Video Collaboration in Post Production

KARL-MAGNUS WÄNDELL

Master of Science Thesis
Stockholm, Sweden 2010
Video Collaboration in Post Production

KARL - MAGNUS WÄNDELL

Master’s Thesis in Media Technology (30 ECTS credits)
at the School of Media Technology
Royal Institute of Technology year 2010
Supervisor at CSC was Christer Lie
Examiner was Nils Enlund

TRITA-CSC-E 2010:089
ISRN-KTH/CSC/E--10/089--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc
Video collaboration in Postproduction

Abstract
The Chimney Pot is a postproduction company that works with feature films and commercials. It's in postproduction the films are put together and special effects, graphics and sound are added. Today The Chimney Pot uses several applications to organize their projects, employees and their tasks. Their system for handling effect clips, the tracker, is incomplete, not user friendly and too complicated.

The problem can be divided into three categories: bad interface, limited functionality and disorganized file handling. They need a new system that can solve these problems. The aim of this report is to create a proposal for a new system. This proposal could also be of interest for other companies in the postproduction industry.

Workflow and production processes have been investigated and user requirements have been specified. Online standard solutions have been tried and evaluated in order to find a suitable solution. None of the online systems were able to cover all their requirements. A proposal for a new system has been created in close cooperation with artists and producers. The result is a proposal for a new tracker that should be easy to use, able to automatize repetitive tasks and able control the file structure on the network. An interactive prototype with detailed description has been created.

Videokollaboration inom postproduktion

Sammanfattning


Index

1 Introduction ........................................................................................................1
  1.1 Problem ........................................................................................................1
    1.1.1 User interface ......................................................................................1
    1.1.2 Functionality - what the tracker should do ......................................1
    1.1.3 File handling ......................................................................................1
  1.2 Purpose .......................................................................................................2
  1.3 Question .....................................................................................................2
  1.4 Limitations .................................................................................................2
  1.5 Audience .....................................................................................................2
  1.6 Literature .....................................................................................................3
  1.7 Vocabulary ..................................................................................................3
  1.8 Report structure ..........................................................................................3

2 Background .....................................................................................................4
  2.1 Film production ............................................................................................4
    2.1.1 Pre-production .....................................................................................4
    2.1.2 Production ............................................................................................4
    2.1.3 Post-Production ...................................................................................4
  2.2 Digital Asset management, DAM .................................................................4
    2.2.1 What is DAM? .....................................................................................5
    2.2.2 Without DAM and with DAM? .............................................................5
    2.2.3 What can DAM do? .............................................................................6
    2.2.4 Different DAM solutions .....................................................................6
    2.2.5 System components ...........................................................................7
    2.2.6 Multilayer architecture .......................................................................7
    2.2.7 Storage networks ................................................................................9
    2.2.8 Implementing DAM ..........................................................................9
  2.3 Human Computer Interaction (HCI) ............................................................10
    2.3.1 What is HCI? .......................................................................................10
    2.3.2 Benefits of HCI ................................................................................10
    2.3.3 What is usability? .............................................................................10
    2.3.4 What is a usable system? ................................................................10
    2.3.5 Contextual design ..............................................................................11
    2.3.6 Conceptual design .............................................................................11
    2.3.7 User centered approach ...................................................................12

3 Method ...........................................................................................................13
  3.1 Questions ...................................................................................................13
    3.1.1 What functionality should bee included? .........................................13
    3.1.2 How should the files be handled? ......................................................13
    3.1.3 Should the new system be bought, rented or built from scratch? ...14
    3.1.4 How should a new system be designed? ..........................................14
    3.1.5 How is a prototype created? ..............................................................14
  3.2 Approach ...................................................................................................15
5.1.2 Prioritizing requirements .................................................................40
5.1.3 Conclusion ......................................................................................42
5.2 How should the files be handled? .......................................................42
5.2.1 Conclusion ......................................................................................43
5.3 Should the new system be bought, rented or built from scratch? ..........43
5.3.1 Extend existing system .....................................................................43
5.3.2 Rent standard system .......................................................................43
5.3.3 Buy standard system ........................................................................44
5.3.4 Combine systems ............................................................................44
5.3.5 Build a new system ..........................................................................44
5.3.6 Conclusion ......................................................................................44
5.4 How should a new system be designed? .............................................45
5.4.1 Screen design ..................................................................................45
5.4.2 Menu design ...................................................................................45
5.4.3 Icon design .....................................................................................46
5.4.4 Conclusion ......................................................................................46
5.5 How is a prototype created? ...............................................................46
5.5.1 High fidelity vs. low fidelity prototypes ..........................................46
5.5.2 User tests ........................................................................................47
5.5.3 Conclusion ......................................................................................47
5.6 Other issues .......................................................................................47
5.7 General solution ................................................................................48
5.7.1 Software development ....................................................................48
6 Recommendation ..................................................................................49
6.1 Workflow ..........................................................................................49
6.2 Functions ..........................................................................................50
6.2.1 Prioritizing requirements ...............................................................50
6.3 Description of the functions - daemons ............................................51
6.3.1 The flashplayer ..............................................................................54
6.4 Additional features ............................................................................55
References .............................................................................................57
Appendix ...............................................................................................59
1 Introduction

In this chapter the reader will get a brief overview of the rapport. The purpose of the report and five questions are presented. The reader will get an understanding of what is included and not in this report. Several important abbreviations are also explained.

In today’s society the computer has become an essential tool. Everyone that works with computers create many different files. After a while the files can be too many and the user looses control over his organization. Too much time can be spent finding files. When this is multiplied many times, for example in a company, it will get even worse. What is the solution to this problem? One way is to have a German order and be extremely strict while creating and naming files and folders. In big companies this is often very hard to obtain. Another way is to install a program that keeps track of all the files and the changes made to them. The system creates folders and helps the user to organize things. It is one of these systems that is described in this report. (Jacobsen, 2005)

The Chimney pot is a postproduction company that work with editing and making special effects for movies and commercials. The company was started in 1996 in Stockholm and has since then grown and have now nine subsidiaries. They have offices in Stockholm, Kiev, Warsaw, Oslo, Dubai and Denmark. They produce about 700 commercials and 15 feature films per year. (The Chimney Pot, 2009). Each film or commercial is called a project. In a project there are many different files involved. There are image sequences, movie clips, images, pdf-files etc. In big projects it can be very hard to keep track of all files and their versions. Today the company uses several different applications to organize projects and keep track of tasks. They have one program for economy and booking, one “blog” to present clips to clients, one “tracker” to assign tasks and one program for previewing high-resolution files. At the moment they are not fully satisfied with their system.

1.1 Problem

The Chimney Pot's tracker (system for handling tasks regarding special effects) is not user-friendly, too complicated and doesn’t communicate with other systems. There are several unnecessary and time-consuming processes. Another drawback is that all their requirements are not met by the system. They need a new system in order to handle the projects and their files more efficient. This report focus on the system for handling video files and tasks and procedures related to them. The problem can be divided into three sub problems, user interface, functions and file handling.

1.1.1 User interface

The design of their tracker is not user-friendly, has poor design and shows too much irrelevant information. The layout needs to be cleaned up and redesigned. There are several different types of users. Different people use the system in different ways and have different backgrounds. There are three types of users: clients, producers and artists.

1.1.2 Functionality - what the tracker should do

The tracker should have several functions in order to facilitate working and collaboration. In order to know this the requirements of the users must be defined. In standard systems there are too many unnecessary functions and features. The most essential functions need to be identified and sorted out from wishes and other irrelevant functions.

1.1.3 File handling

There are many ways of organizing files and folders. For large companies working with huge amounts of media files it’s extra important to have an organized file system. Files
have to be transferred, modified, conformed and archived. Versions of the all these files must be tracked. Today users create folders as they like and put stuff where they want. In order to prevent this there must be a strict system how the files should be handled.

1.2 Purpose

The purpose of this report was to examine The Chimney Pots current system and create a proposal for a new system. The proposal consists of a prototype and a report explaining in detail how the system should work. A programmer should be able to understand the report and without doubts create the system.

1.3 Question

These sub problems have lead to five questions that have been answered in this report.

- Should the new system be bought, rented or built from scratch?
- How should the new system be designed?
- What functionality should bee included?
- How should the files be handled?
- How should a prototype be created?

To answer these questions the developer first had to understand the company and what it does. Then the users and their needs where identified and their requirements specified. Then other companies were investigated and compared. Then available online standard solutions was tested and evaluated. A design proposal and an interactive product were made in order to describe a new system. The aim was to have a user-centered approach and involve the user during the whole process. Several user tests were made to be sure that the users understood the system.

1.4 Limitations

To use this kind of systems are potentially interesting for all companies that work with film and postproduction. To investigate the whole market and do a general system would be too much for this report. Many of the big standard systems are meant to fit everyone, but fit no one (Swartling, 2006). This report will focus more on one tracker suited for one company. Other companies have been investigated in order to understand how they solved their problems.

It would also be possible to implement the system, in other words to create the program. That would require a lot of studying and research about programming and databases. This will not be included in this report. These techniques are described briefly. A second part of this project has been created. Another thesis is done in the area of computer science. The purpose of that report is to investigate different techniques and methods for programming the system.

1.5 Audience

The audience for this report is programmers that should be able understand the problems involved and create a system for The Chimney Pot. It is also targeted to other people in the postproduction industry interested a system to manage and organize their files.
1.6 Literature
To get a good start mainly two other master thesis was investigated; one in media technology and one in Human-Computer-Interaction. The first one, *Improvement on the Workflow at a Swedish Post-Production Company in Moscow* gives an introduction of what The Chimney Pot does and their workflow (Gomez, 2008). The second report is *Create Simpler worlds* about Human Computer Interaction (Kiby, 2006). Other books in Digital Asset Management and HCI were also read in order to get a better understanding.

1.7 Vocabulary
Some important terms need to be explained in order to understand all the content of this report and how the tracker works.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAS</td>
<td>Network Attached Storage</td>
</tr>
<tr>
<td>SAN</td>
<td>Storage Attached Network</td>
</tr>
<tr>
<td>InfiniBand</td>
<td>High speed fiber connection between computers</td>
</tr>
<tr>
<td>Phyton</td>
<td>Programming language used</td>
</tr>
<tr>
<td>MySQL</td>
<td>Standard for database</td>
</tr>
<tr>
<td>ARRI scanner</td>
<td>A scanner to transfer films to digital medium</td>
</tr>
<tr>
<td>RED</td>
<td>New high-resolution digital film camera. Up to 4K resolution</td>
</tr>
<tr>
<td>Farmers WIFE</td>
<td>Booking program for projects</td>
</tr>
<tr>
<td>FLAME, Smoke, Flint</td>
<td>Integrated editing online system from Autodesk</td>
</tr>
<tr>
<td>Maya</td>
<td>Program for 3D animation from Autodesk</td>
</tr>
<tr>
<td>DPX</td>
<td>Digital Picture Exchange, high-resolution image</td>
</tr>
<tr>
<td>TimeCode</td>
<td>The numbers used to locate frames and cuts in films</td>
</tr>
<tr>
<td>Reel</td>
<td>The physical roll of film used in the cameras</td>
</tr>
<tr>
<td>Flash player</td>
<td>Tool used for video collaboration, drawing on specific frames</td>
</tr>
<tr>
<td>VFX</td>
<td>Visual Effects, special effects</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>DAM</td>
<td>Digital Asset Management</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language, standard for information exchange</td>
</tr>
<tr>
<td>EDL</td>
<td>Edit Decision Markup Language, in-points and out-points of cuts in a film</td>
</tr>
<tr>
<td>Off-line</td>
<td>Working with a low-resolution of the original material</td>
</tr>
<tr>
<td>Online</td>
<td>Working with the high-resolution material</td>
</tr>
</tbody>
</table>

1.8 Report structure
*Chapter 2 - Background* explains how postproduction companies works, what digital asset management systems are and the theory behind creating a new user-friendly system.

*Chapter 3 - Methods* that were used to obtain information and solve the problems. An explanation why the methods where used is also presented.

*Chapter 4 - Result* explains the results of the methods and an analysis of the results.

*Chapter 5 - Discussion* deals with some discussions regarding methods and functions

*Chapter 6 - Explains* is a recommendation to The Chimney Pot how their new tracker should look like and what functions that should be used.
2 Background

This chapter gives the reader an understanding of film production, Digital Asset Management and Human Computer Interaction.

2.1 Film production

A film production can be anything from a big Hollywood production to recording simple wedding films. The process can be divided into three stages, preproduction, production and postproduction.

2.1.1 Pre-production

Here the storyboard and the scripts for the actors are defined. The budget is set and crew assembled. Different shooting locations are checked out and selected. (Ascher & Pincus, 1999)

2.1.2 Production

The actual production is when the shooting takes place. The crew gathers and the scenes are recorded. There are mainly three mediums used for recording media: film, tape and hard-drives/memory-card. In film production the analogue medium film is still widely used. It's been used for a long time and always gives a high quality. The digital mediums like hard-drives and memory-cards are being more and more used.

2.1.3 Post-production

In the postproduction the film is put together. Clips are edited, color graded and special effects and sound is added etc. First the director and an editor edits the film off-line (in low resolution). Then the selected shots are scanned and converted to the correct format. Then the online process starts and the high-resolution material is edited and graphics and sound are added. Many times material is moved between companies and converted into different formats (Austerberry, 2006).

Input

Material can enter the postproduction in different formats, but one thing they have in common is the Timecode. In order to know where cuts are the video and audio are logged towards a timeline. It's based frames, seconds, minutes and hours (hours:minutes:seconds:frames). The maximum value is 24 (23:59:59:24), which is more than enough for most occasions. (Ascher, 1999)

The most common film formats are 16 mm and 35 mm (24 frames/s). Films are stored on reels. A 16 mm film reel is between 27-33 min and 35 mm reels are 10 or 22 minutes. The shorter one is normally used in the camera. In postproduction the films divided into acts of 18 to 24 minutes. The film-camera nowadays has something called a video assist. It diverts light from the camera so that it's possible to watch and record material digitally. This is very handy because the film can be edited low resolution before the film have been developed and scanned. (Asher & Pincus, 1999)

Some of the most common tapes are DV, DVCPRO, DigiBeta, HDCAM and D1-9. A digital example is the Panasonic P2-card. It's a high-speed flashcard used for store high definition formats (Austerberry, 2006). Another format that has become more common is the RED-format. It can have resolutions from HD to 4K.

2.2 Digital Asset management, DAM

Today everyone that works with computer creates many files. It can be word documents, excel sheets, imported photos, e-mails, films etc. When time passes these files, or assets, tend to be
quite many. It is hard for most people to organize their files in a good and systematic way. If one user has a good way of organizing his files it's not sure that other people will understand. This disorganization among the files many times lead to inefficiency. People tend to spend more and more time in finding their files. In big project this is a very big problem. This is a problem that almost all companies have to deal with. (Jacobsen, 2005)

2.2.1 What is DAM?
First we need to define what an asset is. One definition is that an "asset = file + rights" (Jacobsen, 2005). DAM is a system to organize assets. When a DAM system is installed the user doesn't have to worry about where the files are. All files are saved in an organized way by the DAM system and everyone will get a good overview over the assets. DAM also helps you in performing boring and repetitive tasks like making thumbnails convert to other formats etc.

2.2.2 Without DAM and with DAM?
What are actually the differences between having a DAM implemented and not? What can DAM do for you? The most important thing is to have a good project management. If the project management is bad from the beginning tools will not help so much. Some research have been made about project management:

An average user creates 50 - 10 000 new files week (Jacobsen 2005). The user can then move, delete, rename, add files both locally and centrally. If several people work with the same file a new version of the file can be overwritten by an old version. Finding, loosing and overwriting files can take 25-40 % of working day (Jacobsen, 2005). An average creative artist searches for a media file 83 times per week and fails to find it 35 % of the time (Ross, 1999).

