidisciplinary setting and this will be further discussed in the following section.

2.1.1 The multidisciplinary setting

A good way to better understand the complexity that the multidisciplinary setting consists of is to categorise the different interests involved. Wenger (1998) in Marick (2003) suggests that a design team can be categorised as a community of interest, consisting of members from at least two different communities of practice. A community of practice is a group of people who do a certain kind of work together, talk about their work and form a part of their identity from that work. A community of interest involves members of distinct communities of practice who come together to work on a particular problem of common interest. This means that in a design team, members need to learn to communicate with and learn from others who have a different perspective and perhaps a different vocabulary for describing their ideas. This is essential for establishing a common ground and a shared understanding. Since more effort has to be put into this when working across communities of practices, the communities of interest, such as a design team, will face more communication problems than a community of practice. (Marick, 2003)

Some of these communication problems have been pointed out by Preece et al. (2002). Bringing many people with different backgrounds and training together usually means that more ideas are gathered, new methods are developed and more creative design and original designs are being produced. However, the multidisciplinary setting also involves a cost. Normally, the more people there are with different backgrounds in a design team, the more difficult it will be to communicate and progress the generated designs forward. Due to different backgrounds and perspectives, design team members also have different ways of seeing and talking about the world. Hence, what one design team member values as important, the others might, not even see. For example the programmer's understanding of the term representation can be very different from the graphic designer's. (Preece et al. 2002) Considering the number of different roles
with different perspectives involved in a design-team, it is likely that problems will arise.

The communication between different practices has been theorised by Goodwin (1994) and gathered under the term Professional Vision. Professional Vision is focused on perception, vision and the ability to turn an argument around to favour one’s own verdict. Goodwin argues that people see and understand events in socially and organised ways that are answerable to the distinctive interests of a particular social group. Particular social groups construct objects and ways of knowing that define their professional worlds through systematic interactive practices. (Goodwin, 1994)

Working in heterogeneous teams, where roles come from different communities of practices with different backgrounds seems to have a positive impact on the team creativity and the interaction among the team members. As explained earlier, one of the negative implications can be that when team members come from different practices, everyone in the team might not properly understand solutions ideas. However, this provokes questions and the solutions will therefore be analysed to a greater extent than they will be in a homogenous team. Since the heterogeneous team has a need to construct a shared mental model they are forced to ask each other questions. This way of thinking in the design process may cause previously unseen things to come up during discussion. Team members are also forced to find good arguments in order to defend their ideas when confronted by other team members. Hence, the choice of design will be more consciously made and give clearer rationales. Badke-Schaub et al. (2002) argues that this is the main reason why heterogeneous teams repeatedly have been found to outperform homogeneous teams in complex-problem solving tasks.

2.1.2 User and client

To fully understand the design team setting it is important to know the external interests involved in the design team work. The following paragraph therefore introduces what the communication with them concerns.

An overall objective for many design teams is to develop interactive products that are usable. By this it generally means
easy to learn, effective to use and provide an enjoyable user experience. (Preece, 2002) In order to do so it is crucial to involve the user in the design process. However, according to Bevan, (2001) one of the most frequent failures of current design processes is a lack of understanding the real user needs. User centred design requires a detailed understanding and specification of user requirements, and the active participation of users in an iterative process of evaluating whether proposed design solutions meet user needs. A common problem is that many design teams refer to the client and user as the same person. It is important to emphasise the difference between the two: in most cases the client is the buyer, the negotiator, whereas the user is the one who will end up using the system. A solution to this can be to find a receiver among the end users of the product, with the responsibility and authority to ensure the quality of the product. (Ottersten et al. 2002)

The term “client team” is intended to cover both those who pay money for the product, those who posses the operative knowledge about the setting the product should support and those who have knowledge about the market for the product. The client normally makes all the decisions concerning the direction of the project, coverage and costs. Therefore there should be a person or a forum, consisting of the client representatives and some members of the design team, with executive powers. Normally the user should be part of the whole development of the design. Ideally, the user participates through target group analysis and user tests. A common way to organise user participation is to use reference groups. These should mirror the target groups and their cultures for the product. It can serve as a way to return information about the new product to the rest the user group. It can also serve to improve usability quality. (Ottersten et al. 2002)

The aim of this section (2.1) was to give an understanding for what a design team can be, what positive and negative implications can arise when working in a multidisciplinary setting and what the external interests involved normally are. The next section will focus on the work carried out in teams, by describing some of the commonly used methods and models for design work.

2.2 The design work
The work of design teams is generally structured by normative design methods. These have developed throughout the years, from different strategies for designing in architecture, engineering and product design, in order to obtain cost effective and user adjusted results. To understand what activities are involved in team design, some traditional models for team design work and some generalisations of them are now presented. The section ends with criticism that has been raised against these methods.

**The waterfall model**

The *waterfall model* was the first model generally known in software engineering and forms the basis of many lifecycles\(^2\) in use today. It is basically a linear model in which each step must be completed before the next step can be started (see figure 1 below). For example requirements analysis must be completed before design can begin. The names of the steps can vary but normally the lifecycle starts with requirement analysis, moves into design, then coding, then implementation, testing and finally maintenance. A well-known problem with this model is that requirements change over time as business and the environment in which they operate change rapidly. This means that it does not make sense to freeze requirements while the design and implementation are completed. The idea of iteration is not embedded in the waterfall’s philosophy though when used today, some level of iteration is incorporated in most versions and review sessions among developers are common. However, the opportunity to review with the users is not built into this model. (Preece et al. 2002)

\(^2\) The term lifecycle is used to represent a model that captures a set of activities and how they are related. Lifecycle models are used as management tools and are simplified versions of reality. (Preece, 2002)
Another model for software development is the *spiral lifecycle model*. The spiral model incorporates the features of risk analysis and prototyping in an iterative framework, which allow ideas and progress to be repeatedly checked and evaluated. The iterations were not introduced in order to involve the user more but to identify and control risks. Unlike the waterfall model, the spiral encourages alternatives to be considered and steps in which problems or potential problems are encouraged to be re-addressed. (Preece et al. 2002)

*Star lifecycle model*

The *Star lifecycle model* is a model developed within the field of human-computer interaction. It was developed from empirical work on how designers actually perform their work. Unlike the

---

3 Also in Appendix
4 Also in Appendix
lifecycle models, the Star lifecycle does not specify any ordering of activities. The activities are instead highly interconnected: it is possible to move from any activity to any other, provided that the evaluation activity is first worked through. (Preece et al. 2002)

![Figure 3 The star lifecycle model](image)

**Rational Unified Process**

The Rational Unified Process (RUP) is a software design methodology created by the Rational Software Corporation. It describes how to effectively develop software using commercially proven techniques and is a heavyweight process and hence particularly applicable to larger software development teams working on large projects. The RUP defines guidelines and templates for team members to follow during a product’s lifecycle. These are:

1. Develop Software iteratively
2. Manage Requirements
3. Use Components Based Architecture
4. Visually model software
5. Verify software quality
6. Control changes to software

The phases involved in RUP are:

- Inception – the vision of the system and the utility the system will have for the client is specified. The size of the
Theory

A project and what it should contain is also decided during this phase.

- Elaboration - activities and resources are planned. The functionality is specified and the system architecture designed.
- Construction – the system is built from the artifacts produced in the Inception and Elaboration phases.
- Transition – the system is transferred to the end users. The system is delivered to the client and education and support of the system is being offered.

(Wikipedia)

Critique has been raised to the RUP method that it involves too many steps and is not adjusted to fit usability centred work.

2.2.1 General models for design work

Two common ways of modelling design work in a more general way will now be presented. These models are commonly used in design teams, and originally they derive from the more traditional system development methods. The models offer a more general understanding of what design work is about, than the traditional normative methods do, and they are descriptive for design work as opposed to prescriptive. In addition some comments are made on creativity and design work.

Principle, functional and detailed design

This model organises the design process into three different stages:

1) Create a principle design that describes all the different components that the system consists of and how the user navigates between these.
2) Create a functional design that describes all the actions the user can take and what information the system should supply.
3) Create a detailed design that describes the typeface, colours, messages, icons that each interface-component should consist of. (Ottersten et al. 2002)

The first two steps mostly involve the interaction designer, the requirement analyst and the usability architect, whereas the third
step is dominated by the work of the graphic designer and the developers. Each of these stages commonly results in a specification or a document, which will further described in section 2.4.1.

**Conceptual and physical design**

Another model for design is presented by Preece et al. (2002) which divides design work into two stages: conceptual and physical design. The former is concerned with the development of a conceptual model that captures what the product will do and how it will behave, whereas the latter is concerned with details of the design such as screen and menu structures, icons and graphics. The design emerges in these stages iteratively, through repeated design-evaluation-revaluation cycles involving users.

The conceptual design is concerned with transforming the user requirements and needs that have been gathered into a conceptual model. The conceptual design stage as a whole involves thinking through a design problem, understanding the user’s needs and coming up with possible conceptual models. These are then prototyped and evaluated with respect to user experience goals. The prototypes are changed according to these and other design implications, thinking through whether the changes have improved the interface and interaction, and so on. Beyer and Holtzblatt (1998) recommend holding review meetings within the team to get different peoples perspectives on the data and what they have observed. This helps to expose the whole team to different aspects and will thereby give a deepened understanding. Ideas will emerge as this understanding of the requirements is established and these can then be tested against other data and scenarios, discussed with other design team members and prototyped for testing with users.

The physical design involves considering more concrete, detailed issues of designing the interface, such as screen or keypad design, which icons to use, how to structure menus and so on. There is no rigid border between the conceptual and physical design, since the conceptual design evolves through the creation of prototypes and the design gradually becomes more concrete. It is important though; that the conceptual design is not too constrained by the physical aspects too early, as it may inhibit creativity and focus on building the right kind of product.
In the process of creating an interactive product, it can be tempting to begin at the physical level of the design: work out how to design the physical interface and what interaction styles to use. However, a problem in starting on the physical level is that usability goals are easily overlooked. Even though it is necessary at some point to decide on the physical aspects of the design, it is better to make those decisions after understanding the nature of the problem space. It is important that design teams articulate why a certain solution should be chosen. Clarifying the primary usability and user experience goals should therefore be a central part of working out the problem space. (Preece et al. 2002)

2.2.2 Criticism of the normative design methods

Some problems have been pointed out with the normative methods. Researchers such as Badke et al. (2002) argue that designers in practice rarely follow methodology based on normative theories. Mainly because the methods neglect many of the specific factors and constraints that designers need to cope with in their daily work, such as economic constraints, time pressure and how the team work itself is conducted. This can for example result in that documents are not being updated or communicated enough, and there is little reflection over the growing, shared understanding about the design that is created. Schmidt et al. (2002) suggests that it is hard to grasp and translate the work of designers into a formal, normative method, since the setting it is carried out in is often complex. In the same time as it is individual, it is also team based and multi-disciplinary. It also engages multiple professional competencies and perspectives. This kind of work encourages fluent transitions and often several people, or no one at all, feel responsible for the same task and design work is therefore difficult to prescribe.

Löwgren at al. (1998) suggest that the problem often is related to the fact that the work of design teams is often centred on delivering the products on time which results in that work is not made from long-term perspective. It is then hard to follow methods whose benefits are only visible in the long run. The documentation of design decisions is an example of this. It is something often not highly prioritised in design teams, since it takes time and effort to do it. When working from a short-term perspective the benefits of the methods might not be seen.
Schön (1983) is probably the researcher who has taken the most controversial approach by arguing that the design process simply can not be grasped by any methodology at all. He proposes that the work of designers resemble the work of an artist who applies different kinds of methods in a flexible manner. This is done through a process of appreciation, action and re-appreciation, when the designer constantly is reflecting on his work. As a solution to this problem, Badke et al. (2002) suggest that the normative design guidelines should be replaced with a methodology that does not start from a normative point of view but from where practitioners are now. By this they mean that it is important that methods and guidelines take into account the constraints that the practitioners face in their everyday work; such as time constraints, financial constraints and the cognitive overload, for example when working on multiple projects that must be treated simultaneously.

This section presented some of traditional models commonly used in design team work and the criticism towards them. The following section concerns the design team documentation, which is an important part of the design team work, when communicating design ideas.

### 2.3 Design Documentation

One form traditionally used for communicating a design is documentation. This can be a description of how something will work or a diagram showing its components. According to Dix et al. (1997) documentation of design decisions can help a design team to get a good view over the system, in order to understand what is being done. The purpose is to:

1. Describe the system
2. Transfer information
3. Function as a tool during the development of software
4. Explain the result
5. Preserve knowledge for future

To be effective, team members must constantly share information and ideas and therefore it is crucial *how* the documentation is structured and what elements it contains. A trouble when communicating a design through documentation is that a static
description cannot capture the dynamics of behaviour and for an interaction device it is necessary to communicate to the users what it will be like to actually operate it. Preece et al. (2002) suggests prototyping (see section 2.4.1) as a complementary communication tool. By using prototypes potential client misunderstandings can be overcome and they can also be used to test the technical feasibility of a suggested design and its production.

2.3.1 Design Rationale

As mentioned above, two of the purposes with documentation are to “explain the result” and “preserve knowledge for the future”. This is also what the term Design Rationale is focused around. A lot of research has been carried out within the area on how to support the work in a design team throughout the design process. The widest spread is probably the theories on recording the rationale that the members of a design team use to arrive at different design decisions. (Twidale et al. 1993) Design Rationale is a general term that contains a number of approaches; all with the aim to support designers in their work by structuring the design decisions and supplying basic data for decision making. The creation of Design Rationale is an attempt to explain why a design is the way it is. Capturing the sequence of decisions made when producing a design, through a passive recording of the design process as it occurs does this. The intent is to help designers to discuss about a design that has been made and to produce a result. In this way it can help other designers to understand why it was made the way it was and it will be easier to grasp what has been done and why. (Dix et al. 1997)

However, a number of problems have been noted with the notations developed for Design Rationale. One is that the notations have shown problems with acceptability and that they force the designers to decide at an early stage on the nature of a piece of information. A study carried out by Twidale et al. (1993), show that it is important for notations used in the early stages of the design process to support the variability in design activity among the designers. Design teams need a variability and flexibility when using Design Rationale notations, since designers work in different ways. Another important feature of Design Rationale notations is to support a frequent and rapid revision of the design, since concepts mutate over time, which will lead to evolutions of the design structure. Further it showed that the use
of text notes in the notations, allowing the designers to comment on a design, appeared to be very useful when they wanted to make a comment on another team member’s design. Also designers were keen on using a shared view of the system. (Twidale et al, 1993) Some examples of modern design rationale systems will be presented further in section 2.4.2.

2.4 Objects in team design

Much attention in design theory has been paid to the role of artifacts and objects that designers use in their work. Research on communication in design teams has switched from mainly concentrating on social psychology, group psychology and communication theory to a more conceptual attention to the role of artifacts, or objects, within coordinative practices. Schmidt et al (2002) argues that the material work settings and the artifacts used within these, play a crucial role in the effective co-ordination and alignment of co-operative work.