The complexity increases with the time size. When the team size goes over 20 people it is extremely hard to know who did what. (Jacobsen, 2005) In film production the team size will increase many times when the postproduction phase is reached.

Even if backup is done it can be very hard to retrieve lost files. The files have been backed up without history on some backup tapes. It might take a lot of time to find what file and what version that is the correct one on the tape. This might also prevent people from using material from old archived projects.

Another problem is the quantity of old versions. The users create copies of their and other peoples work and later forget or are afraid of deleting them. A lot of storage space is wasted on this. (Jacobsen, 2005)

With DAM implemented can lead to 12.5 -20 % more productivity (Jacobsen, 2005). Because less time is spent on file handling more time can be used for creative work. The artist can then either produces the same result in less time ore a better result in the same time. With DAM implemented a creator only fails to find a file he's searching for 5 % of the time (Ross, 1999).

DAM also leads to less frustration over and overwritten files. Easier and more professional working conditions can also lead to higher motivation. The artist can spend time on the things he likes instead of doing boring repetitive tasks. He can reuse material from old projects.

The system will be more reliable and a brief for the project leader. He doesn't have to worry about small things. The deadlines can easier be met and backup is handled automatically. Fewer misunderstandings could arise because all communication is stored on one place.

It allows artists to reuse old material (Fig. 1) so they don't have to redo everything every time because he cannot find the files (Jacobsen, 2005).
2.2.3 What can DAM do?

DAM can provide many services for the company. Below follows a summary of different features that a DAM can provide (Austerberry, 2006):

**Classification**: Metadata is added to the assets. It can be filename, author, format, keywords etc. The assets can be categorized in different categories. This is essential if the material should be archived and found again.

**Searching**: After metadata is added it will be easier to search among assets. There are many different search techniques that could be used. Simple keywords, Boolean operators (AND, OR etc), word stemming (connections between synonyms), fuzzy errors (spelling correction), natural language (free text search) and Semantic net (the computer have knowledge about grammar, synonyms etc) could be used.

**Task and project management**: Assets could have tasks connected to them. The tasks would then be sent to the different project members. The project leader would have an overview of all tasks and be able to approve them.

**Workflow automation**: Tasks and assets could automatically be sent to other coworkers when they are finished. Automatic messages could also be sent to the different project members.

**Collaboration**: The workers are not bound to work inside the company walls anymore. Tasks could be reached from anywhere in the world. The teams could be spread all over the world.

**User management**: Different users can have different rights. This can be very important for so that things cannot be deleted or added by the same person. It also helps to filter the huge amount of information. Only relevant things will be shown.

**Storage management**: This is one of the most important features. Everyone could be able to see all files, regardless where they are physically located. All assets could be kept at a centralized storage so that everyone could reach them. Local sandboxes (a local attached storage) could be used for users download local copies for editing without affecting the original material.

**Backup and archiving** could be handled automatically. When the projects are finished and the files not used it could be sent to archive.

**Version control**: In order to track changes and prevent that old version are overwritten a version control can be implemented. When a change is made a new version is created. Old versions can easily be viewed.

**Check-in check out**: This will prevent two users from working on one file at the same time. When someone is editing a file first check it out and when he's done he'll check it in again.

**Automatic conversions**: Conversions to other formats could be automatic. Many different formats could be created through the system from the high-resolution files.

**Content editing**: The content could be edited online. A film could be edited in low-resolution while the high-resolution file is created automatically.

2.2.4 Different DAM solutions

There are many different DAM solutions on the market. They are specialized for handling different kind of management issues. The complexity and the number of potential users vary dramatically. Austerberry have defined and categorized them into the following (Austerberry, 2006):

**File systems** are the simplest form of DAM. Examples are Mac OS (GFS+), Windows (NTFS) and Linux (UFS). Here the user himself stands for the management. If the user follows the rules a good organization can be obtained. Backup, metadata and rights management are quite limited and no version control exists. If files get mixed or lost it can be time-consuming and expensive.

**Brand Asset Management** (BAM)

Many multinational companies use BAM to share assets between their different offices in different countries. The assets can be logos, pictures, press release, documents etc. These systems are expensive and training of personal is needed.
**Document management** (DM)
DM is similar to BAM but focus more on documents, reports, calculations, contracts etc.

**Web Content Management** (VCM)
Also VCM is similar to BAM but focus on handling assets in big websites.

**Enterprise Content Management** (ECM)
ECM includes BAM, DM and WCM. These systems can be very complex and take can take many years to implement. They can on the other hand save a lot of money.

**Single-user revision control**
This tracks different file versions on a single computer.

**Software Configuration Management** (SCM)
SCM is developed for software development. Everyone in the project is updated with everyone changes. SCM handles versions and prevents overwriting. SCM have restricted functions for text and media files.

**Digital Asset Management** (DAM)
DAM handles media files such as images, sound and video. Thumbnails, metadata and preview of the files are shown. The companies that uses DAM are for example publishing houses, video production and broadcast companies.

**Production Digital Asset Management** (ProdAM)
ProdAM is more focused on production workflow. It supports most media files and have many times archiving functionalities. The idea is quite similar to SCM, but here binary files are handled. ProdAM can be regarded as one solution that fits all people. It requires huge storage capacities and often needs a long installation time. It is this system that is most suitable for a postproduction company like The Chimney Pot.

### 2.2.5 System components
A system can be either web based or software based. A software based (Client - server) system normally works well for a small business. It is basically software installed on the client computer. In big companies with 100-1000 computers the maintenance of such a system can be very hard and time consuming. A rule of thumb for big companies: the less applications installed - the better. The web based system uses a *multilayer architecture*. (Austerberry, 2006)

### 2.2.6 Multilayer architecture
For complex system it is more suitable to have a multilayer architecture (Fig. 2). It gives freedom to set up as many servers needed for all applications. There are three main layers:

![Multilayer Architecture Diagram](image)

*Figure 2. Components of a DAM system*
The Presentation layer is separated from the application layer. It can be managed by standard web tools, like a dynamic website. It is very simple and no extra application is needed. Even complex functions as controlling external tape recorders can be managed through a web interface. (Austerberry, 2006)

The user interface is often crucial to the acceptance of a new system. Within the limitations of HTML it should be powerful, but still simple (See 2.3 HCI). If more functions are needed plug-ins as JavaScript, Java or Flash could be implemented. For total freedom HTML could avoid and a software could be created. Then the problems regarding software will arise again. An example of a good software application is EasyCut (Fig. 3) from Vizrt used by The Swedish Television, CNN, Fox Business News among others (Vizrt, 2008).

Application layer stands for the business logic. It can be handled by a single server or by different servers for different applications. There could for example be one server handling the indexing, one the conforming, one the workflow etc. It all depends on how many users and how complex the system will be.

In order not to build everything from scratch some software components are available. They are called frameworks and the idea is to reuse already written code. Some of the most common frameworks are (Austerberry, 2006):

- **COBRA** (The Common Object Request Broker Architecture) that handles distributed computer networks. It is used by many of the leading DAM solutions.
- **J2EE** (Java 2 Enterprise Edition) is another framework that supports XML exchange.
- **Microsoft .NET** is a complete software platform. It is built on XML-technique.
- **Django, Pylons** and **Zope** are open source Python based frameworks. Django is a framework for perfectionists and make it easier and faster to build better web applications. Pylons is a light weighted framework that focus on flexibility and easy development. Zope is used for application servers and are used my many Content management systems. (Python, 2010)
- **XML** is a meta language used to discuss other language, like we use grammar. (Austerberry, 2006). It is used for different applications to talk to each other. It is also used to export metadata film project in editing systems.

The data layer consists of the database and eventual media servers. The database is the core of the asset management system where all metadata are stored. Pointers to the media files stored on different hard drives or tapes are also stored in the database (Austerberry, 2006). To make the database faster and easier it's very important to build a good and logical database model. There are several types of databases available: flat file, hierarchical, network, relational, object oriented, XML, etc.
The relational databases are the ones most widely used. They have been on the market for a very long time and are reliable, well tested, scalable and many people are familiar with them. They consist of different tables with attributes (columns) and tuples (rows with data posts) (Austerberry, 2006). SQL (Structured Query Language) is a standard for pulling and modifying data in a relational database.

### 2.2.7 Storage networks

In postproduction companies' and other business working with video files very big storage networks are needed. Simple SD (Standard Definition) video has bitrates of 270 Mb/s and HD (High Definition) five times more (Austerberry, 2006). There are mainly two existing solutions:

**Network Attached Storage** (NAS). A NAS is a server with several hard drives with a network card connected to the network. A remote file system is used that supports Mac OS, Unix and Windows. A single copy of a file could be stored and be reached from all systems. A NAS is very easy install and when it gets full it's just to insert another server. All other computers in the network could access the NAS from the LAN. (Jöns, 2008)

**Storage Attached Network** (SAN) is a network that can distribute and store big amounts of data to users in a network. The computers attached to the SAN regards the SAN as local storage, while in NAS it's a separate unit. Only fiber clients can be connected to the SAN. The information is transferred in raw disk blocks instead of files. The maitance of a SAN is much more complex. When the SAN is full it's just to connect another hard drive. (Jönson, 2008)

### 2.2.8 Implementing DAM

To implement a system there are many different ways to go. If a system already exists it might be possible for the company themselves to extend or modify it. There are often many different scripts or plug-ins that could be used. If this is not possible there are several ways to go:

- Buy a new system
- Rent a new system
- Let a system be modified
- Combine existing systems
- Construct (tailor-make) a new system from scratch

If a solution is found that fits the company requirement it's easy to buy it directly. A system could also be rented. The implementation will go fast and it will be relatively cheap because no hardware or developing team is needed. Support, daily backups, unlimited number of users is provided etc. (Gonnering, 2004) A negative aspect is that all traffic must go over the Internet. If the assets are big, like video files, this can be a problem. If one single system doesn't meet the company’s needs, several different solutions could be combined. Though it's important that they are able to work together. They could also be modified and combined. (Auserberry, 2006)

If nothing suits, a new system can be created from scratch. This is a very hard task and most companies that tried failed. The most important tip is not to aim to high. The aim shouldn't be to build the best system ever. It's enough to build a small working pilot for in-house testing. (Austerberry, 2006)

There is some danger in using a standard system. All users can theoretically use these systems. To develop an own system is much more expensive. Standard systems can theoretically lower the company's developing. A new system will be obtained directly and company doesn't have to develop the system their selves. This often results in a system that is not usable for anyone. Later the system has to be developed and modified. This means that it's not a standard system any more; it's a specialized system. These systems often get more expensive and don't work as good as a tailor-maid system. (Vimarlund, 2006; Swartling, 2006)
2.3 Human Computer Interaction (HCI)

Today we use interactive products at work, at home, while traveling in other words almost everywhere. It can be mobile phone, computer, ATM, TV-set, printer, stereo, mp3-player etc. Unfortunately not all of these products are easy to use. This many times leads to frustration and lost time. At work a normal person spends about two hours per week solving computer related problems (Artman, 2006). Its here HCI enters, in the borderland between humans and machines.

2.3.1 What is HCI?

It's about understanding the users, their situation, information dependencies and change in interaction. This regards needs, demands, alternative design, construction and evaluation. (Artman, 2006)

"Every response you make provokes a reaction from the computer, which leads to a new response, and so on, as the loop from the screen to your eyes to your fingers on the keyboard to the computer to the screen becomes a single cybernetic circuit". (Jayakanthan, 2002)

2.3.2 Benefits of HCI

The benefits from using HCI during the development process are many: The users don't need to spend time fighting with a bad interface. Late and expensive changes could be avoided. Users will think the new system has higher quality. Learning the system will go faster less support will be needed. The product will be more competitive on the market. The users will be positive, they will not fear to use the interface.

The number of failed projects in Sweden 2006 was 84% (Vimarlund, 2006). Some failed totally and some exceeded deadline and budget. Most project fails due to a bad usability. The most common reason is the lack of user involvement and the understanding of user needs (Usabilitypartners, 2006).

2.3.3 What is usability?

Usability is about how easy it is to use a product or a system. (Artman, 2006)

"Usability: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."(ISO 9241-11)

Usability is a desired quality of computers, programs, interfaces and other tools. The usability is high if its easy to understand what a product should be used for, how it should be used and then be able to use it. (Vimarlund, 2006)

2.3.4 What is a usable system?

Usability concerns everything we use, from light switches to advanced cell phones. Something that is badly designed or to complicated, boring, ugly, etc can be regarded as having low usability. An object should be funny, esthetically pleasing, motivating, rewarding and entertaining. An example of good usability can be the iPhone from Apple, while many complicated remote controls for TV-sets and DVD’s can represent bad usability. The goal should be to optimize the interaction with the systems to facilitate for the users to do their everyday tasks. In order to do so the following usability goals have been defined (Preece et al, 2002):

Effective means that a system must fulfill its original purpose. It must be god to do what it is meant to do. A phone should be easy to call with.

Efficient means that tasks should be solved fast and easy, without extra unnecessary steps.

Safe means that the user should be prevented from using the system in a wrong way. Only the available options should be selectable, other options should be grayed out and not clickable. If the user does something incorrect he should be warned with a text message or by a sound.
**Easy to learn** means that all buttons should be logical, simple and look "pushable". They should encourage the user to press them. Everything must be consequent to avoid confusion. A example of inconsequence is the short command for copy and paste in Mac OS and Windows.

**Multifunctional** implies that there should be many different functions and services. They should be relevant and satisfy the users needs or create new.

It should be easy to remember how to use the system. No short commands or abstract menus should be used. Icons should and explaining text should remind the user how to navigate while using menus or pressing buttons.

**Understanding users**

In order to better understand the user we must learn how they think. Higher functions brain such as: perception, language, communication, memory, attention, ability to plan and perform activities and problem solving abilities are called cognition (Artman, 2006). Cognitive problems in user interfaces are invisible and are hard to discover and understand. They affect our quality of life and ability to work.

Cognitive science tries to predict and give norms for thoughts and actions and evaluate methods and result. It aims to give a better understanding of how users work and think and how design and construction processes should be planned. Cognition can be divided in to three levels: (Artman, 2006)

**Automation** involves learned actions that need no extra thoughts (like walking).

**Rules** tells what the user cannot do, for example going towards red light

**Knowledge** regards why something is done in a certain way

Other division is between experimental and reflected cognition. Experimental cognition is when the user effectively reacts towards his surroundings without much thought (bike riding, talking) Reflection cognition is when he needs to think properly, compare possible alternatives and come to conclusion (Precce et al, 2002)

Today it's an established method to apply and transfer cognitive knowledge into digital systems. (Precce et al, 2002). The design of an interface highly affects people’s way of perceiving, involving and learning the functions. If a good design should be obtained much preparatory work is needed. Interviews with both users and developers and different analysis must be made. (Gustafsson, 2006). In order to understand the user even better it is needed to know his background.

### 2.3.5 Contextual design

The context can also be regarded as the background. We have something in focus and the background all around us. To understand the user the background must be analyzed: how the organization works, how work is done, society, time etc. How a user reacts depends on time, what happened earlier, what he is going to do next etc. It can also bee seen as parts of a whole. Sometimes the parts are understood, but not the whole. (Räsänen, 2006)

All artifacts properties are situation dependent. If it’s fast and effective it's regarded as bad if it's not understood. If it's easy and understandable it's regarded as bad if it doesn't fulfill the users needs. A perfect artifact is useless if its not delivered in time. The definition of good design depends on social laws, rules, edicts, deals and agreements. (Solterman, 2004)

### 2.3.6 Conceptual design

The goal of conceptual design is to transform the user requirements into a understandable conceptual model. This is very important and can be crucial in order for the users to understand the system.