One of the theories that have developed about the role of objects for efficient team communication is that of boundary objects. As mentioned before, the multidisciplinary settings that design teams normally work in, can introduce communication difficulties. Boundary objects are objects, such as artifacts, documents and even vocabulary that can help people communities build a shared understanding and thereby coming over this communication difficulty. Marick (2003) argues that communication in design teams is done best by assuming high-bandwidth, highly iterative, face-to face conversations and describes the boundary objects as the means for supporting these conversations. One of the important functions that the objects have is to serve as an aid for planning and structuring work and thereby leading the design forward. All sorts of material artifacts such as drawings, binders, photos, plans, lists and models have this function. These artifacts/objects are to find on walls, shelves, desks and so on. (Schmidt et al. 2002) A common problem is that each community involved in a multidisciplinary design team can only have partial knowledge of what the other communities work is about and the boundary objects can be a means for bridging these knowledge gaps. Further, they can serve to improve communication with the client and the user. (Arias & Fischer, 2000)
2.4.1 Well-known communication objects

Just as in the case of which roles that are represented in a design team, the same yields for the objects used: it depends on the design team setting, the projects they are involved in and the team’s own preferences. The following is a presentation of some of the commonly used objects within design teams. First some of the objects created during the preliminary design work, such as personas and goal directed scenarios are presented. These are made in order to translate the goals into design and to make sense of user data and are used to communicate the user’s perspective through the design process. Thereafter the objects used within the principal design-, functional design- and detailed design- work model are presented. The documents produced during these steps can serve as a good example of what a boundary object can be. The reason why the objects are presented from this model, and not the model of conceptual and physical design, is that documents such as “functional sketch” and “detailed sketch” have become well known objects to designers. The model has therefore become a common way to structure design work in modern design team projects. The documents produced in each of these phases are in theory referred to as the principal -, functional- and detailed specification. The goal of each step and respective object is to make critical design decisions more conscious. The designers mainly produce the documents but all roles should be involved in making them.

Then follows a presentation of other commonly used objects produced during different stages of the design process. Some models that are adjusted to fit different areas of communication, which can be used complementary to the functional design, are described. Further, some objects used to define and illustrate the information flow and the state of the system, such as flow charts and wireframes and the concept of moodboards and other graphical objects are described. Finally, the concept of prototypes is introduced; objects used to illustrate and visualise design ideas to the user, client or other team members.

**Personas and persona-based scenarios**

*Personas* are a user model, developed by Alan Cooper (2003). The personas are based on the behaviours and motivations of
real people and represent them throughout the design process. The personas composite archetypes based on behavioural data gathered from many actual users through ethnographic interviews. By understanding the personas, the design team gets an understanding of the user’s goal in specific contexts and thereby they become a critical tool for translating user data into design frameworks.

Personas can be used in persona-based scenarios. Persona-based scenarios are concise, narrative descriptions of one or more personas using a product to achieve specific goals. Scenario content and context are derived from information gathered during the first phases of the design. Using scenarios is a way of making use of a specific story to both construct and illustrate design solutions. They are anchored in the concrete, but permit fluidity since any member of the design team can modify them at will. Cooper (2003) suggests that designers should role-play personas as the characters in these scenarios, this leads to a synthesis of structure and behaviour -typically at a whiteboard- and later informs the detailed look and feel. Another technique that can be used is use cases. Use cases are descriptions of the user-system interaction and are normally produced by the interaction designer. Cooper (2003) describes the use cases as “exhaustive descriptions” of functional requirements of the system, focusing on low-level user action. An important difference between the two techniques is that goal-directed scenarios are focused on defining the behaviour of a product, whereas use cases do not typically focus on this. Therefore many assumptions about the form and behaviour remain implicit. Cooper (2003) suggests that use cases can be useful for determining that a product is functionally complete but they should be deployed only in the later stages of design.

**The Principal Design documentation**

The principal design describes the basic components that the product consists of and how the user navigates between these and can serve as a boundary object in many different situations. It can be used as a communication object between the interaction designer and the requirement analyst, in order to make sure that a proposed solution corresponds to the demands of the system. It can also serve as a communication tool at an early stage of the design in order to describe different solutions to the client, or at a later stage when justifying why some solutions have been declined. Finally, the interaction designer,
the system architect and the programmer can also use it in order to try out the complexity of developing and maintaining a suggested design. (Ottersten et al. 2002)

**The Functional Design documentation**

The functional design specification is a blueprint for how the designer wants a particular web project or application to look and work. It details what the finished product will do, how the user will interact with it and what it will look like. Further, the visual and audio effects that the product should give in order to give the user the expected experience are described. A key benefit with the document is in streamlining the development process. The developer working from the document has; ideally, all of their questions answered about the application and can start building it. The client must also approve the specification and thereby the design team is ensured that they are building nothing less that what the client is expecting. When the functional design document is completed, there should be nothing left to guess or interpret. Different tools can make the document, and in different formats and styles according to how the documents will be used: will they be online, will they frequently be merged with other documents and so on. Some software programs used commonly to create the documents are word, Photoshop and Visio. (Smith, 2001)

A well known problem when creating the functional design is that it is hard to describe the way the user will interact with the product, since it is not linear. There is not one sequence of actions that the user might take but several and often there are more than one way to reach a specific function. Ottersten et al. (2002) suggest that the functional design therefore should start with describing each component’s behaviours and then return to the use cases in order to check if what has been described will satisfy the aims of the user and the target group when using the product. By applying a thorough functional design and writing everything down will result in a decrease in development time and a more efficient application development process and possibly better applications and happier clients.

---

5 Use cases are carried out when identifying the user needs and establishing requirements. They focus on user goals but the emphasis is on a user-system interaction, rather than the user's task itself. The term scenario is used within the context of use cases. (Preece et al., 2002)
The detailed design document

The detailed design document is the developer's blueprint. It provides precise directions to the programmers about how basic control and data structures will be organised. Typically, the detailed design document consists of tables and diagrams that translate the functional specification into data structures, data flows and algorithms. The detailed design describes font, colours, message texts and icons that each screen picture should consist of. It contains descriptions of all the rules for how the components should behave, for example exactly what lead texts, messages and other text material should say. In the detailed design each window/page/message is adjusted to the real system and the situation it will be used in, by for example grouping the material sand setting the colour of the content. The type of project will decide how detailed the descriptions are. In bigger projects that are in progress for a longer period of time, involving several programmers and designers, the design should be fully detailed. In smaller projects the detailed design can be limited to describing only the most important parts of the product. A detailed design can contain a suggestion of the graphical design but can just as well be presented without it. If choosing the latter, there is a risk that the design document gets stuck in describing details. (Ottersten et al. 2002)

---

6 Based on Ottersten et al.’s document example. Ottersten et al. (2002)
7 (www.epri.com)
Adjusted models

Smith (2001) suggests different models should be made as a complementary part of the functional design document. They are made in order to sort the information gathered through user studies and can be a way to convey an understanding of the components that make up the system. The functional specification writer is responsible for creating information models to help convey the concepts of the application or product being developed. Smith (2001) suggests three different models: a user model, a designer's model and a programmer's model.

The user model is based around the user's perception of the system: not what the system actually is, but how the user will perceive it. Therefore things like “text fields” or “recursive paths” are not incorporated. By building this model it is easier to develop use cases and use personas. The designer's model serves as a nuts and bolts model for the functional specification writer. It makes it easier for him to break things into objects and classes since the model make the whole system less complex. It defines the interface components and relationships and details the available objects in the user's universe and how they can use them to accomplish certain tasks. The programmer's model is typically only relevant to the programmer. Ideally the user and designer models should be passed on to the programmer who

---

8 Ottersten et al.(2002)
would then build the application. However, every programming environment has inherent limitations and those constraints must be incorporated into the designer’s model. Smith (2001) suggests having a meeting with the programmer assigned to the project and go over the designer’s model with them. From that, the interaction designer will get an idea on what the technical implications are.

![Figure 6 Example of User’s Model](image)

**Technical diagrams: Flow charts and wireframes**

Another way to organise the information is to create flow diagrams, or flow charts, when trying to work through a lot of information or a very deep navigation set. This document should be included in the functional specification. The flow chart diagram illustrates the navigational elements and organises the information. The navigational path through the information should be included on this diagram, such as parent-child directories, where the user can jump to from each page.

At the stage when the product is defined and the functionality and information is mapped out wireframes and mock-ups should be created. Smith (2001) suggests that the art director should make the latter, since he might spot holes in logic and raise good questions. A wireframe is a mock-up of the page that only addresses the layout, not aesthetics. After creating the models, flowcharts, wireframes and prototypes, Smith (2001) suggest that the functional specification can be written.

---

9 Smith (2001) ©. Also in Appendix.
Smith (2001) also proposes that a *design document* should be made: as kind of pre-functional specifications document and allow people to review it and give their feedback. The document is a tool for building consensus and simultaneously managing the expectations of people about what is coming. The design documents is a middle step in the functional design process in order to slowly bring the application in focus, reveal some details but not all about the overall structure of the site and broadly highlight what functionality will be available to the user to do what.

**Prototypes**

A common object used in design teams for visualising design ideas is the prototype. Preece et al. (2002) divides prototypes into two categories: low-fidelity and high fidelity prototypes. A low-fidelity prototype does not look very much like the final product. For example it is built by very different material, than the final version should be made of, such as paper and cardboard, rather than electronic screens and metal. The purpose

---

10 Figures from Smith © (2001). They can also be viewed in appendix.
is to explore different ideas. Hence, they are never intended to be kept and integrated into the final product. The benefits with using low-fidelity prototypes are that they tend to be simple, cheap and quick to produce. Low-fidelity prototyping often relies on sketching. Storyboarding is one example of how sketches can be used, when a series of sketches are put together, showing how a user can perform a task using the device being developed.

Some of the disadvantages of the low fidelity prototypes are that they have navigational and flow limitations, offer a poor detailed specification to code and only have a limited utility after requirements are established. (Preece et al. 2002) Even though the representational capacity of sketches is limited, since not even the most sophisticated drawings fully can substitute the actual experience of an architectural environment, sketches as a thinking tool have a capacity well beyond what is actually contained in the sketches. As representations drawings should be seen as extensions of the persons who use them to aid in thinking. (Laseau, 1989) Preece et al. (2002) argues that sketches, and other kinds of low-fidelity prototypes, particularly serve an important purpose in the early stages of development, during what she calls the conceptual design; when transforming the user requirements and needs into a conceptual model. The designer can then easily switch between sketching concepts and details, moving between a low and high level of representation. Further on in the design process, when the design has reached a more developed stage, focus will instead be on testing the design and discuss it more in detail. At that stage a more developed and advanced prototype will be needed.

Figure 9 Detail of a conceptual sketch

Figure 10 Detail of a storyboard

11 www.sapdesignguild.org
High-fidelity prototyping involves more materials that would be expected to find in the final product and hence, it looks much more like the final thing. A high-fidelity prototype can for example be a digital mock-up, built by software tools such as Macromedia Director or Visual Basic, and can for example be used to test the design on users. (Preece et al. 2002) Some of the benefits with high-fidelity prototypes are that they are fully interactive, user-driven and can offer a “look and feel” of the final product. The disadvantages are for example that they are more expensive to develop, time-consuming to create and not effective for requirements gathering. (Rudd et al. 1996) Marc Rettig (1994) points out several problems with high-fidelity prototyping, such as that they take long to build, testers tend to comment on superficial aspects rather than content, developers are reluctant to change something they have crafted for hours. A software prototype can also set expectations too high and just one bug in the high-fidelity prototype can bring the testing to halt.

Smith (2001) argues that it essential that a prototype is made at some stage of the functional design, before the actual specification is written. It will give the interaction designer a possibility to iterate his design much faster and it will give colleagues and stakeholders a possibility to look at the application and give specific and useful input about what is being done. Allowing this format for presenting ideas is important since not everyone can look at an informational flowchart and realise something is the way they want it but most people can when looking at a prototype. For example a prototype for a web application can at this stage be a set of static html pages set together to show the key pages of the application and the user interface.

A different perspective on prototypes

Houde & Hill (1997) present a different perspective on prototypes in their article “What do Prototypes prototype”. They are critical to the traditional way of categorising prototypes, such as dividing it into low and high fidelity prototypes, since they think that such divisions take focus away from the issues that should really be in focus. They argue that focus must be on the purpose of the prototype and they emphasise the importance of that the designers and people using the prototypes understand its role. This is particularly important since prototypes often are used for a broad audience and if for example used in multidisciplinary teams there will be different expectations on
what the prototype should deliver. In order to support designers
in their use of prototypes, Houde & Hill suggest a model to
decide what the focus of the prototype should be (see figure 11).
The model represents a three-dimensional space, which
 corresponds to important aspects of the design of an interactive
artifact. They define the dimensions of the model as role, look
and feel and implementation. “Role” refers to the question about
the function that an artifact serves in a user’s life –the way in
which it is useful to them. “Look and feel” denotes questions
about the concrete sensory experience of using an artifact – what
the user looks at, feels and hears while using it.
“Implementation” refers to questions about the technique
through which an artifact perform its function – the “nut and
bolts” of how it actually works.

Given a design problem designers can use the model to separate
design issues into three classes of questions which frequently
demand different approaches to prototyping. A prototype may
explore questions or design notions in one, two or all three
dimensions of the model. Houde & Hill give the following
practical suggestions to designers: define “prototype” broadly,
built multiple prototypes, know your audience, and know your
prototype and prepare your audience. Prototypes should be
defined as any representation of a design idea: sometimes very
simple representations make highly effective prototypes. When
working on complex problems designers must be prepared to
throw some prototypes away and to use different tools for
different kinds of prototypes. Designers must be prepared of that
broader audiences might require higher resolution
representations and some prototypes, such as interactive
storyboards, may work well for a design team but not for
members of the supporting organisation. Further, the audience
must be prepared for viewing the prototype; especially clarify
what is and what is not addressed by the prototype. Houde &
Hill argues that by using the model, prototypes will have a more
clear purpose and designers can then better use them to think
and communicate about design.
2.4.2 Workgroup collaboration support

The section above introduced some commonly used objects in design teams. However, research is currently carried out on how communication can be improved by the use of new means, for example by using new technology. One example of this is research carried out by Jonas Löwgren et al. (2002). They suggest a new object for team communication by a concept design made of a system which they call “post-hoc work notes”. It is a database where video and audio recordings are stored together with documents and email. By recording meetings, conversations and other professional communication the system would provide powerful tools for subsequently processing and accessing information. One guiding observation that the team based their design work on, was that design teams can not know at the time of recording their daily work what information will be useful later. Hence, Löwgren et al.’s approach is to capture as much information as socially possible and then concentrate on powerful tools for the subsequent processing, accessing and use of the stored information. The use of the system includes accessing and managing existing database contents as well as creating new contents. For example a conversation from last week can be skimmed, searching for historical information on a specific topic, looking for inspiration, and team members can remind themselves on who said what. These are all examples of accessing existing material. Examples of creating new contents include adding a remark on a topic of study, making annotations, talking during a recorded meeting, preparing a summary on topic, and so on. Löwgren et al. (2002), means that the multimedia communication holds a great potential in many fields of digital media use. The “post-hoc work notes system” is one way of illustrating how it can be better used than it is in many other systems available at present, who often are too heavily inspired by the sequential nature of VHS tapes and music CDs.

---

12 Model originally from Houde & Hill (1997)
2.4.3 The function of boundary objects

As pointed out earlier, the boundary objects have several important functions within a design team. Some important functions have already been introduced in section 2.3. However, they concerned design documentation specifically and the theory of boundary objects involve many other types of objects and Marick (2003) and Schmidt et al (2002) have some more philosophical and general ideas about the function objects can have. In this section ideas are introduced further.

One of the most important is that they serve as a common point of reference for conversations, in the way that they can help communities in general, to agree that they are talking about the same thing. However, Marick (2003) points out that a common problem is that members of communities of interest attach different meaning to the boundary object. In a multi disciplinary design team this can result in that an object, which for the programmer for example serves as a reminder to change class definitions in their code, might represent something very different to the other team members.

The objects can also serve as a means of co-ordination and alignment. For example they can bring the client and the design

---

13 Löwgren et al. (2002) Can also be viewed in Appendix
team together, in order to agree on what they want and which ideas should be developed. This can be done through objects such as index cards\textsuperscript{14}, which can be a successful and simple way to prototype, an interaction when for example developing websites. The index cards can serve as a way to co-ordinate the different roles but also to create a dialogue with the user about the design that is being developed. (Marick, 2003) The use of coordinative artifacts is connected to their specific physical and graphical form. The way text, sketches, arrows are laid out and organised in the design documents, is meaningful and may indicate relevance. (Schmidt et al. 2002)

Further the boundary objects can serve as a means of translation; despite that they as earlier mentioned can be attached by different interpretations. If continuing with the example of the index card\textsuperscript{15} it can help the programmers understands more about the actions the user needs to take when using a system, the index card thereby can be used to smooth the process of explanation. (Marick, 2003) They thereby also serve as a medium for visualising ideas, since material artifacts are publicly accessible. Their state can be inspected by other team members and what others are doing to the object can be noted and made sense of. (Schmidt et al. 2002)

The objects can also be used as working agreements that can be followed up and adjusted as needed throughout a project. Thereby the different roles can meet around the object in order to agree on what direction design work should take. An important quality of the objects is therefore that they are plastic enough to adapt to changes. (Marick, 2003) They are also important in the way that they can satisfy different concerns simultaneously. For example, having brief task descriptions for what each role should produce throughout a project can satisfy the programmers that they are not committing to do more than they can. In the same time it will also satisfy the client experts that something of actual business value will be produced. (Marick, 2003)

Schmidt et al. (2002) argue that in order to understand the plethora of artifacts/objects, one must take into account that

\textsuperscript{14} Index cards are a way to prototype an interaction and is used when developing websites
architectural\textsuperscript{16} work is different from many other types of work. It is for example different from ordinary administrative work in the sense that the work has never been done before but is constructed through the process of design; planning and construction. The artifacts become the fundamental means for making the “not-yet existing” and “in-the process-of-becoming” work and they are therefore crucial for the creative design process.

\textit{Knowledge boundaries}

In software development projects, knowledge is both a source of and a barrier to innovation and creativity among the members of a design team. What drives innovative problem solving within a function can actually hinder problem solving and knowledge creation across functions. Specialised knowledge can therefore cause deep problems and create \textit{knowledge boundaries} between different functions involved in a project. At the same time as these knowledge boundaries become a critical challenge, they are also a constant necessity: much of what an organisation produces has a foundation in different kinds of specialised knowledge. Modern competence is gained through relevant, hard-earned knowledge that is often very specialised and orientated to solving a particular problem. The high level of self-investment makes individuals committed to their own knowledge and perhaps reluctant to accept that of others. The cross-boundary challenge therefore means that to resolve the negative consequences that arise when working across functions, individuals have to be willing to add to their own knowledge and also be capable of influencing or transforming the knowledge used by individuals from other functions. (Carlile, 2002) Boundary objects can be one way of helping to bridge these knowledge boundaries.

\textsuperscript{16} Schmidt et al. (2002) describe the role artifacts have for architects; however, they declare that their observations should yield for all sorts of design work.
3 Method

In the following chapter the methods used to conduct this study are presented. Since the focus of this study was a design team’s communication in practice, an ethnographic approach was chosen. When working from an ethnographic approach there are several techniques to choose from, by which the most common are interviews and observations. For this study participatory observation and Contextual Inquiry were followed. Contextual Inquiry is a data gathering technique method based on ethnographic methods and is adjusted to be used in design situations and was therefore found to be appropriate for this study. Since focus was on the objects used by the design team, and it was not clear what these were and how they were used, it was believed that the method of Contextual Inquiry would offer a good way for finding the desired data. The study carried out for this thesis is considered only a case study and the results yield for the design team in question can not be generalised to yield for design teams in general.

3.1 Case Study

A case study means the study of a smaller, well-defined group. A case can be an individual, a group of individuals, an organisation or a situation. A common argument for choosing a case study is that it will bring the researcher as close as possible to the area of interest. Since the method brings the researcher out into direct observation in natural environments, it is possible to study subjective factors such as thoughts, feelings and wishes, whereas experiments and surveys often is based on deductive information, from for example tests and protocols. A case study can be carried out when there is no interest or time for making an exhaustive study with several study groups from which the results can be generalised. Case studies also tend to have a broader net for fetching up information whereas experiments and surveys have a more narrow focus. A case study can be the appropriate choice when

\[17 \text{ Author’s translation}\]
the information gained from the participants can not be judged from veracity but rather from credibility. The purpose is not to reach the correct or genuine interpretation from the facts, but rather to clear away wrong conclusions, so that you finally will reach the best and most convincing conclusion. (Merriam, 1994) This was also the starting point for the study carried out for this thesis. The results gained in the study are only case specific and can not be generalised. However, as pointed out, case studies have many advantages and the results gained by conducting them are still interesting.

3.2 Ethnography

One of the traditions within qualitative research is ethnography, which has evolved as an attempt to describe culture or aspects of culture by thorough close-up studies of a particular social and cultural environment. (Eriksen, 2001) Hymes (1996, p. 35) refers to Leininger who defines ethnography as:

“...the systematic process of observing, detailing, describing, documenting and analysing the life-ways or particular patterns of a culture (or subculture) in order to grasp the life-ways or patterns of the people in their familiar environment.”

The method describes what is common in a culture. A subculture can be a hospital, a clinic or a department for health care. The advocates of an ethnographic approach argue that it is the best way for making an inductive analysis and minimising the risks of using prejudices. Olson (2001) says that when using this method, the results are not replicable but the aim is to make small, transferable observations with exact descriptions for content. The limits of ethnography are that the natural environment consists of complex factors, which make it difficult to isolate them and control all variables. Further, the method can be very time-consuming. Eriksen (2001) points out that if the researcher carries out fieldwork in a setting where he masters the language and cultural conventions, there is a risk that he takes too much for granted since the culture is already familiar to him.

3.3 Contextual Inquiry
Contextual Inquiry is a part of Contextual Design, which is a methodological framework for design developed by Beyer & Holtzblatt (1998). It is grounded on the principles that the key to good data is to go where the work is happening and observe it while it happens. The core premise of Contextual Inquiry is to go where the client works, observe the client as he or she works and talk to the client about the work, in order to gain a better understanding for the client. The following sequence refers to Beyer & Holtzblatt’s description of Contextual Design (1998).

When spending time in the user’s environment, every event can be a starting point for discussing similar events in the past. Thereby the researcher can get deeper information about interesting aspects of the user’s environment and work in a natural way. Also, artefacts of work, such as papers, documents and notes can trigger conversations about why they are used and how their structure supported their use in a particular sense. When retelling a particular event, the user/person will be immersed in doing the same activity with all the reminders and triggers that doing that activity provides something, which will give the designer rich information. The starting-point is always that the client/user is the expert in their work and the interviewer is not. There are four principles that guide the adoption the technique: the principles of context, partnership, interpretation and focus. Each principle defines an aspect of the interaction.

**Context**

Since people normally tend to abstract when speaking about their work it is important to observe them in their real work situation and that way get more detailed information. Retelling a past event is difficult since so much of the context is gone. The interviewer must then listen for what is being left out. The aim of the interviewer is to see the work as it unfolds in the client’s workplace in order to get as close as possible to the real work situation. That way he or she will be able to gather ongoing experience, rather than summary experience and hence, get more concrete data than abstract data. One way of conducting Contextual Inquiry is to conduct one-to-one interactions that last for two to three hours, in which the person being observed does her own work and discusses it with the interviewer.

**Partnership**
Contextual Inquiry is carried out with the approach that the interviewer and the worker are collaborators in understanding the work, though the only person who really knows everything about his work is the one doing it. It is important that the interviewer creates a relation to the interviewee that looks like a partnership or that of an apprentice. This will engage him in a conversation about the work, making him aware of aspects of the work that were formerly invisible and thereby establish a partnership of inquiry into the work practice.

A basic pattern within Contextual Inquiry is that of withdrawal and return; periods of watching work are interspersed with discussion of how work is structured. As the observation is progressing the interviewer will alternate between watching and probing. At some point he or she will notice the structure underlying an aspect of work and will then interrupt to talk about it. This causes a break and both the interviewer and the interviewee will withdraw from work and discuss the structure that the interviewer found. When having talked about that, the interviewee will return to ongoing work and the interviewer returns to watching.

**Interpretation**

Naturally the data brought back from observations must be interpreted. The interviewees’ words and actions must be analysed to get an *interpretation* of what they mean together. The goal is to understand what it implies about work structure and about possible supporting systems. By creating and maintaining the right relationship with the interviewee during the observation the interviewer will ensure that the interpretation is true. As the partnership is further established, the person being observed and interviewed will be less willing to allow the interviewer to leave thinking the wrong way.

**Focus**

It is of great importance that the interviewer is working with a clear *focus* throughout the observations. The interviewer must find a way to keep the conversation on topics that are useful, without entirely taking back the control from the person being observed. A *focus* gives the interviewer a framework for making sense of work. Further, it will also challenge the interviewers entering assumptions.
3.3.1 Affinity Diagrams and Work Modelling

One way of organising the material gained through Contextual Inquiry is to create an Affinity Diagram. It organises the individual notes captured during the interpretation sessions into a hierarchy revealing common issues, themes and structures. The diagram shows in one place all, to the team being studied, relevant issues, worries, and key elements of work practice. The groups emerge through induction; one note is put up first and then other notes that are related in some way are placed beneath in the same group. By reading the affinity the designer learns key issues in the same time as they can see the exact data that contributed to identifying each issue in the work. In this way important structures about the work or the system that has been studied can be identified, since all the data relevant to a theme is shown together, creating stories relevant to the design problem. (Beyer & Holtzblatt, 1998)

Another way that Beyer & Holtzblatt (1998) suggest in order to get perspective on what has been observed is to create work models. They suggest five different models: the workflow model which describes the people involved in the work and the communication and co-ordination among them. The sequence model shows the detailed work steps necessary for the people to perform, in order to achieve a goal. The artifact model, which represents the physical things, created to do the work and the cultural model, which represents constraints caused by the organisational culture. Finally they propose a physical model to show the physical structure of work. The models help to better understand the person being observed; what his intents are and which strategies he uses to achieve that intent, which structures he use to support the strategy, and what concepts he use to think about work.
4 The design team study

The following chapter presents how the study was conducted. The design team of the study is presented: the roles that were active, the methods that lead their work and the objects that were used.

4.1 Ethnographical inspired studies

The study was carried out through a case study, where the communication between the different roles in a design team was studied. The aim was to come as close as possible to the teams’ daily work, in order to form an understanding of how work was carried out and where problems arose. The boundary objects that were used within the design team were the starting point for doing this. In order to get as rich data as possible, several kinds of data collection techniques were used. Apart from conducting Contextual Inquiry, with contextual interviews following the apprentice model, deep interviews with each team member were held. Data was collected throughout a longer period of time (three months). Throughout the whole project a continuous dialogue was held with the team members, in order to clarify results as they were gathered and thereby making sure that no misunderstandings had occurred. The study was carried out from end of January to May and focused both on the study of each role’s individual work and the work that was performed together as a team. The observations were made using a field-based methodology.

4.1.1 The Design Team

The design team consisted of 9 members, working from six different roles:

- The project leader
- Interaction designer1
- Interaction designer2
- Art director1
- Art director2
- The copywriter
- The interface programmer
- System developer1
- System developer2

Each role was observed and interviewed at planned occasions and in that way equal amount of planned time was spent with all roles. However, since the researcher’s desk was placed next to the interface programmer’s, and opposite the two system developers, more spontaneous observations were made on the interactions that took place around these roles than the interactions that took place on the other side of the wall, where the designers were placed.

The design team worked under the organisation of an IT consultancy business, based in Scandinavia. The company was created in the 1960s and had gone through more than ten mergers and corporate acquisitions throughout the years. The company delivered IT/Internet and Telecom business-critical solutions within the service areas of Systems integration, Business Modelling, Interactive Communication and Convergence. The company as a whole had around 500 employees.

4.1.2 Observations

Four to five days a week were committed on site in order to observe both the team members individually and the team as a whole; by listening to, talking with and questioning their work. Events were documented by taking field notes, both in formal meetings and informally gatherings around a desk or somewhere else in the office, as conversations and issues arose. As soon as something of interest was uttered notes were taken. The researcher participated in all kinds of meetings, such as Monday’s production meetings or spontaneous check-up meetings throughout the design process, that either the researcher or the members of the team considered as important for the study.

The researcher had her desk next to the interface programmer, which was in the middle of the office and she was thereby placed in the centre, where most of the interaction between the roles took place. The fact that it was a small group, made it easy
to observe the object based communication, both within and across each function.

4.1.3 Interviews

After four weeks of conducting contextual interviews, in-depth interviews with each team member were held in order to gain a deeper knowledge about each role’s view on the team communication. The interviews were held in the conference room, at the design team’s office and were all recorded and transcribed. Starting with one pilot interview, deep interviews were thereafter conducted with each role that was actively represented within the team. In addition another team member, Art director2, who was currently on maternity leave as the study took place, was also interviewed.

The aim with the interviews was to gain a deeper understanding of each role’s view on more specific topics concerning the team communication; such as the boundary objects that were used and in which stages of the design process problems arose. The interviews were semi structured and consisted of 16 questions.

4.1.4 Affinity Diagram

In order to survey the material gained through the interviews and observations, and start the interpretation session, an affinity diagram was made. The material consisted of quotes from the transcriptions made of the audio-recorded interviews. Quotes were selected in the transcribed material and commented in the margin in Word’s “comment balloons”. Each role’s quotes were given an individual colour in order to make it easier to separate them from the rest of the team member’s quotes. This also made it possible to distinguish interesting patterns such as if one role tended to have a lot of opinions concerning a certain topic.

---

18 The team members were interviewed taking a stand in the role that they were most actively representing. Apart from the copywriter who mainly gave his answers from his role of a copywriter even though his main responsibility was to work as a project leader. This was because he was the only copywriter in the team and it was considered as important to get as many role’s perspective on the team communication as possible.
A few problem areas were distinguished at this point, such as the documentation of design decisions, the use of prototypes, the communication of each role’s final design work document and the implementation of new ideas concerning the way work was carried out. The material was grouped into categories that the researcher considered being of interest, such as “documentation” and “low fidelity prototypes”. At this stage material gained (field notes) from the observations made, were added to the groupings.

The groupings were further refined into the categories of different boundary objects, which formed the framework that was used throughout the rest of the analysis. The framework was based on the previously mentioned grouping of boundary objects made by Marick (2003) and Schmidt et al. (2002) and Laseau’s thoughts on graphic notation. These categories concerned objects used to guide the team, keep track of design decisions, visualise design ideas, communicate the work of one role to another, communicate towards external interests such as the client and the organisation and finally the objects used to keep the team gathered. The categories will be further introduced in chapter 6.

Figure 13 The affinity diagram made for this study
5 Results

The following is a presentation of the results from the design team study. First the design team roles, the setting they were active in and how they communicated are introduced. A description of the design team method is then given. The chapter ends with a presentation of the most commonly objects used within the team.

5.1 The design team setting

As presented in the previous chapter the roles represented in the design team were project leader, interaction designer, art director, copywriter\textsuperscript{19}, interface programmer and system developer. It was common that one person had more than one role, depending on the project. Interaction designer occasionally possessed the role of usability expert and project leader and the copywriter most of the time also worked as a project leader. The design team was located in an office of open space character. The project leaders, interaction designers and art directors were placed in the first part of the office and the programmers such as the interface programmer and the system developers were placed in the next part. The office was shared with another team working in the same organisation. Each team member had his or her own desk and computer.

Overall the design team’s prescriptions for the responsibilities of each role and the descriptions of them were similar to the theory descriptions. However, the theory emphasises that the interaction designer’s work should result in a prototype (Smith, 2001), whereas the design team’s prescription said that this was optional.

\textsuperscript{19} The copywriter is the person employed to write advertising or publicity copy
The team members of the design team all belonged to different communities of practice. (See section 2.1.1) The programmers were specialised on their work and formed one practice, the interaction designer’s formed another and so on. Each role worked from their perspective, specialised on the knowledge the work carried out in their practice meant. Hence the design team consisted of at least four different practices and together they formed a community of interest. Set in relation to the client, the team worked as a whole; towards a joint and agreed goal and in that relation/setting the design team became a community of practice that together with the client formed a community of interest.