**Interface metaphor**

An interface metaphor is something that can help the user understand the system better. Something familiar from the real world is taken to represent something in the new system.
Video collaboration in Postproduction

(Preece et al, 2002). An example can be the Finder in Mac or Explorer in windows. All maps represents maps from the real physical world. In those maps different things can be stored. If a good metaphor is found it can facilitate much for the users.

Design

There are three mayor parts in designing a system: menu design, icon design, and screen layout (Preece et al, 2002). All these so called widgets must be selected very carefully. They must fit in the context and be easily understandable by the users.

Menu design: A menu offers several different choices of commands. When a user solves a task he will use a set of commands. It's those commands that are in the menu. It is therefore essential to have the users tasks specified while making the menu. The most important issue in menus is grouping. All functions that are related should be grouped together. (Preece et al, 2002)

Icon design: To design icons is quite a hard task. An icon can be designed fast, but will probably not be accepted by the users (Preece et al, 2002). The symbols must be recognizable and associate to the correct function.

Screen design: When designing a screen there are two main aspects. The first is how to distribute the different tasks among a number of screens. The second is how these smaller screens should be designed. In order to understand the users tasks must first be divided into subtasks and plans of action. These tasks and subtasks can then be placed in its own screen. The smaller screens should be designed so that the users attention is dawn to something useful for them to solve their tasks. Different colors, text, popup windows can be used for this. (Preece et al, 2002)

Externalize thoughts

All properties of an artifact are design decisions, considered action by the designer (Solterman, 2004). All decisions can give expected or unexpected side effects. This cannot be known in advance, the user must test the system. Not only different users act different, even the same user act different in different contexts. This makes the design process very complex and rational decision not possible. All users cannot be pleased.

To be able to handle the complex design process thoughts could be externalized in form of sketches, prototypes, drafts, models etc. The designer now has something concrete to reflect over. The sketch stimulates creative thinking, shows obstacles, unexpected properties and new openings. Ideas could be tested; objects could be turned, twisted and reviewed critically. The sketches structure the designers’ thoughts, logics and remind him of restrictions. His thoughts will also be visible for others to see, test, analyze and criticize. (Solterman, 2004) Prototypes are used to avoid misunderstanding, test technical products in a first stage. Something to touch and feel is much more describing. (Preece et al, 2002)

Conceptual design implies that user tests should be made on place, in the users correct context. It's many times a hard task to combine a digital artifact with people social environment into a closed circuit. (Solterman, 2004)

2.3.7 User centered approach

The users should be involved early and during the development. It's that user that sits on the information regarding the systems survival. Early in the process focus should be laid in understanding the users, their behaviors, attitudes and cognitive ways of thinking. The users should be integrated in user tests with prototypes. The system is meant to support them in their own work. The three characteristics of interaction design (Preece et al, 2002):

* Early focus on users: It is important to understand the users, their needs and how they work. *
* Empirical measurement: The users reactions are first documented. Later are the users reactions measured, recorded and analyzed. *
* Iterative design: In order to obtain robustness in the system a new test should be made after each fixed problem and the effects observed. A cycle is created "design - test - measure - redesigning".*
3 Method

The methods used to obtain information were mainly interviews. Different companies have been interviewed regarding their file handling and workflow. All the interviews have been filmed for later analysis. Many different existing online systems have also been tested and evaluated. The advantages and disadvantages have been mapped.

The main focus has been to investigate how The Chimney Pot works. Interviews have been made with, editors, a system administrator, an executive, producers, conformer and color graders. Some of the users have been observed while working in order to get a better understanding of their work and needs. Direct questioning has mainly been used during the observation process.

3.1 Questions

3.1.1 What functionality should be included?

To answer this question The Chimney Pot's workflow, personal and current DAM system had to be investigated and evaluated. One experienced representative from each category of users was interviewed. The categories are artists, producers and clients. They had a good idea of how their system worked and what new functionality that was needed. An executive was selected to represent the clients. He had been working for a long time and knew what clients wanted. The clients are very many and their experiences and skills are very different. To get a general picture of a client many people must be interviewed. The results would probably be about the same as the answers from the executive. It was therefore decided not to interview any clients. The questions are found in Appendix 1. Later a colorist was interviewed. He also had specific needs (Appendix 1). The interviews were analyzed and the requirements were separated from the solution ideas (Preece et al, 2002). Then the requirements where sorted into five sub-requirements (Preece et al, 2002):

- **Functional requirement** are what the product should do, what functions are included.
- **Data requirements** regard technical information about files and background services.
- **Environmental requirements** define where the product will be used, if collaboration will occur, how support can be provided and on what hardware it will run and what other systems it need to be compatible with.
- **User requirements** define who the user is and what he intend to do.
- **Usability requirements** regard the usability of the system.

In order to widen the horizon the functionality of other companies also where investigated. The biggest media company in Sweden is the Swedish Television (SVT). They have been developing their system for many years and it's working very well. After talking to the development executive an expert on the functionality of their system was found and interviewed. Another interesting company Ducklings, one of Denmark's biggest postproduction companies, where examined. Ducklings had them selves developed a tracker for their company. The main developer of at Ducklings was interviewed. Another expert in Australia that also had been working with the development of DAM system was interviewed. Due to the large distance mostly e-mail was used.

After these interviews sufficient information about what the users wants and what is possible to do have been mapped.

3.1.2 How should the files be handled?

To answer this question more technical information was needed. At The Chimney Pot a system administrator and a scanner operator was interviewed. The scanner operator was interviewed because he is responsible for creating map structure and adding the media files into the system.
Drawbacks and requirements where collected for analyze. More data requirements where identified and added to the list of requirements.

Even here other companies was interviewed. At SVT two other experts was interviewed. One responsible for the backend system and one a responsible for their database were interviewed. The developer at Ducklings was also interviewed. Duckling’s file handling system could be a good roll model for The Chimney Pot. Also STOP (Stockholm Post Production) was interviewed in order to know how they handle their files. They are competitors The Chimney Pot and it was interesting to see how they worked. Here a technical expert that had a good overview of their systems where selected.

After this investigation an overview of the current situation, their wishes and how others solved the same problems was obtained.

3.1.3 Should the new system be bought, rented or built from scratch?

When the requirements were known the different solutions was evaluated. Different promising online standard systems where suggested by the developers at Ducklings. The recommended solutions where Alienbrain, Cerebro, Cozimo, Shotgun, Shotrunner and VFX Nexus. It was then decided if one of those should be bought, rented, combined or modified or if a new system should be created. Combinations could be made; one part of the system could be constructed from scratch and another part could be bought or rented. All these solutions was tested and evaluated. Trial versions where downloaded to all solutions. Their functionality was mapped and compared to the user requirements from The Chimney Pot. Simple user test with an artist was also conduced in order to capture his opinion. The possible solutions were then narrowed down to only a few options. A meting with an executive was then held in order to get their opinion.

3.1.4 How should a new system be designed?

First of all the theory behind the design process was studied and then a conceptual model was transformed from the user requirements. It was first an idea in the developer’s head how the system should work.

After that the developer had to find inspiration for the design process. During the evaluation of the online solution de developer got an over all idea of the design. The most interesting solutions were examined more closely. Cozimo’s drawing tool is unique and is very good for video collaboration, which is one of the most important requirements. The Shotgun is also great for inspiration with its elegant and flexible design. Also the interface at SVT and Ducklings where examined, but didn’t give much inspiration. Before the design process started an inspiration draft was received from the artist. Now the developer had a design frames to work within.

The design process is divided into three parts, menu, icons and screen design (Preece et al, 2002). First the screen design was developed. After that metaphors for the menus was found and the menus where design. Several menus where created in order to give the users more freedom. After that icons where designed and the different tools and object where groped after similarity in the different windows. During the whole design process the Nielsens heuristics was used as guidelines to avoid most common errors (Preece et al, 2002).

3.1.5 How is a prototype created?

Following the design method the first prototype was created. The first prototype was static in order to facilitate eventual changes. A simple image is easy to modify compared to an interactive version. With the requirements, ideas from online systems and the design draft a Photoshop image was created. The decision was taken to construct a high-fidelity static prototype from the to start with. (see 3.4.4). A meeting was held in order to get approval of the design. The static image was shown while the developer explained the system.
The next step was to create an interactive version. The developer chose to use Dreamweaver to do a simple prototype with some clickable links. To make it easier for the developer only a few links were created. One user test with the artist was done. The test was conducted as a "think aloud test". Users where given different tasks and were asked to talk aloud while solving them. At this stage the prototype was so instable so only one user test was enough. The results where evaluated and a revised prototype where created. This time different categories of users where selected for the user tests. It was a producer, colorist, 3D animator, artist and a system administrator. This process "Design - test - evaluate - redesign" continued until the users seemed satisfied (Preece et al, 2002). In order to avoid getting different opinions regarding the same problem, user tests with two users working together where conducted. One producer and one artist worked together to navigate the system.

### 3.2 Approach

The first step was to collect as much information as possible. First literature regarding Post Production, The Chimney Pot, Digital Asset Management, and Human Computer Interaction was read. It is important to get a good understanding of the industry and the companies. In order to get more specific information field studies to different postproduction companies where conducted.

In order to get the work done in an organized way a good approach method must be selected. Preece describes a method that contains four steps: (Preece et al, 2006)

- Defining needs and requirements
- Make alternative design proposals
- Create an interactive version
- Evaluate the result.

This model is a bit simple for this project. With Preece's model as base a new extended version in two parts have been formed and:

#### 3.2.1 Part 1: (Gathering information)

The objective for this part is to get an overview and an understanding of the industry, the companies and the people who work there. The objective is to get answers to question 1-3 above.

- **Understand the company**
  Things was defined where: infrastructure, workflow, programs used etc. This is important in order to understand the user context.

- **Identifying the users**
  Here it was observed how the users worked, what specific tools and programs they used and how do they used them. A user-centered approach was used.

- **Identify needs and specify requirements**
  There are Functional, Data, Environmental, User and Usability requirements (Preece et al, 2002). Users need and wishes and drawbacks in current systems where defined.

- **Investigate other companies**

- **Evaluate available DAM solutions**
  Online standard solutions have been tested, evaluated and compared with the requirements.

#### 3.2.2 Part 1: (Building a new system)

- **Design proposal**
  A design proposal was created and evaluated.

- **Interactive Prototype**
  An interactive prototype was created that meet all the four categories of requirements.
• **User test**
The interface was tested and evaluated by the users.

• **Redesign**
A new improved prototype have been designed, tested and evaluated. This procedure repeated itself until the company was satisfied with the result.

• **Write a report.**
An explanation of the system was written for a programmer. He should be able to understand how is should work and look like.

During the whole process the users where involved as much as possible. A user-centered approach was in focus. Experts where asked in order to solve certain problems.

### 3.3 Collection of information

There are mainly two roads to take when it comes to collecting information. The first one is quantitative methods and is mainly used to compare data, looking for relations and coming to a conclusion. The researcher use techniques that will give them quantifiable or even generalized conclusions. (Bell, 2006) Qualitative methods are more used to understand how people experience their worlds. Here an insight and understanding is wanted rather than statistical numbers. This study used qualitative methods. The aim is to attack a specific problem in a specific situation and come to a conclusion. This method is called action research. The thesis is meant to threat a limited aspect of a problem during a limited time. In other words this is called a case study (Bell, 2006).

#### 3.3.1 Different techniques

According to Preece there are mainly five different techniques of gathering information. It can be **Questionnaires, Interviews, Focus groups, Naturalistic observation** and **studding documentation**. For best result a combination of data gathering techniques should be used (Preece et al, 2002). Each technique gets information from a certain perspective. In this study al techniques except questionnaires have been used. **Questionnaires** are good for answering specific questions and are mainly used in quantitative studies. (Preece et al, 2002) The population that was investigated was quite small. The information obtained was complex and a dialogue was needed for better understanding.

**Interviews** can be structured, semi structured or unstructured. In structured interviews only have to mark the answers in the premade forms. It is on one hand important to give the stakeholder freedom to talk about what he thinks is relevant, but on the other hand also to have a certain structure in the interview. (Bell, 2006) In this thesis the interviews can be located somewhere between unstructured to semi structured. The stakeholders are experts in their area and know what's important. In order to give them more freedom unstructured interviews were selected.

**Focus groups/expert groups** and are good for collecting multiple viewpoints. In focus groups several people will discuss an issue. The users stimulate each other to make new observations (Swartling, 2004). It's not uncommon that users within the same organization have different views on things (Preece et al, 2002). Something that can be hard with single interviews is when different persons say different things. Whose opinion should be prioritized? In focus groups they can talk with each other and come to a single conclusion. Bell mentions that a common error is that the focus group often consists of people that are equally minded (Bell, 2006). In this report users representing different categories joined. The groups were used to try new ideas, and get feedback.

**Naturalistic observation** is good for defining the user context. The stakeholders are observed during a normal working day. Notes are taken and questions asked. (Preece et al, 2002) In this report most of the work where conducted at the company. This gave a good overview of how the different users worked and acted.
Studying documentation is god for learning about procedures, rules, standards etc. (Preece et al, 2002) This part was conducted in the beginning of the work in order go get a good understanding of the industry.

3.3.2 Data recording
There are several different ways of recording data during interviews and user tests. Here the different techniques are discussed and compared:

- Notes + digital camera
- Audio + digital camera
- Video

Taking notes is the simplest form. One problem is that it can be difficult to write and make the interview at the same time. The interviewed can also get bored if the writing takes too long time. This problem could be solved if there are two interviewers and one act as a secretary. (Preece et al, 2002) Recording audio is a good alternative to notes. It gives the interviewer more freedom and he doesn't have to focus on taking notes. Recording sound is also more discrete than video recording. Video recording has the advantage of recording images but is more intrusive. Both video and audio recording is very time consuming when it comes to transcribing (Writing it in text). To transcribe one hour of material will take about four ours (Bell, 2006). The advantages of video and audio recordings are that it gives a higher reliability (Preece et al, 2002).

In this report mainly video recording have been made. If the interviews are not recorded it is likely that some important information will be lost. During user test the video is crucial. It shows how the users act and react. Video recording have been made in order to get higher reliability, give more freedom to the interviewer and to be sure not to miss any information.

3.4 Evaluation methods
While evaluating a website there are several methods to choose from. These methods are designed for experts, not regular users. Either an inspection or a walkthrough can be made. An inspection needs a prototype but a walkthrough doesn't. (Preece et al, 2002)

3.4.1 Cognitive or pluralistic walkthrough
"Cognitive walkthroughs involve simulating a user's problem-solving process and each step in the human-computer dialog, checking to see if the user's goal and memory for actions can be assumed to lead to the next correct action." (Nielsen and Mac, 1994) In this process the user is not involved. The aim is to create a model of how the user thinks and acts the first time he use an interface. The evaluator describes a story of how the user step-by-step navigates through the interface. The goal is to simulate how the user will perform a task (Swartling, 2004).