5.2 Communication in the team

The communication within the team was usually made directly - eye to eye. Team members communicated individually with each other when needed. The team also communicated with each other more as a whole at the times when the project leader gathered the team to spontaneous meetings, for example to update the team on new demands from the client. On Mondays a more formal production meeting was held when each role informed the rest of the team of their planned work for the week and updated with information on how far they had come in a project. Some team members used chat systems such as ICQ and MSN Messenger to communicate with each other. There was also a bug handling system used, in order to keep track of bugs in the system that was developed. This bug handling system worked in the way that one role would assign another role a bug that was on that role’s responsibility to fix, in order to be able to move on further with the design. The project leader was responsible for closing the bugs when they were solved. The communication with the client was mainly made by the project leaders and the interaction designers through emailing, phone calls and meetings, which normally were held at the client’s office.

5.3 The Design Team Method

The design team method, in line with several other traditional methods for usability engineering, derived from the Rationale’s system development method RUP. RUP consists of the phases of
inception, elaboration, construction and transition. Figure 14 illustrates how the design team had adjusted the RUP model to better fit their work, by introducing more steps of usability work.

The design team prescribed their method, as starting with a *definition phase*, where a pre-study was made in order to define the target group and decide what kind of technical solution would be used. The content was defined and a project group set together. The documents produced in this phase were normally the principal documentation and personas.

During the next phase, the *design phase*, the graphical design was produced and the functionality and the interactivity decided. The content was further designed and text material produced. The system design started. The documents produced in this phase should normally be the functional sketch, the detailed sketch, text manuscript, demand specification and system documentation.

The *production phase* consisted of the interface programming, the system development and changes in the detailed sketch. No new documents should be produced at this stage. After this phase followed the *test phase*, when the system should be tested in the environments and situations specified. A user driven test should be carried out which would result in the test documentation. In the *delivery phase*, the system should be delivered; for example on a CD and then installed during the *installation phase*. Preferably this should take place in the client’s user environment. Tests should be carried out until final acceptance is reached.

![Figure 14](image)

*Figure 14* The design team’s adjusted version of RUP

---

20 The blue arrows point out where usability work was emphasised, by letting the different usability roles come into the process.
5.4 The Design Team Objects

As mentioned before, a common way to describe the working process is to use a model consisting of the steps of principal design, functional design and detailed design. The design team had adopted this model for describing and naming their main objects for transferring the design work between the different roles. Apart from the objects described below, the design team also had prescribed documents for:

- Requirement specification
- Test documentation
- Text manuscript
- Technical specification
- System design and architecture
- System documentation
- Documentation of technical environment

Most of these objects were supposed to be created by the team as a whole. However, since these objects were not as commonly used within the design team as the other documents were and since they were never produced during the time this study was carried out, they were not a part of the analysis. Personas and persona-based scenarios had been used rather frequently but were not used in any project during the study, and therefore it will naturally not take part of the analysis.

The Principal design document

The design team's way of describing the principal design was similar to description given in the theory chapter. The design team emphasised the role it had for giving a fast, comprehensive and structured picture over what the product should consist of. This is also emphasised in design theory. In addition they also emphasised the role it had for clearly describing each step in the interaction design, and in that way, it became something that the team could refer to during the project.
In contrast with what was described in the theory chapter: that the interaction designer should make the principal design in collaboration with the requirement analyst, the design team's document prescribed that it should be made in collaboration with the project leader. As presented in the theory chapter the principal design is a good tool in the communication with the client for explaining ideas at an early stage. It can also be used to justify design decisions when being further in the design process. On the contrary the design team says that it should be used in order to check with the client that the document corresponds to decisions made in the requirement specification.

**The functional design document**

According to the design team prescriptions the interaction designer's work should result in a document called the *functional sketch*, where the design framework with flow and dialogue is described. At present the functional sketch consisted of squares named with headers of the specific part of the interface that it represented. Each square had a number and in the margin you could read explaining text of what the specific element was, what happened when interacted with, and so on. The information appeared structured in a sequential order. An important difference between the theory and the design team's prescription of this document is that the theory describes a more thorough working order, which consists of many different steps before the actual specification/document is written. For example Smith (2001) suggests different models adjusted to different purposes of communication, such as a user model, a designer model and a programmer model. He also suggests the creation of mock-ups and wireframes and emphasises the importance of creating a prototype before the actual writing of the document.

Further differences are that the design team’s functional design describe functions, content and structure *page by page* as opposed to Ottersten et al's (2002) prescription who said that the functional design should not get specific; details for each screen picture for example, should be described in the detailed design. Also, differently to what Ottersten et al (2002) and Smith (2001) say, the design team’s prescriptions did not mention what could be problematic when making the functional design. Therefore it did not contain examples such as how to get around problems
by using scenarios or by referring to use cases in order to make sure that all the components' behaviours were described in the right way.

Figure 15 Example of the design team functional documentation

The detailed design document

The art director in collaboration developed the design team’s detailed design document with the interaction designer and to a certain extent also the interface programmer and the system developers. The design team had divided the graphical documents into two objects: the detailed design and the graphical documentation. Therefore many elements described in the theory’s detailed design did not appear in the design team’s version of it. Instead things, such as descriptions of fonts and colours and so on appeared in the graphical documentation. It was a guideline produced to guide the team about what colours and fonts to use, which they could refer to throughout the

21 Also in Appendix
design, in order to see what graphical decisions were made and why. That document is described more in detail below. The detailed sketch was at times referred to as “the layouts” in the design team.

The graphical documentation

In many ways the document that the design team described as the graphical documentation resembled the theory’s description of the detailed design. The design team’s graphical documentation was prescribed to contain descriptions of fonts used in the product and the size different headings should have, which in the theory was prescribed in the detailed design. The aim was to provide the team with graphical guidelines. It was prescribed to inform on whether texts should appear as pictures and whether they should be HTML, in the cases of the product being a web solution. Further it described colours (with RGB-code) and what they were used for and how the page should be disposed and what parts the layout consisted of. The document also described the thoughts behind the design and how parts of the design interacted. The prescriptions also said that it was desirable if influences were described and exemplified with for example a colour, a brochure, and advert or other influences that have been discussed with the client. The document was used throughout the graphical design and could after a certain point only be changed in collaboration with the client. The graphical profile was supposed to be made by the art director.

Styles for «Main»

<table>
<thead>
<tr>
<th>H1 Blue</th>
<th>Lorem ipsum dolor</th>
<th>Lorem ipsum dolor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H1</strong>&lt;br&gt;font-family: Arial, Helvetica, sans-serif;&lt;br&gt;font-size: 24px;&lt;br&gt;color: #111111;</td>
<td><code>&lt;p&gt;Lorem ipsum dolor&lt;/p&gt;</code></td>
<td><code>&lt;p&gt;Lorem ipsum dolor&lt;/p&gt;</code></td>
</tr>
<tr>
<td><strong>H1</strong>&lt;br&gt;color: #111111;</td>
<td><code>&lt;h1&gt;Example of the design team graphical documentation&lt;/h1&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

Prototypes

Also in Appendix
The low fidelity prototypes that were used within the design team were most of the time sketches of the product with loose pieces of paper stuck on top, that could be moved around in order to illustrate interactional and functional aspects of the product. The interaction designers who used them to try out design ideas on both the client and the rest of the design team produced these. It depended on the project and the client, whether it was produced or not. There were no specific prescriptions available for the prototypes, as there were for the other objects produced. Instead each interaction designer produced them according to the methods that they were familiar with. The high fidelity prototypes were either simple programmed digital ones that the art director produced in order to illustrate a visual effect of his graphical work. They could also be more advanced and in those cases they were produced by the interface programmer in collaboration with the interaction designer and the art director. They were created when projects were divided in two phases. In the first phase a programmed prototype was then used and in the second phase work was centred on making that prototype concrete and to develop a product design from it. Whether the projects were structured this way or not depended on the agreements with the client.

**Technical guidelines**

At times technical guidelines for the product were made in the shape of templates that the programmers had decided on. The art director was the one who mostly used the templates. The aim was to make prescriptions for what graphical elements the different pages/steps in the product should contain. By prescribing a number of templates for different layouts, the programmers hoped to reduce their workload, since if these did not control the art director, he could produce layout suggestions that would take far too much time to produce. By deciding a limit for how many different kinds of pages should be produced, this was prevented.
6 Analysis

The following chapter presents the analysis of the results from the observations and interviews. The data, such as field notes and transcriptions were analysed through the affinity diagramming technique, where themes and structures were noted. The results were then further analysed through a framework consisting of six categories of objects, based on crystallised patterns formed throughout the design team observations, Marick’s (2003), and Schmidt et al.’s (2002) categorisation of grouping boundary objects and Laseau’s (1989) theories of graphical notation. As presented before (see section 2.4.2) Marick (2003) proposes that boundary objects could have the following purposes: serve as a common point of reference, guide the team through co-ordination and alignment, have a function of translation, and serve as working agreements. Schmidt et al (2002) argue that the objects have the function of satisfying different concerns simultaneously and are a good way of visualising ideas. Laseau (1989) argues that document objects such as sketches serve an important role for transferring design ideas between roles, since the design team members can share ideas through sketches and other visualising objects. It was believed that the framework would be a good way to learn the key problems connected with each category of boundary object and thereby get deeper understandings of the problems that can arise when using them.

The six categories made in the analysis have been given the names external-, guiding-, visualising-, role-to-role-, tracing- and backbone objects. The chapter starts with a presentation of the results and analysis, category by category. Each object section ends with brief concluding section. A work model was made according to Beyer & Holtzblatt’s (1998) models for better understanding the interviewed person’s working environment. This was made as a complementary part of the framework-analysis and is presented after the analysis of the six categories. The chapter is ended with a concluding discussion.
6.1 External Objects

When the design is communicated outside the team to external interests such as the clients and organisation, objects are an important communication aid. They are often design documents and prototypes produced in phases of the functional and the detailed design. Studying the objects within this area of communication can give important information about the interaction with external environment. Establishing a good relation with the external interests can be crucial for how the design work develops, especially with the clients, since that will help work develop smoother and prevent that misunderstandings occur. Using the appropriate boundary object to communicate the design work will help establishing this relation. From now on these objects will be referred to as external objects.

In the design team the objects used in the communication with the client were mainly the principal sketch, the functional sketch and the detailed sketch. When necessary, objects such as prototypes and personas were also used to illustrate functional ideas, graphical qualities of the layout or to narrow down the target group. However, there were no objects specifically adjusted for communicating with the rest of the organisation, such as the management, who did not participate in the hands on work, when designing and developing the products.

6.1.1 Communication with the client

The following sequences present the problems revealed in the communication with the client:

*Communication failed in complex problems*

The major problem that was found within this area of communication was that the client often did not have the exact demands for the system presentable at the beginning of the project. This made it hard for the design team to decide which steps of the process should be included. The demands from the client changed in time, which resulted in that valuable time was lost and the team might have to go backwards in the design process in order to change or alter new ideas; that would fit the
new conditions. This occurred especially in technically complicated projects. Complex problems means a larger problem space, often due to the many technical aspects involved, which makes it harder to get a good overview of the problems that a solution might imply. This phenomena is a well-known problem and in order to get to grips with it Houde and Hill (1997) argues, as mentioned earlier, that designers must be prepared to throw some prototypes away and to use different tools for different kinds of prototypes. Designers must be prepared of that broader audiences might require higher resolution representations.

It is therefore crucial that the design team is well prepared when these situations occur. It is for example necessary to have knowledge of different prototyping techniques in order to be able to choose from a plethora of objects, instead of focusing on the communication through one. Examples of objects that can be applied are prototypes, for clarifying the implications of different solutions and technical specifications, for providing a better overview over the technical demands. The design team method should be adjustable enough to fit different kinds of projects, so that when working in technically complex projects more time could be committed to the analysing phase of the design process. This is in line with Badke et al.’s (2002) arguments about adjusting the method to better fit the design team’s real working situation.

Dividing the work into two phases is one solution of how work could be conducted more carefully and thoroughly in complex projects. A programmed prototype could be built in the first phase before working on the concrete solution. This would support the team and would prevent that too poor solutions are adopted, since the prototype would help the team members visualise solution ideas and hopefully show on problems that might arise. It would also work as a good communication object with the client, since using a programmed prototype would make it easier to explain and visualise complex ideas.

A divide in perspectives

The client thinks of us as if we were a printing house -all they want is the ready-made product. They don’t see us as creative, thinking human beings maybe…! Art director2

(Excerpt from field notes, author’s translation)
Another problem that made it hard for the team to communicate their ideas to the client was the divide in perspectives between them. Since the client was focused on the ready made product there was little understanding for that interaction design and graphic design are both part of the production process. It was not fully understood that their work such as conducting user tests and committing time for the preliminary studies and the conceptual part of the design process must be included in the projects. Therefore it was often hard for the design team to convince the client that certain steps should be carried out - and paid for - during the production process.

A suggestion is to make the design team’s work more explicit. The design team should aim for building up a common ground with the client, something, which could be done by using more efficient objects. The communication would be improved if using other objects than only the functional design document. The interaction designers could, for example, use more low fidelity prototypes: such as writing scenarios and storyboards with explaining sequences of how their ideas would be used in the product and what this would mean. The functional sketch could also be accomplished by the graphical layouts and set together in a simpler high fidelity prototype. This way the client would get more involved in the designer’s work and thereby hopefully get a better understanding of the work they conduct. This is in line with Smith (2001), who suggested that, the graphical design work and the interaction design work could set together in one document during the functional design. In order to be able to accomplish this, the design team must allow time to try out a multiplicity of prototypes.

Low fidelity prototypes were found insufficient

There are so many things that you can not do with a paper prototype, really. Sometimes there is a need for different kinds of prototypes. What I mean is that it works well in the first step but after that I would have wanted a programmed prototype and then to make a test again; and from that do the adjusting. Interaction designer2 (Excerpt from field notes, author’s translation)

Occasionally low fidelity prototypes were produced by interaction designer1 in order to visualise a design idea to the client. However, as expressed above, they were not always experienced as being a sufficient communication object for this purpose. Interaction designer2 found that the material of the low
fidelity prototypes she produced could not visualise the sequences and flows of the system in a satisfying way. For example it was difficult to visualise the results of user actions such as that a drop down menu would unfold when pressed a certain option. As argued by Houde & Hill (1997) some prototypes, such as interactive storyboards, may work well for a design team but not for members of the supporting organisation. In order to nourish the communication with the client it is important that design ideas are communicated through the appropriate object. The design team needed support in their choice of object and therefore a model, similar to the one presented by Houde & Hill (1997), section 2.4.1, could be used when deciding the prototype focus. In technically advanced projects there might for example be a need to focus more on the “implementation”, and “role” instead of “look and feel”. A model would encourage the team to reflect upon why they are producing the specific prototype, something which, at present, was not done to a great extent.

It is also important that the objects are communicated to the right persons. As mentioned in the theory chapter, it is a common that the client is confused with the user, as if they were the same people. The end users were not always part of their projects and in order to make sure that the user perspective is always represented, the design team could do as suggested by Ottersten et al. (2002). They suggested finding a receiver among the end users of the product, with the responsibility and authority to ensure the quality of the product. If having the possibility to always communicate the objects with them, as well as the client in general, that would probably lead to more meaningful results. However, if communicating with “power users” of this kind, never guarantees that the data will represent the real user group. This is a problem but in many cases it is not practically or economically feasible to have that communication with the users at all. Therefore, if choosing between having a “power user” or no communication with the users at all, “power users” must be to prefer.

6.1.2 Objects used with the organisation

The following sequences present the problems revealed in the communication with the rest of the organisation:


Lack of object for gathering team and organisation

Sometimes I feel that those kinds of things should be a part of our working process – whether the client pays for it or not. How that could be solved economically though I’m not sure… You constantly get back to the time aspect! Art director2 (Excerpt from field notes, author’s translation)

Through interviews it was revealed that the team members did not always agree on where focus should be in individual projects. Some team member’s wanted to focus more on the preliminary work and wished for more time for testing their ideas on users outside the team, or having the opportunity to update their documents, such as the functional sketch, a second time. On the other hand there were team members who were more focused on the demands from the surrounding organisation. For example that the time aspect in the projects must be prioritised and that these things cost money. This resulted in that some team members experienced a lack of focus in their work concerning what they were actually working for.

It is possible that the team would find more acceptances for changing their working process and including the steps mentioned above and also finding solutions to how to solve the time aspect problem, if given more opportunities to communicate with the rest of the surrounding organisation. Involving the organisation around the team, such as the management, on a formal meeting for each individual project, could be a good way to make work visible and accepted within the organisation as a whole and not just within the team. If implementing a meeting, interesting parts of the design teamwork could be presented by using a system such as the “post-hoc worknotes” -system developed by Löwgren et al. (2002). The benefit would be that no new object, such as a document describing their work, would have to be made. Instead they could illustrate their work, in a simple and fun way, by showing the video cuts recorded when carrying out their design work.

6.1.3 Concluding analysis

The communication failed in complex problems. There was a divide in perspectives between the external interests and the design team. There was a need to find the appropriate communication object and not just focus on the use of one. There was a lack of object for supporting the communication
between the team and the organisation. A suggestion is therefore a video recording system, similar to the one developed by Löwgren et al. (2002).

6.2 Guiding objects

Another area that needs to be investigated to understand how to improve the work of design teams are the objects used to guide them in their work, because they have the function of giving direction on how to work, what to work for and why. The objects in this category can be: guidelines/templates for design specifications or documents describing the methods to be used or the roles involved and their responsibilities. As mentioned earlier, graphical notation can help individual roles in a design team to direct their work in their respective areas of specialisation (Laseau, 1989). From now these objects will be referred to as guiding objects.

In the design team the guiding objects for many projects were mainly the prescriptions available for methods, processes and roles. These were kept on the shared server where the team members could find them if needed. The guiding objects for individual projects were guidelines or templates produced by the art directors and the programmers in order to set up rules for the graphical/technical limitations of the product being produced. By using the term guiding objects I intend to cover the objects used to guide the team generally throughout the working process and many projects; such as prescriptive documents for design methods and processes, definitions of roles and responsibilities. I also intend to cover the objects used to support the team specifically in individual design projects; for example objects such as graphical guidelines, which were reproduced in each individual project.

6.2.1 Guiding objects used for many projects

The following sequences present the problems found in the design team associated with the objects used for many projects:

Documents lost and inactive
A fundamental problem concerning this category was that many of the documents, such as the ones defining the roles and their responsibilities were never used and it was found that to some extent the team members did not even know that they existed. There were numerous documents kept on the shared server but it was hard to get a good overview of them, since they throughout the years had grown into a complex structure kept in different files and folders. This made it hard, especially for new team members, to keep track on where documents were kept and to know that the information was there. Hence the documents were, to a great extent, inactive, which meant that valuable information that could have been used for guiding the design team was lost.

This was due to the fact that the company had gone through continuous structural changes over the last few years and thereby more and more documents had been added to the shared server, where they were not organised or not in the right place. Another reason that can explain why the documents were not used has to do with the size of the design team. When working in smaller teams guiding documents are not as crucial for the success of teamwork as they are in larger project groups. A smaller group is easier to guide and keep track of than a bigger group. Therefore too many guiding documents and formal parts of the process will feel like overkill to the team members. The design team should preferably reduce the amount of guiding documents or join them into a few more simple and straightforward documents. That would make the documents easier to access and digest. A few documents with rich content would be preferable to many, with a more vague content.

The fact that the prescribed guidelines were not followed when set into practice, is not a surprise. It confirms what has earlier been pointed out by Badke et al (2002), and others; that many of the guidelines were never designed with respect to the real working situations and therefore they often fail. Another reason is that the formalised, linear way of working was originally designed with bigger projects in mind. (Löwgren et al, 1998) For a small team, such as the design team of this study, the guidelines could allow a more flexible working order, since small teams could handle this better than bigger ones. That would suit the team better and the benefit would be that the different roles could use each other as resources by getting feedback and developing the work more as a team throughout
the whole design process. This was done to some extent but
should be encouraged more. A suggestion is to modify the
design team method and base it on a model such as the Star
Lifecycle model (Preece et al. 2002), which is adjusted after
designers actual way of performing work and allows a more
flexible structure of the working order.

![Star Lifecycle Model](figure17.png)

Figure 17 The star lifecycle model

6.2.2 Guiding Objects for individual projects

The following sequences present the problems found in the
design team associated with the objects used for many projects:

**Technical/graphical constraints were not clarified**

I discussed a lot with the interface programmer before, about having a
technical/graphical document that you look at before starting up a project.
That would make my job easier as well. Otherwise I might make up a new
size for the text for example. –Everything that could make our work easier is
good! Art director2 (Excerpt from field notes, author’s translation)

It was revealed from the interviews and the observations that the
functional, graphical and technical constraints for individual
projects were not always clarified. This could be explained by
the fact that the prescribed graphical documentation was
normally not produced. Instead details about the graphical
documentation were to find in the detailed sketch, however, the
information was in general not very developed. The functional
sketch and the detailed sketch were produced in all projects but
these were not sufficient for giving a clear guidance on these

---

23 Previously presented in section 2.2.1
aspects of the design that was produced. The art directors thought that more guidelines would help them adjust to the technical devices and constraints proposed by the programmers. Therefore, this could be done with an over-all-embracing design document, informing about graphical and technical aspects of a specific project. This document could contain information about for example which fonts to use, which colour it should have, what size etc. It could also describe the general graphical expression the product should give. Technical aspects to inform on could be for example if templates should be used and what they should consist of, in order to ease the work of the developers and avoiding unnecessary programming to be produced.

There must be a reason why the graphical documentation was not produced. Probably, the design team found it sufficient to work with one document and started to make the graphical documentation in the detailed document instead. And the result was obviously not satisfying. However, if the document was produced in co-operation between the art director and the interface programmer, and the technical and graphical constraints were set in relation to one another, the document would probably make more sense. A suggestion is therefore to have a cross-role design document where each role has stated their preferences and suggestions for a specific project. This would direct work and support the team when struggling with dilemmas about the design. If this guiding document was put together by the team members themselves, according to each role’s needs and opinions, they would probably be more motivated to work from them. Art Director 2 was one of them who thought that this would be useful:

*And then people can not come and say that I want this because it looks good—instead it will be balanced with “it looks like this and that’s why we need a model like this…It is a pain getting rules thrown at me when I’ve sat down to start working on the layout. But then there could be a graphical suggestion at this (early) stage already. But that presupposes that we can work more together at the same time. Maybe sit next to each other somehow. Art director2 (Excerpt from field notes, author’s translation)*

By creating this kind of document there is also a greater chance that each role’s voice is heard and it can be a way of letting all team members influence what direction the design should take. One thing to bear in mind though is it might have negative implications on the design process if being too formal and dictating. If too many “rules” are set up at the beginning of the project, there is a risk that that also will affect the solutions, since
they will be controlled by these formal frames. Therefore it is important to emphasise that a guiding object like this is kept on an abstract level.

6.2.3 Concluding analysis

It can be concluded that documents often were lost and inactive. Also the documents were not relevant enough. A suggestion is therefore to gather guiding principles concerning the roles, methods, processes and objects in few more direct, concise and easily accessible documents. A joint graphical/technical document should be made.

6.3 Visualising objects

Some objects have the function of carrying the design process forward within the team; altering new ideas. As Schmidt et al. (2002) argue they become the fundamental means for making the “not-yet existing” and “in-the process- of –becoming” field work and they are therefore crucial for the creative design process. As discussed earlier the low fidelity prototypes can be a good way to visualise a design idea to other roles. Also, they serve as a good material to gather around when using creative methods, such as brainstorming. Examples of the objects used during the conceptual stage of the design are sketches, scenarios, storyboards and moodboards. Objects used for visualising the design at a more developed stage are normally digital prototypes produced with the help of software tools. From now on these objects will be referred to as visualising objects.

The design team objects represented in this category were the low fidelity prototypes, which served as base material during brainstorming and prototyping, when decisions were still on a rather abstract level and the high fidelity prototypes, which were used in the more developed stages of the design.

Prototypes associated with prestige

System developer1 about the use of low fidelity prototypes:
Most team members agreed on that the low fidelity prototypes were a good means to explain and visualise a design idea. Interaction designer2 found the paper prototypes very useful since it engaged the programmers at an early stage of the design process, which gave the programmers the opportunity to give their opinions before the interaction designer had carried an idea too far. However, they were not produced in all projects since which boundary objects being used depended on each role’s own preference. One of the interaction designers found that his sketching was not good or “professional enough”, for showing the client or to use within the team, and therefore he rarely produced any low fidelity prototypes, neither to use in the interaction with the client or within the team. To some extent, it therefore appeared to be a question of prestige whether low fidelity prototypes were produced or not. This is unfortunate since sketches and low fidelity prototypes can serve as a good base material in the first conceptual stages of the design process. Especially they have an important function in bringing the different roles together in the way that it offers team members a means to gather and discuss around and as Schmidt et al. (2002) argues, it thereby becomes more explicit what the intention was with a chosen design idea.

Interaction designer1 thought that it was more of a job for the art directors to produce the prototypes, since they were more trained and skilled for that. This would probably be a good idea since useful knowledge from both roles could be shared, which would ease the production of prototypes such as sketches, and storyboards. However, just as Preece et al. (2002) suggest, prototypes created in the conceptual stage are never meant to replace the real product, and therefore they should be encouraged within the team and without too much emphasis on their appearance. Introducing the prototyping concept further, within the team, could do this. In situations when the interaction designers feel insecure, producing them together with the art director, as suggested by Smith (2001) could be a good idea.

Prototyping for the right purposes
"It feels like building a house starting with the shell and then building the floors and stuff on the inside." Art director1 (Excerpt from field notes, author's translation)

Also the high-fidelity prototypes were highly valued by most team members but just as in the case with the low fidelity prototypes there were some different opinions about them within the team. Not everyone in the team was convinced about the actual use of them:

"It takes time since it is often something rather complex that you want to show and then when we are going to use it in the real (design) solution, something has changed and then I have to abandon all the code I made to start all over again. All time is wasted -you do not have any use for it afterwards." The Interface programmer (Excerpt from field notes, author's translation)

The problems appeared when more advanced prototypes were built, since the programmers in general were sceptic as to what the actual use of these more complex, high fidelity prototypes was. For example, the interface programmer often found that he could not reuse his code for the prototypes, when moving further on in the project, since changes that were made forced him to start all over again. Therefore he experienced them as a waist of time. This seems to be a classic example of professional vision, (Goodwin, 1994) where the interface programmer sees the prototyping issue from his professional perspective. He did not see the value of producing the programmed prototypes since they do not contribute with any direct value or result to his work. Viewed from his professional, i.e. technical, point of view, the prototypes do not have as much value as they have for the designers, who see them as a good base material for testing their interactional and graphical work on end users and team members. But the value that these prototypes might have for developing the design and the value they might have for the communication both externally and internally must not be underestimated.

It is possible that the problem is that the roles relate to objects in different ways. Visualising objects, such as prototypes and sketches, (for now referred to as type1 objects) normally require a meeting in order to be transferred. They have the character of encouraging discussions and analysis of the design among design team members and external interests. On the contrary design documents such as guiding objects and tracing objects (for now referred to as type2 objects) are normally written in a document and do not involve the interaction of others. They are produced in order to guide the team in their work or to
document important design decisions down, in order to keep track of them both during the design process and afterwards, when a project has ended. These objects are more static than the other objects but will serve their purpose for a longer period of time, as opposed to prototypes and objects that people meet around at some stage of the design. Prototypes have a more instant value and once they are produced and used, they are put aside and in general not used again. As Smith (2001) expresses it: “the prototype should be created with the thought of being able to throw it away at a moment’s notice. It is just there for you to have something to put in your hands, so to speak.” The interface programmer seem to refer to the prototype as a type2 object more than a type1 object which caused the confusion when using them.

I am positive about the use of prototypes if you have arguments for how it can improve (the work) for the team and the client. But there must be arguments and it has to be economically practicable. We have to think of quality at the right price! The Project leader (Excerpt from field notes, author’s translation)

As you can see, whether the prototypes should be produced or not, was also an economical matter and they could only be produced if there was a good motivation to why. Indeed, there is also a need to really justify especially why high fidelity prototypes should be produced since several problems with them have been pointed out. As mentioned in section 2.4.1, some negative implications are that they take long to build, and just one bug can bring the use of it to halt. (Rettig, 1994) In order to be able to give fair arguments why a prototype should be produced or not the team needs to apply a model, similar to the one suggested by Houde and Hill (1997). The team would then be more careful in their choice of prototype: it would force them to reflect upon what the focus of the prototype should be and what the team members should use them for. Over all, committing time and effort in the process of applying the model: analysing how the prototypes should be used, and for what purpose will save time and give better results. The project leaders would be more willing to accept the use of them if having good arguments for its purpose and the programmers would be more committed and ascertained of the use of their extra work they have to put into them. That way the model could be a way of bridging the professional vision gap. Another way to come over the professional vision could be, as suggested in the previous section: to educate the team more about how to use them, what the variety of prototypes that can be used are.
That would introduce all team members to the benefits of prototypes.

\subsection*{6.3.1 Concluding analysis}

The use of prototypes was associated with prestige. There is a need of more “education”; making the team better aware of the objects’ use. There is a need of applying a model or find some other way to better motivate their existence and guide the team to use the prototypes more efficiently. A suggestion is to use a model for prototyping for the right purpose.

\section*{6.4 Role- to -Role Objects}

These objects deliver the final work of one role to the next role/roles in the “working chain” and serve the base material to communicate from, throughout the design work. These objects can for example be the functional design specification, the detailed design specifications and prototypes. They will from now on be referred to as \textit{role-to-role objects}.

The design team role-to-role objects were \textit{the functional sketch} and \textit{the detailed sketch}. The work of the interaction designer resulted in the functional sketch and was handed over to all roles. However, it was mainly used and followed by the art directors, when creating the detailed sketch and the system developers, when building up the databases. The art directors handed over the detailed sketch to the interface programmer and the system developers. The interface programmer used the detailed documentation as a base material throughout his work with creating the HTML-code. The HTML-code, the functional sketch and the detailed sketch all served as the base material to work from for the system developers. The following is the analysis concerning this category of objects:
6.4.1 The functional design documentation

In the following text the problems found in the study associated with the functional documentation in the design team is analysed:

**The document not adjusted to work within the team**

“The functional sketch is 85% accomplished when used with the client but when used (within the team) ... it fails rather fast!” Interaction designer2
(Excerpt from field notes, author's translation)

![Figure 18 A template for the design team’s functional design document](image)

Throughout the interviews it was revealed that the functional sketch, as the design team called their functional design document, did not serve as a good object for fully communicating the interaction designer’s work to the rest of the team members. The problem was that the same document was used within the team as in the communication with external interests, such as the client. The document was mainly adjusted to serve as a communication object between the team and the client. But the needs within the team were different in some aspects to the client’s, and hence, problems arouse. The only role the object seemed to satisfy fully was the system developer. To him the sequential and linear way the document was structured in was a crucial tool and base material in their work. However, other roles were not as satisfied. This will be discussed

---

24 Previously presented in section 5.4. It can also be viewed in Appendix.
more in detail in the next sequence. For now, the point to be made is that as long as the same object is used to communicate the work of the interaction designer within the team as externally, there is a need to fit the object to work also within the team.