"Pluralistic walkthroughs are another type of walkthrough in which users, developers and usability experts work together to step through a scenario, discussing usability issues associated with dialog elements involved in the scenario steps". (Nielsen and Mac, 1994)

3.4.2 Heretical inspection
The objective of heretical evaluation is to find usability problem in the interface design and later fix these problems. An expert goes through the interface while looking a list of guidelines, so called heuristics. The expert looks if the system compares it with the heuristics. Empirical studies have shown that five users can find 75 % of the errors (Preece et al, 2002) (Fig. 4). The reason for this is that different users find different problems. The user problems are then collected in a list, ordered after importance.
Nielsens heuristics are (Preece et al, 2002):
1. Visibility of system components
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Help users recognize, diagnose, and recover from errors
6. Error prevention
7. Recognition rather than recall
8. Flexibility and efficiency of use
9. Aesthetic and minimalistic design
10. Help and documentation

In this study only heretical evaluation and user tests have been used. The heretics were more used as a guide to create the proposal. The system is too complex to do walkthroughs. The designer himself and the experts don’t really know all the steps involved in navigating the system. A prototype was needed in order to concretize the system. Most focus was laid on the user tests. There are several potential users, but only one expert available.

### 3.4.3 User testing

In these test users are given tasks while their behaviors and reactions are observed and measured. It gives a good understanding how users work and interact. The tests can also been made on several versions of a system and the system doesn't have to be ready. (Swartling, 2004)

In this report the method "Think aloud" was used. Users were given different tasks and were asked to talk aloud while solving them. They should say what they were thinking, what they were trying to do, questions that came to their minds, what they read etc. While taking notes or recording it's important to link the user actions to what they say. (Lewis, 1994)

The tasks can be handed over in a written form, but in this study they were handed over orally in order to stimulate talking. In the beginning of the test the test leader explained the meaning of "think aloud" and underline the importance of talking all the time. The screen was filmed so that all that the user did was recorded. The films was looked at and analyzed. It is interesting to follow the mouse pointer. It can reveal the users cognitive way of thinking. This regards experimental cognition and it shows what the user automatically tries to do. The user himself is not always aware of all cognition problems.

Sometimes the user may sit silent, trying to understand the interface. He gets stuck in something that he doesn't understand. In this situation it is good to asked question like: Where do you think you are now? What do you think happen when you press this button? What happen now? Sometimes the users in this project needed some questions to keep going.
3.4.4 Prototypes

The idea of a prototype is to evolve the ideas about the system. The objective of a prototype is to answer questions. There can be two types of prototypes, low-fidelity and high-fidelity prototypes.

A low-fidelity prototype is very different from the final result. It can be made from another material, for example paper. The advantage of these prototypes is that they are quick, easy and cheap to make. The prototype is flexible and encourages modification. (Preece et al., 2002)

A high-fidelity prototype is more similar to the final result. They are normally built in software (if it's an software prototype.) They are normally more interactive and can give a more realistic image of the final system. It can be almost fully interactive and can clarify the navigational scheme. The disadvantages are that they take long time to build, users comments details rather than content, users don't want to criticize something that have taken long time to develop and a single bug can stop the testing. Another danger is that the developers can thing that the prototype works and then consider fewer alternatives. (Preece et al, 2002)

There can be two approaches while building high-fidelity prototypes. One is called evolutionary prototyping, which means that the prototype will evolve into the final product. The other one is called throwaway prototyping, which means that the prototype will only be used temporarily, and then be thrown away. In this project the later approach was used.

The design process continues in a loop: "Design - test - evaluate - redesign - ..." (Preece et al, 2002) until the company is satisfied with the solution.

3.5 Validity and reliability

High reliability means that a method or instrument can give the same result if it's repeated under the same circumstances (Regarding quantitative methods). (Bell, 2006). In this study the information recorded on video, which is more reliable than notes (Preece et al, 2002). The problem is that the users can easily change their mind. What people say cannot be trusted (Swartling, 2004). Today they might think something is a good idea, but tomorrow they think its bad. Another factor that's important is time. Technical information can easily get out of date (System Administrator, 2010).

Validity is a measure how good a question is to measure what it's supposed to measure (Bell, 2006). In this thesis interviews, observations and user tests have been made. Answers can be interpreted in different ways. If the answers are recorded on video the chase for misunderstanding will get lower. The interviewer will get more time to analyze what have been said. In a small qualitative study like this validity is not of high importance.

One problem arose when the different users had different ideas and thoughts about the new system. One user said do “this” and another user said do “that”. I order to solve this problem focus groups were held. Now the users could argue with each other about their different ideas. The interviewer didn’t have to be a communicative link between the users.

In order to broaden the perspective and increase the validity even more, employees from two other companies where interviewed in the same way. Their way of working where totally different and could hardly be compared to The Chimney Pot. The interviews gave an image of the diversity that exist between the different media companies.

Due to the small amount of people that has been interviewed (about 15 persons) it might be hard to generalize the information. The interviewer has also filtered everything that has been said by the users. He has mostly considered the things that he thinks are interesting. The information doesn't need to be generalized. It is gathered to give understanding of a certain company, its system and users.
3.6 Criticism of methods

If this would have been a general system this methods used would have given a to narrow view. Other companies and experts were questioned, but the weight was laid on the information obtained from The Chimney Pot. The intention was to do tracker suited for The Chimney Pot that later could be implemented by other postproduction companies. If the information is to narrow tracker may only fit one company.

Another important issue is that clients never were asked. There is an interface for clients, but no clients have been asked and no user tests have been done. An executive from The Chimney Pot represents the clients’ opinion from. He has much experience from clients and knows what they want. To get a good overview of what the clients wants, many clients must be asked. This was too much work for this thesis. The result would probably not differ so much from the responses from the executive anyway. The executive has good technical knowledge and have very easy to understand most functions in the system. A higher validity would be gained if the clients had been questioned.

No questionnaires where sent out. In a later stage a questionnaire could have been sent out to the employees within The Chimney Pot concern and other postproduction companies. The results of the interviews and user tests could have been verified, falsified or modified. The questionnaires could just describe the new functionality of the new systems and the users could agree or not agree.

Some disadvantage with the user tests is that they are in a controlled environment and that the test persons might be too critical (Swartling, 2004). Another problem is that the test leader always must act neutrally. It is very hard not say too much. Just the tone of voice can reveal test leader. It seems like the user is tricked to go into a set trap. Test leader not explain and defend bad design and say, "Try again". All the users ideas should be regarded as great. (Ljung, 2006) Though it is very hard not to help the user. If he totally misunderstands something it might result in that he doesn't understand or misinterpret other functions.
4 Result

First a technical description of The Chimney Pot, their users and requirements are presented. Then other postproduction companies and online standard solutions are described. Finally the development process of a new system is described. All results from the interviews, observations, evaluations, prototypes and user tests are described in detail.

4.1 The Chimney Pot

In order to understand the users and their context the company have been carefully investigated. The workflow has been examined in order to understand what the users do and what they expect from the new system. It is essential to understand the old workflow in order to develop a new. The infrastructure has been investigated because it sets the technical limitations of the tracker. It's important to know what it's possible or not. Bottlenecks and drawbacks have also been identified. The current DAM systems have also been carefully investigated. What the systems can do and how the users work with them can provide most of the requirements for the new system. Their strong sides was transferred to the new system, while the drawbacks was eliminated.

4.1.1 Current workflow

![Figure 5. The Chimney Pot workflow (The Chimney Pot, 2009)](image)

Projects are first divided into commercial or feature film. The main parts of the workflow are:

**Preproduction and production**
- The work starts and the project is created in Farmers Wife.
- Ideas and storyboards are discussed
- Shooting with a VFX specialist from Chimney on location

**Postproduction**
- Films, tapes, off-line, EDL and description is delivered from the client
- Scanning, capturing and conforming of the original material
- Color grading
- Online editing: VFX, CGI and compositing (parallel with the color grading)
- Sound editing (parallel)
- Mastering
- Archiving
Preproduction and production

If The Chimney Pot enters the process before anything has been filmed the parties start to exchange ideas and send different attachments to each other. It can be links, images, QuickTime movies etc. These files are normally sent by e-mail. The attached files are not stored in any particular folder and the files might therefore be hard to find. When The Chimney Pot and the client have reached an agreement the shots are filmed. Normally a supervisor from Chimney is present at the shooting in order to avoid unexpected surprises.

Postproduction

An offline, EDL and the high-resolution material are then sent to the postproduction company. They normally have the following formats:

- Original tapes (HDSR, Digibeta etc.)
- Film reels (16 mm or 35 mm)
- Hard drives (RED-material)

Scanning

The film reels are scanned with the ARRI scanner, tapes are digitized and RED-material conformed. Everything is conformed to DPX-sequences and stored centrally on the SAN. The ARRI scanner is controlled by software that reads an EDL and stores the DPX-sequences in a predefined map structure.

Color grading

The color graders don’t download the shots locally; instead they work directly from the server through a fiber connection. For faster access the shots are caches locally. Before and after the material is graded it’s rendered locally and recorded to HD-SR tape in source order. The clips are then imported to editing systems and conformed to its delivery format. The network is not used for file transfer because it is to slow. It’s faster to record to tape in real time and ingest than to use the network. The network speed is about 10 % of real time transfer. During grading the colorist and the client often finds details that must be fixed. He then puts post-it notes with time code and what should be done. These post-its are then given to the producer that decides if new tasks should be designed.

VFX - special effects

The grading goes on parallel with the VFX. The artists prefer to work on graded material, but most of the time the production is so tight that this is not possible. If graded material is used the artist will see exactly how it will look like in the end. In order to know what will happen in the grading the editors use an estimated grading called LUT (Look Up Table). It doesn’t work perfectly, but is the best solution for the moment. Today the editors work on the shots locally. The commercials are mostly made in the Flame and Smoke while the feature films are made in several different machines like the Film Master, Nuke, Shake and more. The graphics are made parallel and added in the end.

Delivery

When all shots are approved the films are conformed by the Nucoda Data conformer, Nucoda Film Master or Final Cut Pro to its final format. They are then exported to film or master tapes (DigiBeta, HDSR etc.). A “digital package” with a DVD and ten different digital formats is also attached. (Gomez, 2008)

The project needs to be kept a couple of month before it’s archived and deleted. The client sometimes comes with changes after the project is done. Today the entire project map dumped to data tape, regardless of it’s content.
4.1.2 Infrastructure

In total they have about 30 workstations/servers with 1 - 10 TB local storage. The two color-grading systems and the assistant each have 7 TB of storage. Their most important machines:

- Online editing: Smoke, Flame, Flint, Final Cut Pro
- Color grading: 2 Nucoda Film Master, Grading assistant
- Conform/scanner: Scanner, Data conformer assist, Machine rooms capture system
- CGI department: 2 Nuke, Maya and Softimage
- Film recording: ARRI Laser
- Other: Tape robot (32 tapes of 400 GB), IQ – data restoration system

Network

All the files are stored centrally on two SAN and the X2. One SAN volume contains a mixture of shots and the other one only scans. The X2 contains the following:

- The sound editing systems
- Dropbox (deleted after two days)
- Projects with links to the SAN
- Users with user data
- Autodesk (Flame, Flint and Smoke)

There are five to six other storage facilities outside the SAN connected as fiber clients. They are the Film Masters, scanner, scanner assistant, backup system and the SAN gateway. All other workstations use the gateway to reach the SAN.

The fiber network is about 2 Gb/s while the network is 1 Gb. The Stockholm Internet connection is 100 Mb/s. The connection to Ducklings is very good (30 Mb/s), to Warsaw it's good but to Kiev and Dubai it’s quite slow. They plan to use a Data Acceleration software with will speed up the connection between the offices.

They have a user database (LDAP) on X2 and everyone who wants' access to the network must have a password.

4.1.3 Current DAM systems

The company mainly has three systems for handling projects and film clips. One system for project information such as booking and economy, one to handle tasks and information regarding VFX shots and one for client approval. The new system is meant to improve and replace the two later ones. The most important factor in order to create a new system is to exactly understand what these systems are capable of.

Farmers’ wife

Farmers’ wife is mainly a program for booking but can also store info about master tapes. Information about all masters in the tape archive and thousands low-resolution version can also be found. This program works well and they intend to continue using it. Their project data is currently saved in the Farmers’ wife database. It can be information like: production name, producer, client, artists, director and their e-mail and telephone number. This information can be reached from an application called "Barn". This data could be used by the new system.

They use the Farmers’ Wife database but have a sequel mirror locally. Data is added through Farmers and their database is also updated. It is possible to write back to the Farmers database, but it's not widely supported.
FTP Tracker

In their current tracker is used in feature films and complicated commercials (1 of 20). In every project there is a long list of all the shots (Figure 7). All information has to be added manually. All shots can be assigned to different artists. The shot can have different status, so the users can easily see where in the process it is. Users can attach files and write comments. The program is appreciated by the users it facilitates their work, but it have many deficiencies.

After interviews with an artist, a producer and an executive the following drawbacks where established:

- The program is described as ugly and not user friendly.
- Everyone can see and delete everything.
- Too much irrelevant information is showed at the same time. There are a many empty and unnecessary fields in the shot tables and bottom of the page.
- It is too complicated to change and add information. The user has to enter many submenus, which leads to too much unnecessary clicking. To write comments, change deadlines, approve shots and add remove users and machines becomes too hard. Today they have persons and machines that don’t exist because it's too complicated to change.
- The page has to be refreshed manually by pressing F5.
- Files and thumbnails need to be uploaded manually.
- There is no support for DPX.
- All attachments in a project are put on the same place - hard to find.

The blog

The blog was created to for clients to approve shots (Fig. 8). The producer or artist loads a clip on the blog. Then the client can go in and play the clip and write comments. The artist can respond and upload a new version. Versioning is supported by the blog. There is a function that enables the user to filter between the different shot and versions. All updates are mailed (HTML) to the users every day at 16.00.
After an interview with an executive the following drawbacks where established:

- Navigating requires too much unnecessary clicking.
- User handling is complicated.
- It's too hard to jump between shots.

### 4.2 File handling

In order to create the new file handling system it's essential to know how the old system works. They are not happy with the way the files are handled today. The most important task is to define drawbacks. There are two different maps created for each project. One map contains scanned films one for low-resolution files, graphics, storyboard etc.

**Project files**

File handling is mostly handled manually. The producers have script that creates a file structure with 15 empty maps. The users then put the material in these maps. When things are sent to archive the whole project map is dragged and dropped on a tape. They have a tape robot with 32 tapes. 60 more tapes stand in the shelf.

**Scans**

The ARRI scanner is controlled by software that reads an EDL and creates a predefined map structure on the SAN. Scans are put in three different maps: one for feature films, one for commercials and one for RED material:

- PROJECTS/PROJECTNAME/
- PROJECTS/FILM/PROJECTNAME/
- PROJECT_RED/PROJECTNAME
They have two ways of organizing the DPX-sequences. The first is used when material is sent to other agencies. The second is used for in-house productions. It’s easier to find a certain DPX from the editing stations.

- PROJECTNAME/REEL/FRAME.DPX (standard)
- PROJECTNAME/REEL_STARTFRAME/FRAME.DPX (ex 101_96788)

After an interview with an system administrator the following drawbacks where established:

- Anyone could go in and create maps and delete files almost everywhere.
- No organized file handling exists on the network. The users have a hard time finding the files they are looking for. Users have a tendency to put everything in the same map. There are many empty folders. Much unnecessary information goes to archive.
- The trashcan is used wrongly. Users through stuff there and says: "It's in the trashcan".
- No versioning exists.
- No metadata exists on the network. Only codes and names are found, which don’t tell the user anything.
- Double work when clients add project info into word document and storyboard in a pdf file sends to the company. This information then has to be entered again to the tracker.
- Two standards for scanned material

### 4.3 Requirements for the tracker

Most of the time was spent collecting information to establish all requirements for the new system. The requirements are the foundation to the new system. From them the new functions, file handling and some design issues can be established. In other words they are the key elements to answer most of the main questions in this report. The different requirements organized into the categories five categories (Preece et al 2002).