It is not likely that the same object can be used in two very different areas of communication, where the purpose with the communication is very different. Instead the team should consider separating the two objects or find a way to adjust the object when used within the team.

**Different roles demand different things from the objects**

*I would like to have more patches of text and more arrows into a layout. So that it (the layout) is in focus, kind of… When I see it, I don't want to be forced to look through a lot of pages; I want it to be there, straight away. But I know that Interaction designer1 thinks I am funny, because I am really bad at reading, you see - I see everything in pictures! Art director2* (Excerpt from field notes, author's translation)

It was a general opinion within the team that something was missing in the functional sketch and many of the team members had ideas on how the document could be changed. These ideas concerned both what the document should consist of and how that content should be illustrated. Art director2 found the information in the documents too much gathered into long texts. In the document used at present, the explaining pieces of text were kept separately from the illustrations of the elements, which forced the reader to first look at the illustrations and then search for the numbers they referred to. Art director2 would have preferred the information to be presented in a variation of shorter pieces of text, more explaining illustrations and arrows showing the connection between them. In that way the document would be *more direct* in its appearance; by giving an immediate comprehension when looking at it; without having to search for text and elements; going back and through many different pages. This is important since the art director and the programmers continuously need to refer to the document in their work, throughout the whole production process. In figure 19 the top image shows the functional sketch as structured in the design team at present. The one below is art director2’s
suggestion of an alternative structure. Today the document consisted of a lot of sequential text with numbers referring to different parts of the design. Art director2 preferred more arrows and shorter patches of text.

Figure 19 Two ways of structuring the functional design document

In order to satisfy the art director with more directness and explanations on the interactive information, a solution would be to use an additional object that concentrates on giving a first quick impression of the flow and interaction and how to interact with the product. This object could be either a simplified version of the functional sketch or a totally different object such as a sketch or a simple high fidelity prototype: for example mock-ups set together in an interaction. The interaction designer could construct the prototype, perhaps as Smith (2001) suggested, in collaboration with the art director, who could contribute with some of the preliminary layouts. This prototype would offer an object that is one step closer to the interface programmer’s work, since his work is based on the work of the art director, and thereby the prototype would encourage the involvement of more roles to take part of the functional design. This object could be simpler than the functional sketch and
would satisfy all roles that felt in a need of having the interactional and functional aspects clarified and illustrated in a more direct and interactive way.

*It is as if the functional sketch just describes the surface and then I try to define what is behind that surface.* Art director1 (Excerpt from field notes, author's translation)

As mentioned previously, the document did not give enough information about the technical implications of the different actions that a user could take. Art director1 believed he would have been helped from getting more information on what actually happened “behind” the interface, in the code, when an action was made. He requested a more comprehensive presentation of the system as a whole, in order to better understand how he should build up his layouts and continue his work from the sketch. The interface programmer also confirmed this. He found that the functional sketch was not worthwhile to read since it was too long and not enough connected to the detailed sketch. His work was not as directly affected by the interaction designer’s work as the art directors’ or the system developers’ work was, and therefore he had other demands on the object. Perhaps he could have been helped if a model, such as the programmer’s model suggested by Smith (2001) had been created. He proposed that the interaction designer should have a meeting with the programmer in order to get feedback on his ideas and from that a programmer’s model could be built. Always making these models (user-, designers- and programmers-), as a part of the functional design would be a good idea, since it increases the chances of that the object will be successfully used by all team members.

*There is nothing worse I think than…trying to describe a system in a linear way, which is not linear at all. We should be able to make diagrams and maybe even prototypes.* Art director1 (Excerpt from field notes, author’s translation)

The interaction designer’s work is not suitable to just be described in the functional sketch, since it only offers a linear way of presenting the material. There is therefore a need to open up for more ways of communicating the functional design, something which will be further discussed in chapter 7.

**Deficiency in the transferring of the object**

*…we almost never speak until we really have to.* Interaction designer1
Throughout the observations it appeared that the functional sketch was not satisfactorily transferred to the rest of the team members. One problem was that there was no formal meeting planned for this during the design team’s working process. Therefore it was not clear to what extent other team members actually took part of the transferred object or not. According to interaction designer1 the team members did not communicate with each other in the design work until they really had to. Therefore a formal meeting could be needed in order to make sure that work is really transferred sufficiently.

**A need to make the object updateable**

_Art director2 thought there should be room for changing the concept of the functional design later on in the process, in the stage when the art directors were developing the design. The problem she experienced was that, at the stage when the interaction designer produced his work, he did not yet know what the final appearance of the graphical work would be. It could therefore be hard for the interaction designer to imagine what the final solution would look like at the stage where no layout suggestions yet were made. Art director 1 expressed it as “they don’t have enough imagination to do this” Art director,_.

One way of avoiding this problem would be to make it possible to update the work of the interaction designer as the work of the art director develops.

A problem connected to this was that if the art director made changes, it could result in that the design decisions made earlier by the interaction designer would be affected. When the original ideas about interaction and flow were replaced by new ideas from the art director, the functional documentation was not _updated_ with these changes and there were no comments concerning interaction and flow in the detailed documentation. Therefore the interaction designer’s material became _inactivated_. Therefore it seems necessary to involve the interaction designer in the design process again, in order to make sure that important
information is not lost. That way the art director would also get valuable feedback from the interaction designer that he, from his perspective, agreed on the changes made. Hence, more interactions between the roles during this part of the design work are suggested.

**A need for cross-role work**

*I feel that I have ideas there as well; I think I have quite a logic sense of thinking. I've worked for a long time and I have a lot of experience. So then I think I can question the work of the interaction designer -I think I have the right to do that- just as well as they can question my design –that dialogue has to exist! Art director2* (Excerpt from field notes, author's translation)

It was revealed that one of the art directors wanted to interact more with the interaction designers when producing the functional design document and the detailed design document but there was some confusion in the team to what extent this should actually occur. Art director2 found that she had valuable feedback to give and she therefore required that there should be a more open dialogue between the two roles in the making of each other's objects but when this dialogue was held, both positive and negative implications arouse:

*Art director2 works a bit like an interaction designer and thinks more in that way than art director1 does. We end up having long discussions about things that I actually could have decided upon myself. In the same time it can be very profitable, since it means that we revise a solution. Interaction designer2* (Excerpt from field notes, author's translation)

Working in “overlaps” between roles is probably profitable to some extent but it must never be forgotten that it is only the “real” interaction designer that is the expert within his area, since that is where he has his specialised knowledge. However, it should be possible crossing borders, in order to see something from someone else’s point of view and in that way be able to give useful feedback. If the roles would work in a more flexible and interactive way, there would be more opportunities to share and spread knowledge. Another benefit would be in the case when someone leaves the team. It will then be easier to introduce the new person replacing the one who left, to the work of that specific role, since the other team members will be familiar with that role’s work. This does not mean, however, that all roles will get specialised on the same things and occupy the same amount of knowledge. Rather it would mean that the team
members get more chances to interact and explain their thoughts and ideas and thereby building up the common ground that is so important when working in cross discipline teams.

6.4.2 The detailed design documentation

In the following text the problems found in the study associated with the detailed documentation in the design team is analysed:

Content confusion

I've worked like that sometimes (and commented my design) but it feels as if they are not looking at it! Art director2

Sometimes I have a feeling that I document things that are never being read. I imagine it is because of a lack of time; instead of looking carefully you think that you know how something works. I can't say that it happens all the time but I'm sure it happens - just as there might be things in the functional documentation that are there, without me reading them. Art director2

(Excerpts from field notes, author's translation)

The design team had prescriptions for two documents to be used by the art director: the detailed design and the graphical documentation. However, the graphical documentation was in most cases not produced and instead details about the graphical design were to some extent documented in the detailed sketch, which was the art director’s main object in his work. A problem though was that it was not clear to what extent details should be documented and how detailed descriptions of the graphical work should be given. Art director2 used to write comments on specific elements in her layout in order to make sure that the programmers would understand the different steps of the detailed sketch. It could be information such as how many pixels a picture should be or information about the graphical effects (as mentioned before, the prescribed graphical documentation was not used). However, she was not sure whether the work she put into it was really worth it, since it seemed that the programmers did not read the notes. According to the interface programmer this was because the information was not always relevant to his work. It therefore seems that there is a need of a new template for what the object should actually consist of and how that content should be structured. Since it was mostly the interface programmer who was dependent on the work of the art director
this should preferably be worked out in collaboration with him. During discussions the interface programmer gave some suggestions of what this information could be, in order to make the ultimate detailed documentation:

1. The graphical elements should be *well structured*; with the different layers made in Photoshop in named folders; so that it is easy to find the information the interface programmer wants.

2. Information about *visual effects* should preferably be described in the document; such as for example if a link should change colour when selected and so on.

3. It should be possible to *activate* and *inactivate* the different layers in Photoshop; in order to explore the consequences of a “mouse over” for example.

4. *All possible elements* that can appear on the page should be *presented* in the document.

This could be a beginning of a general guideline for what the detailed sketch should consist of. Regarding the structure of the document it was suggested that explaining notes about the interactional and functional aspects could be put in the detailed documentation, as long as the notes contained relevant information. Using the post-it-note function in Photoshop could make these notes. Those way post-it notes with explaining comments in the layout could be used in order to elucidate interactional and graphical aspects on the design.

*Figure 20* Example of a well-documented graphical design

---

25 Previously presented in section 5.4. Also in Appendix.
Template confusion

...and then the art director can continue following those modules and those pages that are created for the rest of the website. Because then you know that you are working on a good basis. It's something we never have tried but maybe we should! The interface programmer gives a comment on further development of templates for individual projects. (Excerpt from field notes, author's translation.)

Throughout the interviews it was found that attempts had been made, both apart from the interface programmer and the art directors, to adjust their work to each other's needs. For example, the interface programmer continuously tried to keep himself updated with new ways of solving problems, which increased his chances that he would be able to realise the work of the art director without having to change it too much. Another way to co-operate was to create templates for individual projects that described how the structure of the product should be built up. If the different steps in the layout were built up in the same way; always consisting of the same elements placed in the same way, at the same place, all the way through the layout; that was the ultimate starting position for interface programmer. Because that meant that he did not have to produce so much code. If following the templates, problems such as the art director producing layouts that later turn out to be too complicated from a time perspective to code can be avoided.

However, when this had been tried out in one project, it had turned out that a template for the art directors was something different than a template for the interface programmer. Apparently, there had not been discussions on the topic in order to clarify what a template really should be. This was needed since it was obviously something rather complicated for the art directors to understand. Also it had never been documented what the templates actually were, something that probably would have helped the art directors when being confused. This is a typical situation where communication could be altered and clarified by using a communication object. The templates should be further developed into guidelines that are documented down. That way they can be referred to throughout the whole individual project. The template discussion mainly concerns the art director and the interface programmer but it would also be desirable if the interaction designer was involved in this discussion. This since his thoughts and ideas on functional and
interactional aspects might affect the decision on which templates to use.

![Figure 21 The interface programmer and the system developer working with templates.](image)

**Functional aspects disappeared**

When we introduced the conception of a detailed sketch it was with descriptions of the different elements, just as in the functional sketch. They (the art directors) don’t do that but there is a need for it. Interaction designer1 (Excerpt from field notes, author’s translation)

A problem that was revealed with the detailed sketch in the design team was that it over the years had changed from its original intended shape into something more simplified. And as already mentioned it had also become a replacement for the graphical documentation. Elements were more carefully described in the past and due to the present lack of these descriptions it did not contain enough information about interaction and functionality. This stands in contrast to what is recommended in some of the theories that built on the steps of principal, functional and detailed design. (Ottersten et al. 2002) It is also against the design teams’ own prescriptions that described the detailed documentation and the graphical documentation as two separate documents. To come to terms with the problem the design team could start working on new prescriptions for the objects used. These should, in line with what Stempfle et al (2002) suggest, take their stand in the design team’s actual working situation and be adjusted to that. The design team had gradually abandoned the idea of working with both the detailed sketch and the graphical documentation. Instead the work had
been transformed into one document. It is possible that the best thing would be to use one document instead and focusing on how the document in the best way could deliver both the graphical aspects and in the same way be a continuation of the functional design.

**A difficulty to guide each others’ work**

“I often pass by (the art director) to have a look. I try (to see if there will be any complications) but sometimes you don’t realise things until you start coding it.” The Interface programmer (Excerpt from field notes, author’s translation)

Another problem that occurred in the communication between the art director and the interface programmer was that it was hard for the interface programmer to tell the art directors at an early stage if a graphical design solution would be technically feasible or not. Often he had to start programming and get his work going before he knew if all the art director's ideas would be able to realise or not and this took time, since the code had to work in many different web browsers and different browsers worked differently. The result could be that the art director ended up doing a lot of work on layouts that in the end never could be produced and he would then have to rethink and work on alternative solutions.

One way to improve their work, which also would decrease the chances for that this happens, would be if the working order was loosened and if it was possible for the art director and the interface programmer to work more in iterations. Art director could start creating the basic ground layouts. Thereafter the interface programmer could have a go on the code; trying those graphical ideas out and then the art director could continue work again. By working more in iterations and alternating who takes control, both roles will get an opportunity to discuss and solve the problems together as they occur. There was also a need for interacting more between the roles:

“I would like us to work together more initially, with a suggestion of the functional sketch and maybe with a graphic suggestion and guidelines.” Art director2 (Excerpt from field notes, author’s translation)

One way of structuring these iterations, in order to make sure that they take place, could be to let the art director start working on the document and then the interaction designer could enter his work and they could collaborate in a phase shortly after that.
The detailed documentation could then be updated with feedback from the interaction designer. A suggestion is to have a prescribed phase for the updating of the document. This, instead of just suggesting working in iterations in general, since it is believed that the interaction will not occur if not having a formal, prescribed phase for it.

A lack of transfer

“If I don’t ask for it, it’s unusual that I get to see the detailed documentation. I don’t know why. I think they don’t need any input. But we do check at each other’s desks. That discussion is necessary but we must end. We can not offer the client alternatives—we have to come to finish.”
Interaction designer1 (Excerpt from field notes, author’s translation)

It appeared that it was mostly the interface programmer and the project leader who actually took part of the detailed sketches. However, it is probable that for obtaining satisfying solutions feedback should be given from the interaction designer on interactional and functional aspects of the graphical solution. This, just as well as feedback should be given from the interface programmer on the technical aspects. A suggestion is therefore to implement a formal transfer meeting for transferring the objects that takes place in all projects as soon as a role has finished their final work.

“A couple of days would be desirable at least. Then there will be time for the client to give feedback and the interface programmer doesn’t have to feel that he comes in as the last.” Art director2 (Excerpt from field notes, author’s translation)

Concerning the transfer of the object it was also found that there was a need for having more time for passing the final version to the interface programmer. If having a few days for actively transferring the object, the interface programmer would be more involved in the last bits of the art director’s final work and he could then give his feedback in a more straightforward way.

6.4.3 Concluding analysis

The design team focused on one object: the functional design document, to a too great extent. There was a need to either adjust the document to different communication purposes or produce other, complementary objects, to the functional sketch.
The functional sketch’s linearity does not fit the entire different role's needs. A suggestion is to use more objects, or make the object adjustable.

There was a need for more guidelines concerning the content and layout of the detailed design document. The roles needed to work more in iterations when working on this document. It is important to give time to the transfer-process of the object. It is suggested that a prescribed phase for updating the object with functional information and an active transfer of the object.

### 6.5 Tracing Objects

In order to be able to track design decision throughout a project, or afterwards when the project is finished, it is common that design teams use some kind of documentation for keeping track of what has been decided in an individual project and why. This can for example be done through Design Rationale systems, earlier discussed in the theory chapter (section 2.3.2). These objects can be documents such as the functional specification or system specification or meeting protocols and will from now on be referred to as *tracing objects*.

The tracing objects prescribed to be used within the design team were the specifications for different stages of the design. This chapter presents the results about the design team’s routines for doing this and it is discussed which tracing objects could ultimately be used in the design team.