#### 4.3.1 User requirements

There are mainly three different categories of users that will use the tracker. They are artists, producers and clients. All of them have different kind of needs and experiences.

The producer is responsible for the whole project and the contact with the client. In the tracker he needs to get a good overview over everything in a project. He easily wants to see the updates since he was last logged in. He should be able to create projects, add shots and assign tasks to artists. When a shot is finished the producer wants to see it and approve it before it's sent to the client for approval. He normally has many things to do and needs everything to go fast and only see the relevant information.

The client (one per project) needs a simple and clean interface. Many clients don’t have so high technical skills. They should be able to see shots in full screen, approve them and write comments. They should easily be able to click between shots. In the beginning of a project they should also be able to upload storyboard and other project related information into the tracker.

The artist will get his tasks through the tracker. He wants to be notified when he have a new task and easily see what he needs to do. He should be able to download and upload DPX-sequences into the tracker. He needs a path to the DPX-sequence to be able to edit centrally stored files in his editing program. He should be able to send both finished and not finished shots to the producer or client for approval. Many of the artists use graphical tablets and want to use them to navigate the tracker. This means that the button should be big and clear.

#### 4.3.2 Functional requirements

Functional requirement are what the product should do, what functions are included. They where established after interviews with artist, producer, executive and system administrator.

- It should be possible to make comments on every frame by drawing and writing text. The duration of the comment should be changeable.
• Different users should have different rights. They should be connected to the projects they are working on. Everyone should not see everything.
• Thumbnails should be generated automatically when a new file is uploaded or added on the server. FFMPEG could be used.
• QuickTime’s should automatically be generated when DPX-sequences are uploaded.
• Shots should be able to have status (not started, started, done, in revision).
• All recent updates should be sent by e-mails 16:00 every day. Only important information such as new, done, or approved shots.
• Two month after project is done the responsible user should get e-mail that the project is going to archive. Only the latest version of all clips should be predefined. They have 2 days to select other clips. If archived material is needed the tracker sends notification what tape that should be loaded. Both "Archive" and "Archive and delete" should be selectable.
• Versions should be handled and the latest version selected by default.
• It should be possible to assign tasks to users. Subtasks should also exist.
• It should be possible to upload different kind of attachment like QuickTime movies (hidden by default).
• Shots should be able to be played in full screen.
• It should be easy to jump between shots.
• There should be different tracks for films, effects and elements. The user should be able to see all shots or only one.
• The user should see the path to the high-resolution material, its duration and whether it's online or not. The VFX field should have some predefined chooses and free text input

4.3.3 Environmental requirements
Environmental requirements defines where the product shall be used, if collaboration will occur, how support can be provided and on what hardware the program will run and what other systems it need to be compatible with. These requirements were established after interviews with executive and system administrator.
• The tracker should first be used in Stockholm and later be implemented in Kiev.
• Users need to collaborate on shots and write comments
• The application should run on a Linux server.
• Data should be stored and read from Mac OS, Windows and Unix.
• It should work in Internet Explorer, Safari, Firefox on both PC and Mac
• Big files (up to 1 Tb) should be transferred between offices (WPN or FTP). When something is downloaded it will first get on the local server. Low-resolution material should be mirrored to all offices.

4.3.4 Usability requirements
These regard the usability of the system. These requirements are very important. If they are not obtained the user might start working around the system (Jacobsen 2005). They are the foundation of all other requirements.
• The interface should be user-friendly, clean, simple and unimportant information hidden.
• People with less technical knowledge should understand it.
• Everything should go fast to do.
• It should be easy to add and remove columns and change the data in the fields. The user should be able to change the information directly without any submenus. It should be easy to add and remove users and machines.
• Everything should be updated automatically.
• It should not look data like and film terminology should be used.
• To see details of a shot the row could get wider or a new window could open.

4.3.5 Data requirements
Data requirements mainly regard technical information about files and background services.
The tracker should create the file structure for projects, shots and version. There should be predefined storage places for different media. Users should not be able to create own maps outside the tracker. Artist should be able to render directly to a certain map. A new map V01, V02 should be created for each new DPX-version. When the shot status is done it should be locked.

- Huge files should be able to be moved or kept locally (on grading systems).
- The user should be warned when the disk is 90 % full.
- DPX, QuickTime, different images and pdf-files should be supported.
- The filenames could be created from the map name. (Maybe E_ = elements, A_ = scans and _v01 in the end) The names for the shots should fixed master time code.
- A special drive should be designated for the trashcan. The files should be deleted after 2 days. It should also be able to do hard delete.
- It should be the same login everywhere.

### 4.3.6 Other wishes
They want to see the shots on a timeline, have deadlines, import data from Farmers (project name, client, coordinator, producer, director, editors and their e-mail and phone number), pdf-export and undo function. The artist also wants the tracker for browsing the effect library (with generated thumbnails and QuickTime’s).

### 4.4 Other companies
Three other companies except The Chimney Pot were also investigated. They were investigated to give this report a broader view and to gain inspiration and new ideas for a new system. The Swedish Television was selected because they have a very advanced DAM system. Their system can serve as the perfect model. Ducklings are The Chimney Pots counterpart in Denmark. They have developed their tracker by them selves and have a very good file handling system. STOPP (Stockholm Post Production) was chosen because they are one of The Chimney Pots competitors in Sweden.

#### 4.4.1 The Swedish Television (SVT)
SVT’s policy is that everyone should have access to everything. Users are able to see and add what they want, but it’s hard to delete things. Their central computer is called "Centralen" and files there will automatically be archived. Unused files will be removed after 30 days. Files on local servers named in a certain way will automatically be transferred to "Centralen". Low-resolution MPEG-files and thumbnails are generated for all clips. A new thumbnail will be generated for each image change (when data rate is high).

They have about 500 servers all over Sweden and two 180 Tb SAN (about 6 mount of coverage) in Stockholm. Low-resolution material is stored forever on a 50 TB server. All metadata is stored in a huge database. They have a disaster recovery site that mirrors everything. They have daemons running on 16 different servers. Their standard file format is DVCPRO50 or 25.

Filenames are generated automatically in the following way:

```
PG-1127978-003A-TESTA2009 =PROGRAM-ID-EPISODVERSION-PROJECTNAME
```

Editors must use this filename. They copy the name to their editing program and post the clip. The clip will then be found by the system. All files have unique IDs. When a new version is made it has to be posted with the same filename and -01, -02, -03 will be added automatically. The latest version is always selected, but older version can be selected. Only the sent version will be archived.

An application called Precut that enables the user to edit existing clips and create new. The user can also drag and drop a file to avid and a new project will be opened. It's handled with XML import and export. They also have a special page where background transfers can bee seen.
4.4.2 Stockholm Post Production (STOPP)

STOPP is slightly smaller than The Chimney Pot and have less automated workflow. Artists at STOPP receive their tasks early in production. They are often present during production and know what to do. After shooting an editor together with the director make off-line version. After that the material is digitized.

On their network they have a script that automatically creates the file structure. Everyone has to put everything in its correct folder. Only projects, 3D-stuff, sound, EDLs and graphics are stored on the server, no high-resolution clips. To transfer files internally they put them in a folder called "transfer" and notify their colleges.

The client normally approves the shots in a suite with high-resolution monitors. QuickTime files are also sent through FTP for approval. The clients normally call after they have seen the shot. When a project is finished it is archived to 400 GB tapes that are inserted manually. Only stuff that cannot be rendered is archived.

4.4.3 Ducklings

Ducklings DAM system handles high-resolution files that can be play in real-time 2K. It's a local application and is not used outside the company. The tracker creates all folders on the server. The user cannot create own folders or move files. The files are stored on different locations depending on what it is. They use mounted drives so if they run out of space they just put in another disk. When a new asset is loaded a new map is created. They only have two folders per project: graphics audio. The shots can have several jobs that are assigned to different users.

All new projects must be imported from Farmers. In a project there are normally 2-5 sequences where the different assets are stored. Under shot all assets and their tasks and status (Red, yellow, green) is shown. The user can get the path by opening a new window and copy it. They have versioning and any version can be set as active. Thumbnails are rendered automatically.

They have archiving function that collects all the assets and sends them to archive. When something is moved to trash all siblings becomes illegal. The file structure remains but the files have been moved.

They have the made a function for conforming DPX-sequences to QuickTime but it's not implemented yet.

4.4.4 Summary

The Swedish television have a very good system to transfer files between offices, archive etc. They have smart naming functions for filenames and folders. Their system is very good, but to advanced for The Chimney Pot. STOPP is not interesting because they have no DAM implemented. Duckling is about the same size as The Chimney Pot and their way of handling files are interesting for The Chimney Pot. The design is simple but not too attractive.

4.5 Online standard solution

Here information was collected in order to respond the question "Should the system be bought, rented or built from scratch?" One of the developers of Ducklings successive DAM system recommended: Cerebro, Cozimo, Shotgun, Shotrunner and VFX Nexus. Alienbrain was recommended by one of the artists. The first process was to register to evaluate all the different solutions.

The intention was to carefully test and evaluate all the online solutions. All system where tested together with an artist. Design and functionality was tested and evaluated. The mayor drawback with all the online solutions is that they cannot handle the file system and the high-resolution files on the SAN network. Everything has to be sent over the Internet. Only one of the solutions also has the required function to draw and make comments directly on the films.
Avid Alienbrain
- DPX and support for many other files
- Advanced versioning
- Archiving
- Drawing, but not on films
- File handling
- Automatic thumbnail generation

Alienbrain is the most widely DAM-solution used for media projects (Alienbrain, 2010). It has a robust back-end management and handles versioning efficiently. It has support for many files and supports collaboration of high-resolution files. It looks good seems simple and logical. It seems that they have taken inspiration from a simple Windows explorer. They have functions for adding comments and drawing on pictures, but not film clips. It's software, not a web based application.

Drawbacks: Software, no flash player

Cerebro
- Task management
- Paint and leave notes on pictures and other attachment
- Software, but no installation needed
- Message handling
- Supports DPX, QuickTime and many other formats
- User management
- Automatic creation of thumbnails
- Attach files and directories
- Automatic transfer between offices
- Excel export

Cerebro looks more like their system. It becomes cleaner with gray color. It seems too complicated and it's not a web application.

Drawbacks: Complicated, software (local installation), no flash player, no file handling,

Cozimo

Figure 9. The Cozimo flash player (Cozimo, 2010)
- See recent activities and history
- User management
- Supports PDF and most image and video formats
Video collaboration in Postproduction

- Draw and leave notes on films and images
- Chat function
- E-mail notification

Cozimo have a very interesting tool for video collaboration (Fig. 9). I let the user draw and make comments directly on films, images and pdf. In general the site looks clean, but there are too many tools. It is meant to suit everyone, but suits no one (Swartling, 2006). It's hard get an overview over all shots. A simple user test was done.

Mayor drawbacks: No status on shots and no file handling.

Minor drawbacks: Too many tools in the menu. Drawn objects cannot be changed. Buttons are misinterpreted. All actions cannot be undone. The duration of comments cannot be changed. The player can only jump between every 5:th frame, which is not sufficient if explosions should be commented.

Shotgun

- Personal pages
- Task management
- E-mail notifications
- Status of shots
- Attachments
- User management (Account with LDAP)
- Flexible interface (move, add, remove columns and customize)
- Open Python API
- Secure encryption
- Local and remote backup

This application looks very good. It has many of the function that chimney wants. The problem is that it has too many functions. Columns can be added, removed and moved. It has "Mac"-inspired design (Fig. 10). It seems like the best solution so far.

Drawbacks: To complex, no Paint tool

Shotrunner

- Assets divided by sequences, shots and elements
- Task management
- Forum for comments
- Attach files
- E-mail notification
- Tracking pre-production
- Project homepages
Well organized and have a SQL database. Users must have some experience in order to navigate the system. While clicking new frames are loaded all the time. The user has to jump back and forward too much. Too much unnecessary clicking is needed. May parts are kept together but in general it looks to complex.

*Drawbacks:* No drawing tool, no file handling, complicated, static interface.

**VFX Nexus**
- Personal pages
- User administration (rights)
- Powerful search tool
- Attach files
- Task management
- Mini blog for each shot
- Calendar

This application looks simple and clean. It has many good search options but it becomes to complicate. On some pages too much information is listed. It has a little bit of a data look.

*Drawbacks:* No file handling, no paint tool, interface not flexible

### 4.5.1 Summary

The standard solutions that could be used are Cozimo, Alienbrain or shotgun. The other online solutions were not regarded any more. Of these three the Cozimo is the most interesting. It doesn't have all the specified requirements, but could be combined with another solution. The alternatives where: these three systems alone, in combination with each other or a new system. This is discussed more in more detail in the next chapter.
4.6 Design proposal

4.6.1 How should a new system be designed?

The new system was designed with the requirements as a solid base. Inspiration for the graphical layout was gained from The Chimney Pots current systems, online solutions, other companies system and the design sketch from the artist. The systems chosen as roll models was Cozimos flashplayer, Shotguns flexible and elegant design, and file handling functions from Ducking and table structure from Chimneys FTP tracker. The design sketch from the artist also got a central roll. During the whole design process Nielsens heuristics was used as guidelines to avoid most common errors (Preece et al., 2002).

Screen design

A common design for the screen was selected. It had a main frame and a top and left menu (Fig. 11). It is a standard design that should be understood by most users. The upper and left frame should contain menus. It should be links to all functionality in the tracker. The tracker parts were placed in the main frame. It should be (according to requirements): A film to draw and make comments on, thumbnails of shots, comments, shot info etc. The flash player has a central roll and was therefore placed centrally. Thumbnails are placed to the left in most tracker solutions (SVT, FTP Tracker, Shotgun... etc.). Other metadata was then grouped after similarity in different blocks. The sizes and placements of the blocks depended how important the functions are and the esthetical aspect. The design process is divided into three parts, menu, icons and screen design (Preece et al., 2002).

Menu design

The next task was to design a menu. Then several menus where created in order to give the users more freedom. Several metaphors where used for the menus. The upper menu has a tab menu (Fig. 11. 1). In the first try other functions where added to the tab menu. In order to facilitate navigating a menu was built into the tabs. Each tab now had three functions: list mode, open mode and full screen. The users only had to click on the different parts of the tabs to maximize and minimize. The left menu was based on the Microsoft Explorer (Fig 11. 2). The user only had to click a project, film or shot to see it's content. A small button menu was also added at he bottom of the big frame. In this menu the user should be able to add shots, approve shots in the current window he was.

Icon design.

All icons and button where designed from scratch. In order to make it more clear to the user common keyboard symbols from the font Wingdings where used as base. Those symbols exist on all computers and are standardized. Some new buttons was also created to give a more elegant design.

4.6.2 How is a prototype created?

Only high fidelity prototypes where created. The tools used were Adobe Photoshop and Dreamweaver. First a static image was created and evaluated. The next step was to make the first interactive prototype.
4.6.3 Prototype 1 - Static

According to the design approach described, a prototype was created in Photoshop. The first prototype was a high-fidelity static prototype. A simple image is easy to modify compared to an interactive version. With the requirements, ideas from systems, the design process above the first prototype was created.

![Prototype 1 - static](image)

Figure 12. Prototype 1 - static

One artist and one executive were shown the first prototype. Because it was only a static prototype the developer showed the prototype while he explained how it should work. They thought the interface should be cleaner with fewer tables. It should be as little as possible. They didn't always understand the buttons. They said the important data for shots is reel, original time code, download and upload function, one responsible, machine, deadline, hours and if DPX exists. Project information is seldom viewed. Project doesn’t need descriptions and films only a short description. Elements only need a description. They didn't like that the status jumps between colors.

New requirements/wishes discovered:
- While clicking archive or delete a list of checkboxes of all files should open.
- A page called "my tasks" showing all the user's current jobs.
- Artists should be able to send "work in progress" to clients or producers.
- Transfer status must be shown while transferring files.
- A page called "updates" where the user can see all recent actives.

4.6.4 Prototype 2 - Limited interaction

After the last prototype was updated an interactive version was created in Dreamweaver. In total 10-15 pages were added. The prototype was simple in order to facilitate the development process for the developer. One user test with the artist was done.
User test 1.1: Artist

A "talk aloud" test where held with a technically experienced artist. The artists where told to explain everything he thinks or do.

He thought the upper menu was very strange and didn't understand that each tab had modes: list mode, open and full screen. He pressed "new project" and didn't understand what happened. It was an error in the prototype and it said "shot" instead of "project" in the tap. He had to be shown how to create a project. When he clicked on the project he didn't understand if he was in project or just look at it. He wanted to import shots and looked desperately in the left menu, but didn't know what button to press. Then he tried to add a film, but that link wasn't clickable. He didn't understand the lower menu at all. He tries to look at an older version, but wasn't clickable. He doesn't understand the download buttons or "delete project" button. It should be an administrator thing to delete things. The user shouldn't be able to do it by mistake. When he pressed the archive button it said select files to delete instead of archive.

Evaluation

Unfortunately the user almost didn't understand anything. He thought all menus where confusing and illogical and buttons and symbols where hard to understand. He didn't understand where in the hierarchy he was.

The prototype itself had many deficiencies. There where many functions and links that where only graphical, but should be clickable in the real system. This lead to that most of the links where not clickable. This confused the user even more, when he didn't know where to click. Some of the links was incorrect and this halted the test completely and created more confusion. The menus must be clearer. Right now it's three menus: the top, left and lower ones. Buttons must be easier to understand. It must be easier to create projects, films and shots.

The user test also didn't have a clear structure. The tasks for the user to perform during the test weren't clear. Most users first wants to see what has happened and if they have any new tasks. All users writes comments and upload attachments. Some user specific tasks where defined.

- Artists (Create films, shots and elements, upload & download shot)
- Producers (See project status, create projects, films, shots and elements, assign tasks, approve shots)
- Clients (approve shots)
In order to do more user tests the following things were changed:

- Incorrect links were fixed
- Functionality from the tabs was removed (max 2 functions per tap)
- Lower menu removed and the buttons inserted in the windows
- Many more links and new pages were created (ca x3)
- Tables rearranged and modified
- Path to shots added at the bottom
- Status indication redesigned
- Many other details changed

At this stage the prototype was so instable so only one user test was enough.

### 4.6.5 Prototype 3 - Extended interaction

The results were evaluated and a revised prototype was created. Now there were much more clickable links. The designer wanted to give the user more freedom while clicking around in the system. This time different categories of users were selected for the user tests. It was a producer, colorist, 3D animator, artist and a system administrator.

**Figure 14 - Prototype 3 - Extended interaction - Project view**

**User test 2.1 - Producer 2**

The producer first thought project are shots. He tried to select the project he already was in, and didn't understand why nothing happened. In the left menu he clicks on film and shots comes up, he gets confused (Fig. 14). After a while he realizes the structure. He creates a project and realizes that he can import data and clicks. Then he wants to go back, but he cannot. A back button must be added.

He doesn't know how to change status on shots. He understands "my pages" with updates and tasks. He didn't understand that most things could be done in three ways. He thinks the thumbnails in the project list are confusing. He doesn't understand download to server or locally.

**User test 2.2 - 3D graphic**

He doesn't know what is clickable or not and can't find the buttons. When a button is pressed he wants it to change color. He doesn't understand what delete project means. He thinks he can delete the films under the buttons. First he doesn't understand the two modes in tap. He clicks on
Video collaboration in Postproduction

a project and expects to see the films. He doesn't understand the connection between open project and the film list.

User test 2.3 - Artist
He still doesn't understand the upper menu and thinks some buttons are strange. Thinks the films in project are the off-line versions. He doesn't understand what project status 5/15 means (5/15 shots done). He was in project mode and thought he was in shot mode. He doesn't understand "download project". It's much confusion between elements, shot, film and versions. He doesn't know how to change shot status.

User test 2.4 - Colorist
He doesn't understand the difference between project and film or the left menu. He thinks it would be good to be able to hide all tools in the flash player. He wants to be able to change the shot status manually.

User test 2.5 - System administrator (heretical test)
He doesn't understand the project status "10/15". He doesn’t like the terminology. He thinks that other tabs shouldn't be selectable before a project is selected. He also thought the films in the project view was off-lines. In the film view he clicks on the shots to the right and gets surprised when he comes to the shots. He doesn’t understand "Add project" and "Create new project". Under elements there are versions and elements that is confusing. The user should get an indication of what will happens when he click on things.

Analysis
The mayor problem was still the upper and left menu. It was still hard for the users to see where in the hierarchy they where. They didn't know if it was project, film, shot or elements. The project thumbnails were confusing. Everyone thought they were shots. Many of the buttons were still not so clear.

A big discovery was finally made. No one was using the left menu because no one understood it. It was not logical. It should have worked like the Windows Explorer; a map is opened and it's content is shown. The cognitive model was unfortunately incorrect (See 4.6.1).

The following changes were made:
  • The logic in the left menu changed - "you see what you click"
• Upper menu redesigned - they all got different symbols
• Thumbnails have different frames if they are project, film, shot or elements
• All tables were organized and cleaned up
• Buttons and status bars were redesigned

During the user tests new requirements or wishes where discovered:
• See the clip before and after the effect shot in the film view
• Zoom the timeline in the flashplayer
• Insert a new shot in a film so that the other time codes changes
• Create tasks in the film view in the flashplayer.
• Attach sound file and hear audio while watching the movies. Be able to select in-point on the sound in order to sync with video.

4.6.6 Prototype 4 - Extended interaction 2

User test 3.1 - Artist and Producer 1
The different several times had different opinions about functions or design. To straighten this out it was deseased to have a user test with two concurrent users.

The artist still doesn't understand the two functions in the tabs (Fig. 16). It's too much information in tabs. In the open project they still thinks the thumbnails belongs to the selected project, not other projects. They therefore cannot find other projects. They think that comments and shot info should be related to version. There was no direct link to the shot list. It was still confusion between film, shot and element. This must be clear. Only shot and comment should be shown, the rest hidden by default. They think it's annoying to be able to select a project in order to click on the other tabs. The producer tries to open a shot by double-clicking but fails.

Client representative - overall view
He thinks the interface is gray and boring. It looks like high-tech data and needs more color. The left menu looks too complicated. As little as possible should be shown. He doesn't understand what project status 67 % means. The left menu is still confusing. It is not clear that the thumbnails in the open shot view are other shots other shots. He doesn't understand how to download DPX and want path to use the path. He wants to skip the thumbnails for the project list. The shot information is still too complex. The clients will get scared. It must be scaled down further. The design must look more like "Apple".
New requirements/wishes discovered:
- Able to deactivate a project
- Select between small and large thumbnails in the list mode
- Add "go to end" and "go to start" buttons in the flash player
- Warning if shots with wrong duration are uploaded. If the time code is changed the time code of the other shots should also be changed.
- The frame rate, 24- or 25 f/s, must be defined for each project.

4.6.7 Prototype 5 - Final prototype

The users were satisfied with the overall design (Fig. 17). They wanted it to be even simpler for the client. The design should be more Mac-like, brighter and with more white and green. If felt like the design was done.

New requirements/wishes discovered:
- Help function needed
- Be able to open shot in Nuke
- Search function to find specific files
- Head and tails should be showed in element mode in the flashplayer. They should be grayed out on the timeline and not played.
- Import and export shots to the Autodesk systems Flame, Smoke and flint. Either with soft import or background import.

4.7 Summary

We have now learned how the company works, what systems they have and what their requirements are. We also know how other companies have solved their problems. The different online standard solutions have been investigated and evaluated. A new prototype for a new system has also been developed. During the development procedure several other requirements and wishes have been found. It was now up to the designer to decide which of these wishes that should be implemented. This is discussed further in the next chapter (Discussion). After the final prototype other requirements and changes have arise. Because the prototype has to be redesigned anyway it seemed pointless in updating it.
5 Discussion

In this chapter all the questions from the introduction and method parts are discussed in detail. Possible solutions are presented and problems and drawbacks are commented and analyzed.

5.1 What functionality should be included?

This question builds the base for the other questions. During the process many different wishes and new requirements popped up. In order to locate the most important functions all requirement and wishes have been analyzed. It is very important that requirements are separated from wishes and other ideas. If not the list of requirements can become very large. The users and developers will always find out new things that would be good and useful to implement. The problem with this approach is that the most essential requirements might get lost: User-friendly, simple and easy.

The most important tip is not to aim to high. The aim shouldn't be to build the best system ever. It's enough to build a small working pilot for in-house testing. (Austerberry, 2006)

5.1.1 Hierarchical task analysis

In this kind of analysis the tasks are broken down in subtasks. These subtasks are in then broken down in sub-subtasks etc. This will give an overview of what the users will do in the system. (Preece et al, 2002). The most essential part of the requirements is those that are needed for the user to perform their tasks. Those key tasks will be the base to the other requirements.

5.1.2 Prioritizing requirements

Needed for the users to be able so perform their tasks: (essential)

- User-friendly, clean, simple and most information should be hidden.
- Everything should go fast to do
- It should work in Internet Explorer, Safari, Firefox and on both PC and Mac
- People with less technical knowledge should be able to use it
- It should be possible to assign tasks to users
- Upload attachments (and thumbnails)
- DPX, QuickTime, different images and pdf-files should be supported.
Video collaboration in Postproduction

- Path to the high-resolution material
- Shots should have status
- The application should run on a Linux server
- Warnings if the user does something wrong
- Versions should be handled
- Easy to jump between shots

**Needed for the user to work more freely: (needed)**
- Comments on every frame by drawing and writing text
- The tracker should create the file structure on SAN
- A page called "Updates" where all the users recent actives are shown
- Different users should have different rights
- Flexible interface
- Page called "my tasks" showing all the users current jobs
- Data should be stored and read from Mac OS, Windows and Unix.
- It should not look data like and film terminology should be used
- Everything should be updated automatically

**Needed to improve and fasten workflow the user tasks: (required)**
- Thumbnails should be generated automatically
- QuickTime’s should automatically be generated
- E-mail notification
- Projects should be achieved
- Transfer files between offices
- Backup function

**Needed to give the user full control: (wanted)**
- Full screen mode
- Import data from Farmers
- Deadlines
- There should be different tracks for films, effects and elements.
- DPX-its duration and whether it's online or not
- Search function to find specific files
- Huge files should be able to be moved or kept locally
- Trashcan - files deleted after 2 days. Even hard delete
- It should be the same login everywhere.
- The user should be warned when the disk is 90 % full
- Flash player timeline
- Help function needed

**Needed to make the perfect system: (additional)**
- See the clip before and after the effect shot in the film view
- Select between small and large thumbnails in the list mode
- Be able to open shot in Nuke
- Head and tails should be showed in element mode in the flashplayer
- Automatic creation of filenames
- The artist also wants the tracker for browsing the effect library (with generated thumbnails and QuickTimes).
- Create tasks in the film view in the flashplayer.
- Zoom the timeline in the flashplayer
- Attach sound file and hear audio while watching the movies. Be able to select in-point on the sound in order to sync with video.
- Import and export shots to the Autodesk systems Flame, Smoke and flint.
- Undo function.
- Pdf-export
5.1.3 Conclusion

As Austerberry said it is important not to aim too high. The developer should therefore start implementing the "essential", "needed" and "required" requirements. When the essential requirements are fulfilled users can start evaluating the system and come with valuable feedback. When the three first categories are working great, "wanted" requirements could be implemented. When they also work fine the system will get closer and closer to perfection. If all requirements are met a fully automatic system that supports all files, locations and workstations are obtained.

5.2 How should the files be handled?

After an interview with a system administrator four mayor problems regarding file handling was established:

- No organized file handling exists on the network
- Anyone could go in and create maps and delete files almost everywhere
- No versioning exists
- No metadata exists on the network

When a DAM solution will be implemented these problems will be solved. The most basic functions of the tracker file handling system are: create unified map structure, prevent users from creating and changing maps, handle versioning, connect metadata to the files. Already after the second stage (see 5.1) this will be obtained.

The interview with the other companies gave a better understanding how the problem was solved and what is required. SVT's system could serve as a roll model for the perfect system. All files are transferred automatically between offices. The archive is automatic. They also have smart naming functions for filenames and folders. Their system is very good, but to advanced for The Chimney Pot.

The file handling system at Ducklings is more suitable for The Chimney Pot. At Duckling’s the user have to do everything from the tracker. Maps cannot be created or renamed outside the tracker. The tracker predefines the names of the maps, but the user has freedom to create his own filenames. In their tracker everything seems to be on the same physical location though it's not. The tracker reads information from several physical locations and combines them. (Ducklings, 2010) The file handling system at Ducklings serves as a good roll model for The Chimney Pot.

Saving all versions vs. latest.

There has been a big discussion if all versions should be saved. At Ducklings they save all versions of shots and elements. Duckling means that it today doesn't cost so much more to save all versions. Hard drive has become very large and cheap. If a film is made and only the last versions are saved the company might regret them selves. If the film turns out to be a huge success all behind the scenes material should be used. On the other hand it is relatively seldom that old project are opened and reused. (Ducklings, 2010) The Swedish television saves all versions of a film that have been sent. If there are 7 different, as they call it, revisions only sent one will be archived. (SVT, 2010) At stop they don't archive any material that could be rendered. If setup files were included while a new version was uploaded only the low-resolution shot and the setup file could be saved. The user should have the freedom to choose what ever he wants to archive. The latest versions should be selected as default.

Freedom vs. restriction

Another question that have been discussed in this project is how much freedom users should have when it comes to handling files and maps. The users them self doesn't mind some more freedom, while system administrations want to prevent them from doing too much. Users should
not be able to create maps; all maps should be created from the tracker. Files on the other hand could be added from the outside. It would facilitate for the artist if he could get the source material from one path and render to another. The tracker will monitor the users "out-map" and will see everything he renders to that folder. (Ducklings, 2010) A low-resolution QuickTime will not be rendered until the artists presses done.

I the users are allowed to drop anything they want in the tracker unnecessary material might be added. Deleting things should be harder. The user shouldn't be able to delete stuff by mistake. This should therefore be restricted to the tracker interface. (System administrator, 2010)

STOPP is not interesting because they have no DAM implemented. Duckling is about the same size as The Chimney Pot and their way of handling files are interesting for The Chimney Pot. The design is simple but not too attractive.

**Backup**

When a file has been broadcasted at The Swedish Television it will automatically be sent to archive. This system is too complex to implement at The Chimney Pot. I order work like this a new infrastructure must be bought. The ways the two companies use their files are also very different. At a television channel the filmed material will be used over and over again. They are used to illustrate something that happened, for example some news. At the Chimney Pot most films are commercials. When they have been sent it's very seldom that the material is reused. The Chimney Pots tape robot could continue to be used for backup. It has 32 tapes of 400 GB each. If an algorithm could define the possibility of reuse this system would be sufficient. When the tapes are full the robot spits out the tape with smallest chance of reuse and start saving on the new tapes. A temporary online archive could also exist. When the files then have stayed for a certain amount of time they will be put in the deep archive.

**5.2.1 Conclusion**

A similar system as Ducklings have should be created. Maps should only be able to be created from within the tracker. The user should not be restricted from creating maps and putting material everywhere. It should be some places where the material could be put. The user can select filename himself. The archive should be semiautomatic. The tracker will first put the material in an online archive. If the files are not used they will be put on tapes. When the tapes are full an operator have to load new empty tapes. If a user should need some old project an operator has to load that tape manually if it's not already in the robot.