*Design decisions were difficult to capture*

It was found through the observations that the design team did not commit much time to document design decisions. Specifications, such as principal specification, functional specification and so on, were rarely produced. Some information about decisions and so on was written down in emails or meeting protocols by the project leader but in general there were no routines for doing this in the team. Rather, it was up to each role how to organise their work and most the team members thought it was overkill to document those things. They also thought that it was difficult since the design goes through so many different stages and is constantly changing. This is also what Carlile (2002) suggested to be one of the dilemmas when
working with team design; team members know more than they can tell, since knowledge is *embedded* in the doing of an activity. It is therefore hard to put words to what has been done throughout the design process and to motivate why it was done.

In order to motivate the team members to document their ideas, and to make the process easier, the design specifications should be changed. The specifications’ weaknesses were that they were divided into too many separate documents to make it worthwhile for the team to use them in most of the projects. For bigger projects they were sometimes created but it was normally not necessary to create a separate document for each phase of the design process and it would also take too much time. These lead to that many documents were abandoned in smaller projects, even though it would have been useful to document design decisions also in those projects.

If the team instead had the possibility to document the design decisions better in the other objects that were used - such as the functional and detailed sketches - that would increase the probability for that they actually did it. There could for example be a prescribed part of the document where there was room for making notes about why something was done the way it was, when the decision was made and in collaboration with who. That way the team members would not have to feel the pressure of having to produce another document. Using some kind of documentation of decisions is necessary since it can prevent misunderstandings about what was actually decided. The problem with gathering the information in the same object, as for example in the functional sketch, is that the information is not as easily accessible there as if it was in a separate object. However, if the information was always put in the same part or parts of the document, the team members will learn where to search for the information. The benefits will still be bigger than if the documentation was not made at all.

Another suggestion is that a template for a simpler version of the specifications is made. It would probably be more accepted to use one united document for all the different phases in smaller projects, in which each role could update their motives for the decisions made during the part when they had been active in the design process. This document could be handed over from role to role, when transferring the final work to the next person in the working chain. The information is then gathered in a
A need for a complementary object

We know each other well which leads to that we just suppose and suppose and then suppose again and finally we don’t know what was actually decided. Interaction designer1 (Excerpt from field notes, author’s translation)

Since the design team of this study was a small team it was easy for team members to get hold of each other. Therefore communication throughout the design process was often made through spontaneous informal chats or meetings between two or several roles. This way of communicating suited especially the programmers, since they normally were not motivated to attend formal meetings. Throughout the observations it was found that the spontaneous communication had many benefits since it is direct and misunderstandings can be cleared up straight away. As discussed earlier, having too many formal meetings can have negative implications on the design work and for the continuous communication between the roles, direct communication should therefore be encouraged to a greater extent than the formal one.

A problem that was found though was that since communication often occurred through informal chats, it was seldom documented what was said in these conversations. It was therefore at times hard for the team members to keep track of what had actually been decided and why. Another thing that at times caused confusion was that team members in general assumed to a too great extent that they knew what was decided. This was an effect of working in a team where everybody knew each other well. It was positive in the sense that the different roles had learned about each other’s preferences about how work should be carried out. On the contrary it was negative in the sense that it also made team members believe that they knew more about what had been decided concerning a design, than they actually did:

In order to make sure that important things are remembered it can be a good idea to use some complementary object to the specifications discussed above. One way is to use some kind of design support, such as a computer based design rationale system. There is also new technique available that supports the
decision making such as digital white boards with audio recording equipment.

A problem when using these systems is how the information that is saved in the design support systems is spread so that it does not stay just between the team members that were present when the discussion took place. It is important that all information is transferred to the rest of the team in order to support the co-operation and co-ordination in the design work. A suggestion to come to terms with that problem is to implement good routines for sharing the information. The team should preferably use the formal meetings that already take place today, such as the production meetings, to present what had been saved in these systems. Another possibility is to keep the information saved in a folder on the shared server and emailing a reminder as soon as something new is updated in that folder. However, the most straightforward way would be to present it at a meeting since that would give all team members a chance to comment what had been said. A good example of a system for doing this would be Löwgren et al.’s (2002) Post-hoc worknotes system discussed before.

**Team members needed more inspiration**

At present no such systems as the ones above were in use. However, the design team had a bug tracking system, which in some aspects worked like one. The system was used to report bugs, i.e. a problem needed to be solved in order to make the product work. All team members could report a bug and ascribe it to the role that was responsible for solving it. In a way this system help the team members to keep track of the current state of the design and therefore it is a kind of a design support system. If for example the interface programmer needed input from the art director before he could continue his work, he could ascribe the art director a bug through the system. When the art director had solved it he would report that to the interface programmer and perhaps add a few comments on what he had done. According to Twidale et al. (1993), it is important that design support systems support frequent and rapid revision of the design provided. It is also important that it supports variability so that it supports different team member’s individual needs. The bug tracking system gave a good overview over what was needed to be done and who needed to do what, before the next stage in the design could be entered. Especially the programmers found it hard to keep track of all the elements of
the design that needed to be completed, if not using the system. However the system was not supporting variability to a great extent. It was mainly adjusted to fit the programmers, who were the ones using it the most and who had the most positive attitude towards it. Therefore the system seemed to satisfy the programmers’ needs to a greater extent than the designers do, mainly because the designers’ tasks are not as concrete as the programmers.

Not all team members were positive to using the bug tracking system, since in a small team the projects are small and it seems like it would take more time doing the documentation than the actual decision making. It was therefore used mainly to keep track of bugs but not so much for documenting how something was solved and why. Even though the team thought they could be better at documenting things down, since it would be helpful when going back to search for information in an old project, the general opinion was that *time* could be *used to something better*.

*Sara: So do you go back to what you documented in the bug tracking system later on?*

*The interface programmer: Yes. But if I’m gonna be able to see how I solved something I have to write that down. I guess there are reasons for doing so but one cannot bother.* The Interface programmer (Excerpt from field notes, author’s translation)

Since the team members felt little motivation for documenting their design decisions it would be interesting to find other, new ways of documenting that would inspire the team more. Work group collaboration support systems such as electronic whiteboards for documenting decisions directly; at meetings and so on, could be considered. That way the design decisions could be documented as they are being made; not having to write it down in two steps. Another example is the post-hoc worknotes system developed by Löwgren et al. (2002), which was earlier referred to. With a system like that, the team members could video and audio record meetings, write notes on important recorded sequences, and search among the material for things that they want to refer to or bring up to discussion again. The post-hoc work-note system could be a fun and inspiring way of dealing with design documentation in the same time, as it is a communication support tool in general. If the system works the way it is intended to, one of the greatest benefits will probably be that the documentation will be made automatically, without having to put much effort into it.
6.5.1 Concluding analysis

There is a need of making the documentation process easier. It is suggested that the documentation could be encouraged through a post-hoc work note system. Design decisions could be documented in the role-to role objects or they could be gathered in one single document instead of various.

6.6 Backbone Objects

- A steady base to rely upon in times of change

As argued by Laseau (1989), one of the important functions that documents for communicating within a design team can have, is to serve as a model presenting new challenges and problems to be solved. This becomes particularly important in times of change; through structural reorganisations for example. Objects can serve as a support and guidance, leading the team forward, offering a steady base to rely on. These objects can either be material objects such as documents and guidelines or prescriptions of a role within the team, with the responsibility of implementing ideas and visions. From now on these objects will
be referred to as *backbone objects*, since they have the function of being a backbone in the team's work.

The following is a presentation of the results concerning the objects carrying the function of inspiring the design team in their work and help new ideas to get established and realised. The design team was in a need of using more objects in this category since the company they belonged to have gone through several reorganisations. This had affected the conditions the team worked under in terms of number of people working in the team and economical/time constraints. These reorganisations have led to instability in the team. Therefore the team needs objects that could keep them gathered and that could offer them stability and focus in their work.

**Work was carried out repetitively**

*We do it as we did it last time – then we know we will reach home.*

*Interaction designer1* (Excerpt from field notes, author's translation)

The team had to work under a continuous time pressure and due to this and the reorganisations mentioned above their work had become repetitive. Rather than trying out alternative methods or objects in their work as a change, the team members seemed to prefer to perform work in a manner that assured them they would come to a satisfyingly result. This way of working can seem like a good idea in the short run, however, it will most likely have negative implications in the long run; such as that the solutions will lack of originality and the team will feel less inspired when performing their work.

**Good old objects had disappeared**

*Before we made –what did we call it? These big A3 paper sheets; and then we drew them like the functional sketches and we used post-it notes in different colours and attached them on; then you could move them around. But we’ve kind of lost that… to be honest with you. Interaction designer1* (Excerpt from field notes, author's translation)

*We have worked together for a long time and we take the easiest way out. Maybe that’s a reason why we stopped using prototypes etc. The copywriter* (Excerpt from field notes, author’s translation)
Another problem was that good objects that had been used before in the team, such as different kinds of prototypes were no more in use, without that anyone really knew why. For example, the interaction designers used to print out A3 versions of the functional sketch and put them up on the walls with coloured post-it notes with important user-aspects and suggestions on the design in general. These could be moved around and the interactive material also invited other team members to interact with it. This object was no more in use, which was a loss, since it had served as a good means for communication between designers and developers in the creative and collaborative stage of the design process.

The feeling of being a team was lost

I think it would be good if the projects were carried out more in teams...if the work did not follow a chain, we would cooperate more. The interface programmer (Excerpt from field notes, author's translation)

..Then we could work in teams that were only doing that and then you can create much better solutions. If we had one meeting per day maybe... The interface programmer (Excerpt from field notes, author's translation)

Further it was revealed through the interviews that the team members had lost the feeling of being a team, since work was often carried out by the roles separately, without much interaction between them as a whole. This resulted in that the team did not feel connected to each other and the less they interacted with each other the less feedback they could give on each other's work.

I like having meetings; to sit down with the little team; exchanging ideas; just ideas on functionality; how to solve things technically and still making it functional and easy to use. So we do need each other and I sometimes feel that we underestimate those occasions. We always say we don't have any time for many meetings, but being three persons spending two hours (in a meeting) -that's only six hours! Art director2 (Excerpt from field notes, author's translation)

What Art director2 says above is another example of the team members wishing for more time to discuss ideas and that more emphasise should be on those occasions. The time restrictions could be questioned when considering the discussions as an investment in the design.

A team in the need of guidance
A further result was that when the team members came up with new ideas, about for example how work could be improved, they were never developed enough to blossom. One example is that the team had tried to improve the interaction between designers and developers by organising triangular meetings between three of the roles; the interaction designer, the art director and the interface programmer. The meetings served as a good opportunity for bringing three different perspectives together when discussing design ideas. However, the three got stuck when they found it difficult to act intuitively throughout the meetings. Instead of trying to solve the problem and change the meeting or just continue working on it until they found good routines for it –because the intent with the meeting was really good- the meetings petered out to nothing and only two or three meetings were held. This is probably a result of a weak guidance and it seems that the team is in a need of someone having the role of encouraging the team members in the different directions where their work was carried out.

6.6.1 Concluding analysis

There were not enough methods or roles actively which could actively support the team in their work. The team needs more support here in form of methods objects roles. Further suggestions will be made in chapter 7.

6.7 Analysis on Working Environment

As a complement to the analysis made through the framework consisting of the categories presented above, a work model was made. It was found appropriate for this study to create a model concerning the physical environment. The model was useful in the way that it gave good overview over the physical characteristics that constrained the work of the design team. The model can be seen in Appendix.

The open office landscape
The working environment is one of the aspects that can affect the efficiency of the communication within the team. At present, the team was working in an open office landscape with only one wall that semi separated the team into two halves; leaving the project leaders and the designers on one side and the developers on the other. It was neither difficult nor far to walk over to any of the other roles' desks, but nevertheless the wall seemed to divide the team more than it actually should have to do. Being divided by a wall can probably have the unconscious effect that even though it is not far to walk over to the “other side”, there is still some kind of border there. This results in that there will be more interaction in between the people on each side, than across the border. It is probable that there would be much more interaction in between all roles if sitting opposite to each other without any borders at all. This would be fruitful for the discussions held and in the end it will affect the solutions being produced. Therefore, sitting even more close together; gathered as a whole team, would probably enhance the interaction in between the roles.

Openness breeds the team but not the individual

... It's good because it's social and you get to know each other. You get to grips with the others. Interaction designer2 (Excerpt from field notes, my translation)

There was a divide among the team members whether the open office landscape was a good working environment or not, even though most of the participants were positive to it. In general the open character of the office seemed to contribute to good communication and more interaction in between the team members.

I can't hear what G (art director1) is up to so I have to pass by all the time to see what he is doing. If I don't make sure that I can realise his ideas before he starts working on them, I might be programming on something that should not be. If he had been right across my table I could just have asked him. The interface programmer (Excerpt from field notes, my translation)

Some of the team members would even have preferred an even more open environment. The interface programmer for example, would have preferred to sit even closer to the art directors, since he had a continuous need throughout the production to exchange ideas with them. Hence it seems that gathering all roles would result in a more direct communication. But then there were also team members who got very disturbed from working in this way. One of the system developers expressed that he found it worthless. Not having the possibility to get any
privacy and peace and quiet in their work, was the main reason to way some were not in favour for the openness.

The negative about it is that you are forced to sit and listen when others are talking. You easily get disturbed. It is not as obvious (in an open office landscape) that you're concentrating. Then you have to put your headphones on and everyone is not that good with concentrating with music in their ears. Interaction designer2 (Excerpt from field notes, my translation)

As you can see each team member seem to have their own personal and professional preferences and needs. It therefore seems that the open landscape office breeds the team but not the individual.

Whether you prefer to work in open or in a more private way is personal. The best would be if there was also an alternative setting to the openness, such as corners or rooms where you could go when you needed peace and quiet. Several of the team members had suggestions for this, such as having libraries or concentration rooms that would offer an alternative when having a need for more privacy.
7 Discussion and conclusions

This chapter consists of a concluding discussion of the results and analysis made in the previous chapter. The main conclusions from each category of objects are presented. Some suggestions were already made in the result/analysis chapters. These will now be further discussed and developed. The chapter ends with a reconnection to the problem statement and an attempt to answer the research questions of the thesis. Some suggestions for future studies are also made.

7.1 External Objects

It was obvious that the design team needed further object support in their communication with the external interests. The communication problems arouse especially when working in complex projects. As suggested by Carlile (2002) there is an increased need of using objects in complex problem situations, since the problem space is larger and designers will need more support in their work. According to Houde & Hill (1997) it is important to produce a multiplicity of prototypes, especially when working on complex problems, since requirements will change in time and it is hard to choose the right prototype from the beginning. The design team therefore needs to extend their use of objects in these situations, instead of only relying on the functional design document.

Above all, the team was in a need of making more careful analysis of which objects to use. This was particularly important since the divide in perspectives between external interests and the design team often caused communication problems. It is therefore extra important to make sure that the appropriate communication object is being used. Applying a model could enhance the process of choosing the right prototype or the right object in general. Further, the communication with the rest of the organisation, such as the management could be made more efficient if using a team communication support, such as the post-hoc worknotes, developed by Löwgren et al. (2002)
7.2 Guiding objects

In order to prevent that documents simply are placed on the shared server among many other documents, where they in time are lost and become inactive, the design team should gather their guiding principles concerning the roles, methods, processes and objects in few more direct, concise and easily accessible documents. Documents must be relevant and should be adjusted to the present working setting the design team is in. There was a need for making graphical/technical constraints more explicit and therefore, for each individual project, a joint graphical/technical document should be made.