**5.3 Should the new system be bought, rented or built from scratch?**

To implement a system there are many different ways to go. The systems have in close cooperation with users been tested and evaluated. The most important issues have been what the different solutions have to offer and how easy their interface has been.

**5.3.1 Extend existing system**

If a system already exists it might be possible to for the company itself extend or modify it. There are often many different scripts or plug-ins could be used. (Austerberry, 2006) The Chimney Pots current system was developed in Ruby (a programming language). The source code for the system could not be obtained. Ducklings also recommended us to develop a new tracker using the language Python (Ducklings, 2009). The existing system should then not be modified.

**5.3.2 Rent standard system**

The implementation will go fast and it will be relatively cheap because no hardware or developing team is needed. Support, daily backups, unlimited number of users is provided etc.
A negative aspect is that all traffic must go over the Internet. In these standard solutions all users needs are regarded, which often leads to a system that doesn't work for anyone (Swartling, 2006). While doing the evaluation with the artist most standard solutions seemed too complicated, missed important functionality or lacked in design. They were dismissed after only a brief overview without more detailed user tests. The solutions chosen was Cozimo (for being able to draw directly on films), Alienbrain (robust system with advanced versioning) and Shotgun (Flexible and elegant design). None of those could alone serve The Chimney Pots all needs. After careful evaluation it became clear that locally installed software was not an option. The system needs to be available on the Internet. Clients, freelancers and offices in other countries are going use it (System administrator, 2009). To have everyone install software is too complicated. Therefore the Alienbrain was not selected. Another issue was that Alienbrain didn't have any possibility to draw on movies, which was one of the specifications. The shotgun seemed like a very interesting solution. The mayor problem was that this application was too big. They have many unnecessary functions. It should be as simple as possible. Then it was only Cozimo left. To rent Cozimo was also dismissed. All files should have to be sent by Internet (Austerberry, 2006). The option was then to buy it and install it on a server on the Chimney Pot.

### 5.3.3 Buy standard system

During the project much time were spent discussing the Cozimo flashplayer. It is a collaborative tool that The Chimney Pot where very interested. Cozimo was contacted and asked if it was possible to buy their solution. It was possible; the flashplayer could be bought and implemented locally. Unfortunately this was quite expensive (per year) (Cozimo, 2010):

- 10 users - $5,000
- 50 users - $15,000
- 100 users - $30,000
- + $3,000 in development fee.

The developers of Cozimo suggested that 10 concurrent users would be enough for Chimney to start with. If Chimney should start using it for about 5 years it would add up to $25,000. This is many times more than the budget of the whole project. They offered a free trial and the software was tested locally.

### 5.3.4 Combine systems

If one single system doesn't meet the company’s needs several different solutions could be combined. Though it's important that they are able to work together. (Auserberry, 2006) The only system left was Cozimo. It could be combined with a new built system.

### 5.3.5 Build a new system

To get a system that perfectly suits all needs it has to be built from scratch. The most important tip is not to aim to high. The aim shouldn't be to build the best system ever. It's enough to build a small working pilot for in-house testing (Austerberry, 2006). After interviews with the Danish expert at Ducklings and the system administrator it was recommended to use Python and an open source framework called Django to develop a new system (Ducklings, 2009).

### 5.3.6 Conclusion

The best solution for The Chimney Pot is to build their own application. The Cozimo flash player is too expensive. If it were possible to build an own flashplayer it would be the best solution. No similar flashplayer have been found on the market. It seems like a quite hard task to develop. If it after a while seems like the flashplayer couldn't be implemented the Cozimo could be bought and installed at The Chimney Pot.
5.4 How should a new system be designed?

The new system was designed with the requirements as a solid base. Ideas from the online solutions, the other companies, old systems and the draft gave a base for the graphical layout. During the design process Nielsens heuristics was used as guidelines to avoid most common errors (Preece et al, 2002). Though it was very hard for the developer himself to find the errors and ambiguities. Everything seemed crystal clear for the developer, but the users where of a different opinion.

5.4.1 Screen design

The main screen design with one main frame, one upper and one left frame doesn't need to be commented. The content on the other hand does. There where many different objects like comments, thumbnails, shot info, status, different field etc that had to be placed. They where mapped in groups after similarity. It was a hard task to decide what should be grouped together and where it should be placed. The most important functions were identified. It was the flashplayer and the comments window. All other information was hidden to obtain a minimalistic design (Preece et al, 2002).

5.4.2 Menu design

There where many problems for the users to understand the prototypes. They did not understand how to navigate the menus. One obvious problem was that it was three different menus to use in the beginning. If we count the thumbnails it will be four. A lot of time where spent on redesigning them to make it more understandable, but the confusion remained. The designer could not understand why the users didn't understand his menus. If we go back to the literate the answer is found. There were no big problem with the graphical design; it was the conceptual models that where wrong. They’re where many different conceptual models mixed in the same interface. The models where on top of all used in the wrong way.

The tab metaphor

The tab metaphor is a representation of a real physical map (Preece et al, 2002). In the prototype a menu had been built in the tabs. Each tab now had three functions: list mode, open mode and full screen. The user was able to select three functions in each tab. In the real world maps with three or two functions don't exist. There is only one way to open a tab. The tab metaphor was used to create something that didn't exist in the real world. The users could never understand it. They didn't know why they didn't understand it. They just felt confused.

The windows explorer metaphor

Most people know how the Windows Explorer works. There are bunch of maps that can be open and explored. The Explorer is based on the old file archives that existed in companies and institutions. There where big file cabinets containing many maps. (Wikipedia, 2010)

The biggest problem in the tracker was that the objects in the left menu didn't look like folders. It was projects, films and shots. If the user clicks on a film he expects to see the film, not what's in the film map. Actually projects, films and shots are maps on the server, but this cannot be seen or understood by the user. Further more the explorer is located where the menu normally is located in a web interface. The user expects it to be a menu; they don't notice the small text on the top saying explorer.

The bottom button metaphor

The bottom menu was removed after first the first user test. The buttons was formed like physical buttons. When they are pressed it looks like a lamp inside of them are lightened. The user clearly understands that they are buttons that they can click on them, but there is a mayor problem: they are not connected to anything. The text on the buttons are "Yes", "No", "+" and "rubbish bin". The menu should be under all levels of hierarchy. The problem is that no one
ever will understand what "Yes" and "No" stands for. That "+" and "rubbish bin" means to add and delete something is understood, but what is it that should be deleted is not clear.

The film metaphor
The thumbnails are put in a column and look perforated exactly like a real film. All users understood that it was a film. On a film there are many frames in a sequence. All frames are connected and form together the moving images. Here a metaphor is used incorrectly one more. The frames are used for projects, films and shots. There is no difference at all. If people click on a film they expect to see a film, not any information about project.

The film metaphor effect is so strong so that it doesn't matter if the frames around the thumbnails are changed. They are still in a sequence a film. They are connected and are a unity to the user. He cannot imagine that they are separate projects.

5.4.3 Icon design
Another problem was the design of different icons for menus and tables. Common keyboard symbols from the font Wingdings where used as base for the design of many buttons. The users often didn't know what the buttons represented. Different icons where tested in order to find some understood by most users. It was not an easy task. After many user tests most buttons where understood by the users.

Shot status
The shot status has been discussed back and forward. The idea was to do something good that people understood. Many different proposals have been made. What has been totally forgotten is when the project started the users said that the best thing, with their old tracker was the status system. It had actually been working great. It can then be said that a lot of time and effort have been spent on trying to develop something that already existed and was working fine.

5.4.4 Conclusion
It becomes more than obvious that the designer misunderstood conceptual models and metaphorical models. A lot of extra work was needed in order to remove the confusing elements. Due to many different user tests the problems where solved. Most users understand the model used in the latest prototype. The design proposal is the screenshot of the final prototype. This prototype exists in an interactive version. On request it can be sent to anyone interested.

5.5 How is a prototype created?
Prototypes can be created in many ways but only two where selected: high fidelity static and high fidelity interactive. Unfortunately there are some drawbacks while using high fidelity prototypes.

5.5.1 High fidelity vs. low fidelity prototypes
A high fidelity prototype was chosen despite the warnings from Preece. Preece explains that they take very long time to build and that the user tests often gets superficial with many complaints about details instead of structure. The users might also be afraid to criticize something that has taken days or even weeks to build. Small bugs can also stop the whole user test (Preece et al, 2002). The prototype can also affect the developers. When the developers think that the prototype works they might not consider other alternative. Sometimes they get stuck in their prototype. (Preece et al, 2002). This is exactly what happened in this project. The prototype became more and more complex and took in the end weeks to redesign. When so much time has been spend it is very hard to leave it and start from the beginning. Once again by extensive user tests the product was developed into something the company liked.
Preece's warning should have been taken seriously and a low-fidelity prototype could have been created. "Use low-fidelity prototyping to get rapid feedback." (Preece et al, 2002) "To get a good idea, get lots of ideas" (Rettig, 1994 (Preece et al, 2002)) The problem was that the developer felt ashamed to show his boss a few simple sketches on papers after being away for several weeks. The outcome of using low-fidelity prototypes cannot be estimated. It might not have lead to a better result in the end, but would most probably have saved time.

5.5.2 User tests

In this project a user-centered approach has been the goal. The users have been involved during the whole process and have been asked for opinions and improvements. It was thanks to the user tests that the conceptually illogic menus where discovered. If no user tests had made the system would have been useless. If now one understands show to use the system it would probably never have been used.

Swartling writes that users many times are involved in the project, but that their opinions many times are neglected. The developer does what he wants anyway. This became obvious when reviewing all interview notes. The users had said more or less the same thing during each interview or user test, but the developer was so stuck in his model. He did small changes but the users demanded bigger. Step by step the result was obtained anyway.

5.5.3 Conclusion

The prototype is a part of the design proposal. The final prototypes have about 15-20 pages where the user can click around to "feel" the system. Most common tasks can be made. This prototype exists in an interactive version. On request it can be sent to anyone interested. Despite the lack of understanding of menus and cognitive models the result became good and the company became satisfied.

5.6 Other issues

EDL vs. XML

Another issue was about using EDL to import films. In the EDL information about all shots exists (see 2.2). It would be possible to use an EDL to create films with different shots automatically. Unfortunately there are many problems regarding EDLs. Sometimes frames are dropped in the off-line process and the EDL and the shots are out of sync. Sometimes there are some extra frames before the first shot in either EDL or the off-line. It must be some manual synchronization between EDL and the off-line. The reel in the EDL can have a maximum of 6 characters. In RED footage the reels have longer names. All shots could then get the same reel, even though they don't. It will there fore very difficult to be able to import shots with EDL.

According to the Danish expert it will be much easier if XML is used instead. In the XML the metadata are saved as nodes. It will be much easier for the computer to read it. It is also possible to save much more information about the film than in an EDL. If this function should be implemented it could facilitate the workflow very much.

Film vs. shot

Another issue has been regarding looking at entire films in the flash player. The flash-player could make a play list of all shots, including the non affect ones. This would be an excellent function. The new effect shots would instantaneously being updated in the film. It would then be possible to assigning tasks directly from the flash-player. If the user draws something on the non-effect shot an effect shot will be created and the producer informed. He could then assign an artist. This seems like a very good feature, but the difficulty involved is quite high. This function therefore has to be put on the shelf until more basic functions are done.
5.7 General solution

In order to make the tracker more general a discussion have been held with experts in Denmark, Ukraine and Australia. Much time have been spent in developing a database model that should be usable for other companies. All possible scenarios should be predicted.

Though it's dangerous to think to general. If everything is made to fit everyone it will fit none. The goal has therefore been to adapt it as much as possible for The Chimney Pot. When it works fine for one company it can be tried on other.

One issue that is very important, especially in London, is security. If freelancers should be able to download high-resolution DPX-sequences this would be a huge security risk. According to The Chimney Pot this is not a big problem. They don't think that people are interesting in stealing parts of commercial. Another issue is if the low-resolution film is available (regarding music videos and feature films). This means that everyone that has access to the tracker could see it. Only users involved in the projects will have access to the files.

5.7.1 Software development

There are many things that the tracker should handle automatically. The tracker is meant to simplify as much as possible for the user. In order not to make the tracker core to complex many of the functions should be a separate program. They will serve and assist the tracker. The tracker just sends out commands to the different demon assistants. The demons will be stupid. They will only lay and wait for instructions.

This enables the system to be more flexible both developing and implementing at other post production companies. In the beginning it should only be a tracker core that the users have to serve manually. They have to enter data manually, generate QuickTime clips manually, send the files manually, manually archive the files etc. During the project development more and more demons should be created and the system will be more and more automatic. Some functions are also good to integrate into the tracker core. These functions are described in the next chapter.
6 Recommendation

This chapter gives an overview of how the new system should work. Functions are specified and explained in detail. First a new workflow is described in order to give the reader understanding in what the system should do.

6.1 Workflow

This is a proposal for a new improved workflow while using the new tracker as an essential tool. First a project will be created in Farmers wife. When it exists in the Farmers database it can be imported into the tracker.

In the beginning of a project ideas and attachments will sent between producers and clients through the tracker. All files are saved and organized. In the tracker files can be attached on all levels in the hierarchy. There are two ways of adding attachments before the high-resolution clips are added. They can either be uploaded as attachments or they can be uploaded as shots that later will be replaced by real clips. The attachments can be QuickTime movies, images to pdf-documents, text files etc.

When everything is filmed, the clients will make a low-resolution off-line version and send together with the XMLs and the high-resolution material. The clients could them selves upload these two files on the tracker. (The clients currently use EDL, but when the tracker is done the clients will be forced to send XMLs instead.) When both files are added to the project the tracker starts dividing the off-line version into different shots. All tasks are on hold until the high-resolution DPX-sequences are on the server.

The material is then scanned by the ARRI scanner and stored in the correct project folder. It is only necessary to have subfolders for the effect shots. The tracker will create these folders automatically. There will be a predefined storage places for the different media. When the tracker finds a DPX-sequence on the server it will send it to its correct destination (a workstation or another server in another country). A low-resolution QuickTime file and thumbnails are also generated. When everything is ready the users will be notified. If the
sequence is not automatically loaded to the local machine the editors will download it themselves. There could be a button called "open in Nuke" with a script that opens the clip in Nuke.

The colorist will start working when all the material is scanned. While the colorist is watching the film with clients he often finds new tasks. He should use the flash-player to paint and write and send it to the producer.

The editors will use the non-color graded clips to work with. When an editor has finished a task he will upload a new version through the tracker (it can be either DPX or QuickTime). He can also select to render the material to a folder monitored by the tracker. For important shots the setup files should also be uploaded as attachment. On order to send it to the producer the artist has to press "done". The producer wants to see and approve the shot before it’s sent to the client. If the client doesn’t approve it the editor has to redo his work according to the clients’ requests. When it’s approved the DPX sequence will automatically replace the original sequence on the server. The original sequence will be but in a subfolder.

If the shot is made by another office, the DPX will now be sent to the responsible office. The colorist will be notified (shot name and time code) so that he can re-cache the current shot on his computer. The procedure continues like this until the client has approved all shots. The communication can go directly between the artist and the client to improve speed and avoid misunderstandings. The colorist will always be notified when a new shot is approved so his version of the film will always be the latest. When the client approves the final version the material is exported to its destination format. The project will then be set as inactive.