7.3 Visualising Objects

The design team did not use the prototypes as fully as they should. A problem was that some roles did not see it as a supporting object; rather it was in some cases associated with prestige. Therefore all roles in the team should gain deeper knowledge about different prototyping and visualising techniques and about the purpose of using them. It would then be easier to motivate their existence both to oneself and other team members. By using a model as suggested above in previous categories, the team would find guidance when choosing the appropriate object for the appropriate situation. Applying such model would be a way of bringing different roles together and build up a common ground about why the prototypes are used and how the team can use them more efficiently.

7.4 Role to Role objects

The functional documentation – in a new shape

As you could see in the analysis the functional sketch was only adjusted to one area of communication and therefore it needed to be adjusted to better fit the needs of the team. It is problematic when the functional documentation is primary
adjusted to the client’s needs and only secondary to the team itself. The results found indicate that the team had other needs than the client regarding how to explain, illustrate and visualise the functional design. The way the document was designed at present did not satisfy these needs, despite that it by and large followed the design theory’s and the team’s prescriptions for it. Therefore, the functional sketch might not be the best way for describing, illustrating and transferring the interaction designer’s work. The object should instead be divided into different objects that are all focused on transferring different aspects of the interaction designer’s work. Even though the functional design document’s linearity was appropriate in some directions of communication; such as when communicated to the system developers or to some extent the client, it was not the best way of representing the interaction designer’s work in other directions of communication. As you could see in the previous chapter, there were opinions on that the document lacked of directness in its appearance and that it did not describe the functions behind the “surface” (i.e. interface) enough. As a suggestion the interaction designer could produce a variety of objects, adjusted to the different needs within the team. One object could for example be a document giving a more thorough explanation on the technical aspects used behind the functionality. That document could be worked out in collaboration between the interaction designer and the programmers. Another object could be sketches or interactive prototypes that would focus on illustrating the interactional aspects in a more natural way. This is in line with Smith’s (2001) suggestions about the use of functional models before writing the final functional specification. Other objects that could be used more within this part of the design are prototypes; storyboard, scenarios, sketches for illustrating the interaction.

Another possibility is to continue to focus on the functional sketch but making it adjustable. By making it possible to hide elements any information considered as redundant could be hidden when used externally towards for example the client. That way the team could feel free to make explanatory notes about for example technical aspects in the document, without the client ever having to see them. Preferably the functional sketch should be adjusted to the needs within the group and then an adjustment towards the client would be made. By primarily making the communication work well within the group; the team could improve the design- and production process and become more efficient, something which would also
improve the relation with the client. If it is not possible to adjust the document so that it works well in both situations, different documents should be made.

*Clarifying technical implications of the detailed documentation*

The programmers work did not result in any object that was transferred to the others. This resulted in confusion among the other team members, especially the art directors, about what the technical implications actually were in individual projects. The technical restrictions were not clarified and not written down into a prescriptive guideline. As was also proposed in the category of “guiding objects” there seems to be a need for an object for better transferring these ideas to the rest of the team. That way some of the technical suggestions from the programmers can be transferred, just like the other roles’ work was, by using it as a role to role object. A template that is easily accessible and updateable for each individual project must prescribe this object. It was found that any additional objects that need to be produced must be as simple and easy to make as possible. Otherwise there is a risk that the documents will never be produced. This was especially true regarding the programmers since they were not keen on using documentation in general.

*Transfer the objects through more iteration*

One suggestion for improving the phase where the detailed sketch is developed and the interface programmer is working on implementing it, would be if the two roles could hand over their objects (i.e. the detailed sketch and the development made of it) in more frequent iterations. This can be illustrated by the “running” metaphor, where the art director and the interface programmer take turns in running a distance each in the design process. The art director starts the work with the detailed sketch, which is then taken over by the interface programmer. He tries out these first ideas in programmed code and then hands the document back - updated with feedback - to the art director. There should preferably be time for overlaps when the document is handed over, so that the two roles can interact and
give feedback on each other’s work and “run” on the same distance for a short time. Working in much iteration was also suggested by Preece et al. (2002), who says it is crucial, especially throughout what she calls the conceptual stages of the design.

This way of working could also include the interaction designer since he needs to alter his communication with the art director as the detailed sketch is being produced. It would then be the interaction designer who runs the first distance with the art director in order to hand work over to the interface programmer. And then the design could gradually be developed in this way through these iterations. Probably with less and less involvement of the interaction designer, as the design becomes more concrete.

There is also a need of focusing more on the transfer of the objects and making the role-to-role objects updateable.

### 7.5 Tracing objects

In design teams, such as the design team of this study, where there is little time or no time at all to devote to the documentation of design decisions, there is a need to find an alternative way for documenting that is less time and effort demanding. One way could be to just change the routines around the already existing objects and make it easier to document things directly in the objects used. That would mean that the documentation is not separated from other objects produced, such as the functional documentation, prototypes and so on, instead it would be added and built in those objects. The benefit with that idea is that it will take less time and demand less effort. Another suggestion is to use one single documentation object for design decisions, in all phases. The ultimate change, though, would be to introduce a system for workgroup communication support, which could inspire and encourage the team members and make them feel that the documentation process was fun, easy and meaningful. It seems that the post-hoc worknotes (Löwgren et al. 2002) could have this effect. It is both flexible and does not demand much time or effort of the team members. Since it involves new media techniques, such as digital video- and audio- recording, it would probably be fun to use. Another benefit with such system would
be that the documentation is easier to retrieve throughout the design process and not much time has to be spent on trying to find the documents on the server about what was decided and why. The chance that decisions will actually be there when searching for them, is also great since it would be easy to just turn on the recording system when having important discussions. The system would therefore encourage both flexibility and efficiency in the communication within the design team.

Since the system is flexible and dynamic, it meets the suggestions for Design Rationale systems made by Twidale et al. (1993) who argue that notations used in the early stages of the design process should support the variability in design activity among the designers. A frequent and rapid revision of the design could also be made, since the designers can easily go forward and backward through recorded material in the search of relevant material.

7.6 Backbone objects

Some parts of the design teamwork could be improved if implementing more backbone objects. The results within this category of objects indicated that work was carried out repetitively, good old objects had disappeared, and there was a lack of guidance in the design team work. This meant that many creative ideas, methods and objects were sidelined. As mentioned before, in section 2.2, it is important to emphasise what Preece et al. (2002) call the conceptual stage of the design process and therefore if too many creative methods are sidelined, that will affect the design solutions negatively. In order to make sure that new ideas are really implemented and do not disappear the following suggestions should be considered:

*Implement a new role:* Let someone in the team have the role of having the uttermost responsibility for that new ideas are implemented well and that the work that is committed to them is followed up and encouraged.

*Work out a set of backbone objects* to support the team in the process of implementing new ideas about for example working routines. Each new idea could for example be given a document where it is updated how the implementation of it is doing. This document could pose questions about the problems that are
encountered, ideas on how to get over those problems and with a plan for how the further development and use of it should evolve.

**Revaluate time:** The team will probably not start working differently if time is not revaluated and emphasis is more equally divided between the conceptual stages of the design and the following development of it. Letting the team work more freely in the conceptual stage would probably encourage them more to change their working routines and break the comfortable and safe repetitive manner that work was carried out in.

**Gather more as a team:** It would be easier to hold work together and encourage each other more in the design work if the team could gather more often as a team. The objects proposed above could be one way of bringing the team members together, by encouraging them to involve more in cross role interactions.

### 7.7 A communication resource hub

Now, when the results from each category has been analysed and discussed it is time to gather the findings into some general suggestions about what the design team could do in order to improve their objects and the routines surrounding them.

As suggested above there is a need for more objects for communicating the design within the team. By letting the team use different representations when transferring their work to the other roles, they would get more freedom in the choice of how to present design ideas. Therefore it seems like a good idea to have a variability of different objects to choose from; like a toolbox of design objects such as prototypes, sketches and diagrams. That would probably inspire both the team members creating the objects and the other team members that they interact with. It would support the creative part of the design work and the team members would feel more encouraged doing their work.

Arguing for the use of a large set of artifacts is strengthened by Schmidt et al. (2002) who argued that the multiplicity of artifacts serves an important role in the process of producing a design,
when creating something that does not yet exist and is in the process of becoming.

However, it is possible that the use of more objects can have negative implications. For example it will mean that there will be more documents to keep track on. If having a lot of objects to choose among, it can also mean that each object are not used as often, as if only having one or two objects to choose from. This can result in that the design team learns a bit about many objects and there will never be enough time to implement the objects well enough. Therefore the toolbox of objects must be adjusted to the design team’s working conditions and the projects it is involved in.

This toolbox can be seen as a “communication resource hub” consisting of objects that are optimally good for different things, such as transferring ideas on graphical design or technical limitations. The team should preferably build this "communication resource hub up itself. This could be done by gathering all objects that are used at present and change the prescriptions for the ones that does not yield anymore and for the ones that have been suggested above to be changed, such as the objects used within the functional and detailed design. The team could then decide which objects should be represented according to the team’s capability and the conditions they are working under. It can consist of objects that are created for individual projects, such as the ones discussed above in this chapter; objects adjusted for transferring different parts of the interaction design, such as prototypes, interactive sketches and documents for describing the technical aspects behind the interface. They could also be technical/graphical guidelines used in the development of the detailed design. The communication resource hub could also contain fixed objects that yield for all projects, such as a guideline with illustrations and text about what a template can be. Other objects that should be represented are the ones used within the conceptual stage of the design, since the team in general needs to be encouraged to use these more.

A suggestion is to let each object be exemplified by a template and explained in text, so that the team members easily understand the purpose and use of it. The templates should be gathered in an easily accessible folder on the shared server.
If this suggestion is to be realised, one of the conditions is that it is *practically feasible*: that there is time to set off for implementing this new way of working, and that it is *economically feasible*. It might seem as if implementing this idea would take a lot of time and effort, however, it would probably save time in the long run, since if using the appropriate objects, the team communication will be improved and a common ground obtained. The team members were all positive to change their work if they could see the benefits for the change. So, if being presented the above arguments and if being convinced that it would be both profitable for the team and the external interests to start working this way, it should be realisable. An important observation made throughout the study was that the team was in a need of more guidance in their work and the feeling of being a team needed to be rebuilt. It is likely that the use of the “communication resource hub” would be one step in the process of creating opportunities and possibilities to build a shared understanding about what is being communicated and how it should be communicated. It could be one way of creating a common ground.

### 7.8 Conclusions

An over-riding result is that the design team needs to extend their use of the objects within all categories of focus, in order to be fully supported by them in their work. There is a need of using *more* objects, but also of using already existing objects in new ways. This can be done by adjusting them better to different needs of different external and internal purposes, and by introducing new ways of producing them and using them. For example encourage the interaction between the interaction designer and the art director and make a *detailed functional documentation*: i.e. an updated version of the detailed documentation. The objects could then be used in a more flexible and efficient way.

Schön (1983) proposes that the work of designers resemble the work of an artist who applies different kinds of methods in a flexible manner, which makes their work impossible to capture in a prescriptive method. The results of this study show that normative guidelines must exist but that they should be made more flexible. In the design team this could be done by adding more objects to the methods and prescribing alternative interaction patterns for the different roles, in order to support the
flexible way that designers work in a better way. It is important to keep in mind though that the number of objects must be adjusted to the design team’s resources and capacity.

Houde and Hill (1997) suggest the use of a model for deciding the focus of prototypes, in order to make sure that the right prototype is being used. The results of this study show that this could be applied to design team objects. A similar model could be used in order to decide the focus of all objects.

Reconnecting to the problem statement

The aim of this thesis was to identify the problems connected with the boundary objects that were used in a real design team. The main research question was:

“How should different objects within different areas of communication in a real multidisciplinary design team be used, in order to support the team members creating meaning in their design work as efficiently as possible?”

A “communication resource hub” is suggested where objects optimally good for different things can be gathered. The “hub” should also contain models that can be applied in the process of choosing the appropriate object. The team should preferably build this communication resource hub up itself. That way the team can decide which objects should be represented according to the team’s needs and capabilities. It is believed that the “communication resource hub” would enhance a shared understanding about what is being communicated and how it should be communicated and thereby support the creation of a common ground.

7.9 Future Studies

As a continuation of this work, future studies could further investigate what the resource hub could consist of. Both concerning which objects should be represented and what kind of models could be used as a support when choosing the right objects for the right purposes. In this study it was suggested that the design team could use a model to decide the focus of prototypes. However, it would be interesting to investigate and develop a model, which is appropriate for all boundary objects.
DISCUSSION AND CONCLUSION

It would also be interesting to investigate each category of objects more focused. There is still a need to carry out more case studies in order to gain insight on how design team methods could be better adjusted to the needs of design teams. Research could for example take stand in a model such as the Star lifecycle model. Can work really be carried out flexibly and in parallel?
8 References

Alexandrou, Marios 2004. *Custom software & Web site development.* [www]
From: <http://www.mariosalexandrou.com/glossary/graphic_designer.asp>
24 August, 2004


Baxley, Bob (2002) [www]


Cipid/MIT: *A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development.* [www]


REFERENCES


Sapdesignguild: Example Vision [www]
<http://www.sapdesignguild.org/editions/interact_design.asp>

[www]<http://citeseer.ist.psu.edu/cache/papers/cs/26822/http:ZzzSzwww.it-


Wikipedia:
Appendix:

The waterfall lifecycle model
The spiral lifecycle model
The star model
A template for the design team functional sketch
Detail of flowchart diagram
The Post-hoc worknotes
Work models made for the analysis

1. The waterfall model
2. The spiral lifecycle model

![Spiral Lifecycle Model Diagram]

3. The star lifecycle model

![Star Lifecycle Model Diagram]
4. The design team templates for the functional sketch
5. User Model (©Smith, 2002)

6. A wireframe example (©Alan Smith, 2002)
7. Detail from a flow-diagram (© Alan Smith, 2002)
8. The post-hoc work notes. (Löwgren et al., 2001)
9. Work models
A physical model for the design teams working environment

Explanations to the model:

The two interaction designers (id1, id2) and one of the project leaders (p.leader) were sitting on the right hand side of the office. A bit further into the room on the same side, just before the wall, were the desks of the art directors. On the opposite side of the room were the copywriter/project leader and the manager. On the right hand side, on the other side of the wall, after the art directors, were the conference room where meetings were held. That room had two whiteboards, one computer, a table with chairs and a telephone. Opposite the conference room was a room with printers and faxes. After the conference room were the desks of the programmers (ifp, sd1, sd2).
The green arrows indicate where the communication was working satisfactorily, in the sense that the roles talked to each other and passed by each others desks. The red arrows indicate where the communication tended to fail; mainly between the programmers and the designers.

The figure below illustrates some new working environment elements that could be introduced. To the left a library, with the possibility to take a break from work, read books or magazines and sit in comfortable sofas. The library would offer an alternative environment if a design team member is in the need of piece and quiet or a more inspiring environment than the one of the office. To the left is a quiet working room where team members could go if in the need of peace and quiet in order to concentrate better on their work.

*New concepts to be introduced in the design team working environment*
Titel | Att kommunicera i ett design team - skapa mening genom “boundary objects”.
--- | ---
Title | Communication in a Design Team - Creating meaning in a design team through boundary objects
Författare | Sara Lindstedt
Author | Sara Lindstedt

Sammanfattning
Abstract
This thesis explores object based design team communication. It is assumed that boundary objects in design teams serve as an important communication aid and are considered to have a crucial role in the conducting of multidisciplinary teamwork. Objects, such as design specifications and prototypes, can for example help bridging knowledge gaps between the different interests involved and offer guidance and support in their design work. The aim was to identify the possible problems that might occur connected to the different objects used in a design team. In order to study this, an ethnographical inspired study was carried out. Overall results were that the design team needed to extend their use of objects, in order to be fully supported by them in their work. Further, the existing objects needed to be changed or used differently. A “communication resource hub” was suggested, where all the new and old objects could be gathered. In this “resource hub” there should be room for different models that could be applied as a support for deciding on the right objects for the right purpose.

Nykickelord
Keyword
Boundary objects, team design, communication, roles, interaction design, usability, documentation