After two month the producer will get an e-mail that the project is going to archive. Only the latest versions of the shots are preselected. He has two days to select anything else. It is also possible to go in directly into the tracker and select archive. Then a list with checkboxes in front of all files will be shown. The user could select what files he wants to archive. Then all the files are gathered and sent to the tape robot. An online archive could also exist where the files first are located. If they are not used they will be sent to the tape robot. All the low-resolution material will be kept on line.

### 6.2 Functions

All functions have been categorized after falling priority. The developer should start implementing the "essential", "needed" and then "required" requirements. When the essential requirements are fulfilled users can start evaluating the system and get valuable feedback. When the three first categories are working great "wanted" functions could be implemented. When they also work fine the aim will get closer and closer to perfection. If all requirements are met a fully automatic system that supports all files, locations and workstations are obtained.

#### 6.2.1 Prioritizing requirements

**Needed for the users to be able to perform their tasks:** (essential)

- User-friendly, clean, simple and most information should be hidden.
- Everything should go fast to do
- It should work in Internet Explorer, Safari, Firefox and on both PC and Mac
- People with less technical knowledge should be able to use it
- It should be possible to assign tasks to users
- Upload attachments (and thumbnails)
- DPX, QuickTime, different images and pdf-files should be supported.
- Path to the high-resolution material
- Shots should have status
- The application should run on a Linux server
- Warnings if the user does something wrong
- Versions should be handled
- Easy to jump between shots
Needed for the user to work more freely: (needed)

- Comments on every frame by drawing and writing text
- The tracker should create the file structure on SAN
- A page called "Updates" where all the users recent actives are shown
- Different users should have different rights
- Flexible interface
- Page called "my tasks" showing all the users current jobs
- Data should be stored and read from Mac OS, Windows and Unix.
- It should not look data like and film terminology should be used
- Everything should be updated automatically

Needed to improve and fasten workflow the user tasks: (required)

- Thumbnails should be generated automatically
- QuickTime' s should automatically be generated
- E-mail notification
- Projects should be achieved
- Transfer files between offices
- Backup function

Needed to give the user full control: (wanted)

- Full screen mode
- Import data from Farmers
- Deadlines
- There should be different tracks for films, effects and elements.
- DPX- its duration and whether it's online or not
- Search function to find specific files
- Huge files should be able to be moved or kept locally
- Trashcan - files deleted after 2 days. Even hard delete
- It should be the same login everywhere.
- The user should be warned when the disk is 90 % full
- Flash player timeline
- Help function needed

Needed to make the perfect system: (additional)

- See the clip before and after the effect shot in the film view
- Select between small and large thumbnails in the list mode
- Be able to open shot in Nuke
- Head and tails should be showed in element mode in the flashplayer
- Automatic creation of filenames
- The artist also wants the tracker for browsing the effect library (with generated thumbnails and QuickTimes).
- Create tasks in the film view in the flashplayer.
- Zoom the timeline in the flashplayer
- Attach sound file and hear audio while watching the movies. Be able to select in-point on the sound in order to sync with video.
- Import and export shots to the Autodesk systems Flame, Smoke and flint.
- Undo function.
- Pdf-export

6.3 Description of the functions - daemons

There are many things that the tracker should handle automatically. The tracker is meant to simplify as much as possible for the user. In order not to make the tracker core to complex separate programs should handle many of the functions. They will serve and assist the tracker. The tracker just sends out commands to the different demon assistants. The demons will be stupid. They will only lay and wait for instructions.
This enables the system to be more flexible both developing and implementing at other postproduction companies. In the beginning it should only be a tracker core that the users have to serve manually. They have to enter data manually, generate QuickTime clips manually, send the files manually, manually archive the files etc. During the project development more and more demons should be created and the system will be more and more automatic. Some functions are also good to integrate into the tracker core. The daemon functions are described below:

**Automatic scanning of project maps (demon)**

One demon should constantly monitor (the monitor demon) all the different project maps for changes. If a file is added or changed this program should notify the tracker and other demons. If a new DPX-sequence is added this demon will tell the tracker, the thumbnail generator and the QuickTime generator. The tracker will then update it’s database with the new information. When the QuickTime and thumbnails are done these demons will notify the tracker and put the file on the correct place.

**Generation of thumbnails (demon)**

This is a simple demon. If a DPX-sequence, QuickTime or an image file is provided the daemon will generate thumbnails. It will just lay and wait for a command from another demon. If someone have made some comments on the shot in the flashplayer it might be good to select the thumbnail from this location. The tracker could just attach information about timecode in the instructions for the thumbnail generator.

**Generation of QuickTime (demon)**

This demon also just lies and waits for a command. As soon as someone uploads a DPX-sequence the QuickTime generator will start generating a low resolution QuickTime version. A command is sent from the monitor demon. The high-resolution files should only be in the background and never bee seen by the user unless selects to download them.

**Automatic shot status**

The user shouldn’t need to manually change and update the status on all shots and projects. The tracker can monitor what the users do and change the status itself. This should be integrated in the tracker core. There are five different statuses with five different colors:

- Gray – not started / inactive
- Red – work in progress
- Orange – done
- Yellow – for approval
- Green – approved

When a shot is created and assigned to an artist by the producer the tracker knows that it is a new shot that no one have started working at.

When the artist logs in and downloads the DPX-sequence the tracker knows that the work is in progress. If he receives the file from outside the tracker he have too manually press "Work in progress".

When a new DPX-sequence or QuickTime is uploaded into the tracker the user is able to send it for approval to the producer. The status is now set to “done”. When the producer approves the shot it goes to the client. The client will now get a new shot to approve. If it’s not approved the client makes some comments and the artist will get a new task. The shot will now get back to "not done" but will be in “in revision” mode. All revisions are counted so it's easy for everyone to see. When the artist uploads a new version again he can decide to send it directly to the client for approval. After approval the status will be "approved.

**Automatic file transfer (demon)**

If the high-resolution material doesn’t exist when a task is created the task will be on hold. The user can go in and see them, but he will not be notified by e-mail. When a task is created the producer will select one editor and one machine for the shot. The tracker now knows to who and
where to send the DPX-sequence (if it's another office). The transfer function should be handled by a demon. The tracker just tells the transfer demon what should be send and where it should be sent. If the editor for example works in Kiev, the tracker will know this. When the monitor demon discovers that a DPX-sequence is on the server it will notify the tracker. The tracker will then notify the transfer demon and the file is automatically sent it to the server in Kiev. When the file is transferred the transfer demon reports to the tracker. The tracker now contacts the e-mail demon, which sends e-mail to the editor telling him that he has a task to perform. When the editor uploads a new version, the DPX-sequence will remain in Kiev. Only the low resolution QuickTime will be transferred. The QuickTimes and thumbnails will be mirrored on both sites. The DPX-sequence will only be transferred after the client has approved the shot.

**Version handling**

The tracker should control the map structure for versions. It might be handled by a separate demon as well. The editors and producer shouldn't have to think so much where the material is stored. They will get a task and be able to download the DPX-sequence or open it in the editing program and work centrally on the material. For each new version of the clip there will be a new map (v01, v02 etc.) The scanned material will probably be in the v01-map. When something is put something in the v02 map the tracker will discover it and update the database. When something is put in v02 the tracker automatically creates a v03 map. If something is put in the v03 map, the v02 map will be locked and an empty v04 map will be created. The editor must click "done" on a version in the tracker before it will be sent to the producer or client. This might be a default setting, but the users should be able to change it if they want.

**The smart tracker**

For the application to run as fast as possible the tracker should be "smart". Most times when a user logs in the tracker already knows what the user is going to do. He will most probably look at his tasks, approve or look at what he should do. The tracker should have pre downloaded all the data to the user. When he logs in everything is ready to play. He doesn’t need to wait for data to be downloaded.

While the tracker is generating thumbnails or QuickTime clips the new tasks will not be sent. The tracker will wait until everything is ready to play before the users are informed. Different users will use the system differently. The tracker should be aware of this in order to know what do prepare and download in advance.

The tracker should also remember where on the timeline in a shot the user stopped watching, the last opened project and film that was open and etc. The user shouldn’t have to select the same thing twice.

When something is wrong or missing the tracker will automatically inform the person responsible. The tracker will indicate what is wrong and the consequences that will follow. The user is then guided to correct the errors. If necessary fields are not filled in, for example an e-mail, that user will not be notified that he has tasks to perform. An error message will then be showed in the comments window. Most of these functions should be integrated in the tracker core.

**E-mail notification (demon)**

The tracker will automatically send e-mail to the users about new tasks and updates. This should be a separate e-mail demon. When the tracker wants to send e-mail to someone it will talk to this demon. It is just waiting for a message and one or more e-mail addresses. In order not to spam the user there should probably only be one e-mail per day with all the information gathered.

Everyone should create a folder in his or her e-mail client that’s called Tracker. Now everyone will easily know that something has happened in the tracker. The user will see what has been done, if he has some new tasks etc. The users e-mail box will not be spammed because the tracker related e-mails would be separated from his other e-mails. If such a filter were used,
maybe the tracker would be allowed to send more e-mail. It could inform the user directly when something really important happened.

**Archive/backup (demon)**

Another important function is to send the finished material to archive. The SAN network will quickly be filled with different material. The archiving function should also be automatic. An archive demon should run in the background and know about all files on the network. It could get information by the monitor demon. The archive demon should monitor how long time all files have been on the network and when they where last used. If a file hasn’t been used for a long time it should be deleted or sent to archive. Different files that are used frequently should be put in an online archive.

When a project is marked as done and no one have worked with it for a certain amount of time the archive demon should tell the tracker that in will contact the e-mail demon. The responsible person will now be asked if he wants to archive the project. (It will show up as a new task for him.) What should be archived is predefined. Normally it should be only the latest version of shots and elements. The user only has to click “yes”. If he wants to archive something special he can to go in and check and uncheck the files that are about to be sent to archive. The low-resolution QuickTimes and Thumbnails will be kept online. If someone wants to get something that is archived the tracker just tells the backup demon what it wants. The backup demon gets it from the archive and put’s it on line. When the material is not used anymore it will be deleted (it already exists in the archive).

**6.3.1 The flashplayer**

In order to stimulate and simplify video collaboration both between clients and editors and between offices a new collaboration tool will be used. A simple flash player that allows the user to draw and write on he shots is the solution. The flash player can also be seen as a demon, a separate program that talks to the tracker.

On the upper left of the player the user should be able to select which version he wants to see (ex. 30_sek and 60_sek). To the upper right the user can select to show and hide the Flashplayer tools. Below the film itself time code and frame number can be seen. Below this is the timeline. It consists of two layers (on the film level), one for the different comment and drawings and one for the different shots in the film. The small white vertical lines indicate the start of a clip and the black part is the clip itself. If the clip is an effect shot it should have a different color on the timeline, the shot status color. The shot or comment on the timeline that is not selected is darkened out. These functions in such a Flashplayer are:

**Essential:**
- Draw on any frame in movies and images (red color)
- Write text in any frame
• Play, pause, frame-by-frame, comment-by-comment, go to the end/start
• Show and hide the comments and tools
• Show time code and frame number
• Set duration of comments
• Delete comments

**Good:**
• Have a timeline showing where the comments are
• Film view - see all shots in a film. Two timelines, one for shots and one for comments.
• Make squares, circles, line

If play is pressed in the film view the flash player should play all the different shots in a sequence without any delays between the clips. The information related to the different shots should be changed while the cursor moves. All the comments should also be viewable, but only if the cursor is above them.

The latest version is always shown by default. If a client approves an older version that version will automatically be placed on top of the other versions.

### 6.4 Additional features

**Automatic creation of films and shots (demon)**

The idea is that the tracker should read an XML generated from an Avid or FCP and in that way know what shots the film consists of. When the user wants to import data from an XML another demon is needed. It’s a program that takes the data from the XML and imports it to the tracker database.

An XML is practically a film. All the in and outpoints of both the source material and the finished film are imported. Even data about effects and what has been done to the different clips can be imported. The tracker will now know where all files are located and in what order they should be played. The user will easily be able to jump between the different clips in a film.

When the tracker knows where the different clips are located in the off-line version of the film it can divide it into separate clips. In order to get a match between the XML and the off-line there must exist a point of reference on both files. Sometimes there are some extra black frames on the off-line version. This can vary from time to time and is not constant in any way. The user is asked to mark where the second clip starts in the off-line. The second clip is used as reference because it might sometimes be hard to define the exact start of the first clip. This might be hard because the first clip often starts with a fade from black.

When the user are working in an editing program like Final Cut Pro he is able to set markers on the timeline. On each marker it is possible to enter information. This means that the user can connect information to each frame. The marker can also have duration. If the client then put markers in the program he is working he can tell the artist what to do. When the XML is exported this data comes along. When the client uploads the XML and the off-line he made the tracker will know what shots that have a task and will send them to the producer. He can then go in and look at them, and assign them to a user. All info about the project is already there. This might not be an easy task to perform, neither is it a requirement. But it's definitely something for the future.

**Automatic dividing of QuickTime clips (demon)**

One function in the tracker is to use the off-line material before the real material is scanned. In order to do so the offline must be divided into clips. There should therefore exist a demon (the film cutting demon) that receives one QuickTime file and several in and outpoints. The demon will then generate new QuickTime files, one for each specified shot.
Import to the Autodesk systems (demon)
To handle this function there might be a sub-coordinated transfer demon, we call it the Autodesk demon. Its task is to provide the Autodesk systems with source files and scan their directory for new files.

There are practically two different ways of importing material to the Autodesk systems (Flame, Smoke and Flint). It can either be pulled or pushed. If the material is pushed the Autodesk demon will put it on the Autodesk systems (or communicate with the Autodesk API). Certain folders are created and the DPX-sequence is transferred there. Stone and Wire can be used to push data to directly to Flame. The other way is to soft import the shot and it will be cached on the local machine. This could be done with the application “Stonify”.

Flame has a system called Black Burn that handles background jobs. "It is not hard to store and read clips on the Flame" (Danish expert, 2010). Spark API can be used. There are many code examples in how to get frames in and out and to create projects.

Open shot in Nuke, Flame, etc (demon or script)
To simplify for the artists it would be ideal to open the shots directly into the editing programs. There should be a button in the tracker that says, “Open shot in Nuke”. When it’s pressed a python script will run that starts up Nuke and loads the shot to the program. Then the artist doesn’t have to bother where the file is located. A special script should exist for each editing system (Nuke, Flame, Smoke, etc.) The program could then cache the shot locally but work on the centrally stored clip.
References


Bell, Judith (2009) Introduktion till forskningsmetodik, Lund: Studentlitteratur

Cozimo (2010), E-mail from developer at Cozimo (09-07-02)


Gomez, Anna (2008) Improvement on the Workflow at a Swedish Post-Production Company in Moscow, Stockholm: KTH


Nielsen, Jacob and Mac (1994) Usability Inspection Methods, New York: John Wiley & Sons


Swartling, Anna, 2006, lecture in HCI - 2D162, Stockholm: KTH


The Chimney Pot (2009), www.thechimneypot.se (09-10-21)

Usabilitynet, (2006) *Usability net International standards for HCI and usability*, available,
http://www.usabilitynet.org/tools/r_international.htm#9241-11 (10-0525)


(10-05-14)

Wikipedia (2010) *Interface metaphor*, available: 
Appendix

The interviews regarding the system where (To CEO, Artist, Producer):

- Who are the users?
- How many people are actually using the current system?
- What is good with the current system?
- What is bad with the system?
- What functions exists?
- Which functions are used resp. not used?
- What other functions are needed?
- What is “user friendly” for you?
- How does the blog work? (a system used to show shots for clients.)
- What other programs are you familiar with?
- Mention a good program and a good website.

All other interviews and user tests are located in a separate file. If you are interested please send e-mail to kmwa@kth.se.