Project:
Visualizing environmental data on the web

Book Title:
“Software for Use”
(By Larry L. Constantine, Lucy A. D. Lockwood)

Course:
User-centered System Design
2009

Group 10:
Aboozar Eghdam (Aboozar.Eghdam.5945@student.uu.se)
Fredrik Jakobsson (Fredrik.Jakobsson.3115@student.uu.se)
Hjalmar Wennerström (hjwe8407@student.uu.se)
Mubbashar Nazar (Mubbashar.Nazar.7235@student.uu.se)
Table of Contents

Table of Contents .................................................................................................................................... 1
A brief introduction..................................................................................................................................... 2
Assumptions............................................................................................................................................. 3
Roles and people in the project .................................................................................................................. 3
Purpose of the project ............................................................................................................................... 4
Used references....................................................................................................................................... 4
Preliminary steps....................................................................................................................................... 5
  1. Project starts (the first day) ............................................................................................................. 5
  2. Requirements documentation ........................................................................................................ 5
  3. Exploratory modeling ...................................................................................................................... 6
    The iterations begin............................................................................................................................ 7
  4. Role modeling .................................................................................................................................. 8
    Identifying the users .......................................................................................................................... 8
    Defining the roles .............................................................................................................................. 9
    Mapping the roles ............................................................................................................................. 9
  5. Task modeling .................................................................................................................................. 10
    Identifying tasks ............................................................................................................................... 10
    Mapping the use cases ...................................................................................................................... 11
    Write narratives ............................................................................................................................... 12
  6. Content modeling ............................................................................................................................ 13
    Modeling the contents ...................................................................................................................... 13
    Context navigation ............................................................................................................................ 13
  7. Implementing the design ................................................................................................................. 14
    Prototyping ...................................................................................................................................... 14
    Coding .............................................................................................................................................. 14
  8. Testing .............................................................................................................................................. 15
Reflection of the method .......................................................................................................................... 16
Estimation of cost and time ...................................................................................................................... 17
  Cost .................................................................................................................................................... 17
  Time ................................................................................................................................................... 17
  Time Line ........................................................................................................................................... 18
A brief introduction

The approach in this report is based on the book Software for Use by Constantine and Lockwood. The book describes a “collection of coordinated activities” to develop usable software using so-called usage-centered design principles. The difference between user-centered and usage-centered design is that user-centered design puts the focus on the actual users whereas usage-centered puts the focus on usage. The focus is on what the software will need to supply to the user in order to help them accomplish their tasks. This makes the approach much more model driven as opposed to user-input driven in the case of user-centered design.

The “recipe” for designing a usage-centered system contains many models and modeling exercises where there is some user involvement although limited in comparison with the more traditional user-centered. Another important aspect of this proposed method is the abstraction of models. In user-centered design there are a lot of prototypes, mock-ups and user scenarios. The authors of this book argue that using these methods will lead to assumptions of the intended system in an early stage since the methods focus on details and can therefore cloud the bigger picture. When making models abstract there is more room for innovation since things are defined in a loose and in a non-specific way. They use the term bifocal modeling to reference that this design approach gives both a good overview and detailed views of designs and artifacts.

At the core of usage-centered design is the use of these abstracted models referred to as essential models. They are used to “capture the essence of problems through technology-free, idealized and abstract descriptions”. There are three core models that are stressed heavily; they are the role model, the task model and the content model. The role model describes the relationship between the user and the system. The task model tries to structure the tasks a user needs in order to accomplish a task. The content model is a way to structure the tools provided to the users by the user interface and relate them to each other.

The book is written by and for software engineers and the approach is very much an engineering approach. Things are systematic and the process is fully specified from start to finish. There are very little uncertainties and the approach relies on good models instead of good users. The approach is iterative and according to the authors flexible in what parts to include and what parts to leave out.
Assumptions

There is an existing database with all the data we need and an existing (old) system that has been used until now. The job is to make a completely new system that is more user-friendly. We also assume that there is enough funding so that we don't have to leave out some good aspects due to bad funding.

Roles and people in the project

The core of the project consists of four persons that work as developers but also have other roles. There is one project manager and one HCI expert within these four people. The project also consists of one more person that is a human factors expert working about 20% of his time on this project. Apart from these people that are employed within the project there is also a customer (containing many people) and of course users. We have already identified that there are (at least) two very different users. This is because the purposed system is directed both to the public and to researchers.
**Purpose of the project**

The essential purpose of this project is to design and build a web-interface for visualizing and presenting agricultural data about Sweden. The websites two target end-users are the general public and researchers or professionals with a greater understanding of the agricultural field to use the website. Since the amount of data is huge and containing complex relations there is an essential need to conduct this project with consideration to usability.

The current system that provides the agricultural data works well for professionals but is inadequate and very difficult for the public to use. Therefore the purpose of the proposed website is to make the data more accessible and easier to understand for the public as well as maintaining the high complexity needed by professionals. Since there are two very different types of users the website has to function in a dynamic way to accommodate the needs of both. The difficulty in this task is to keep the requirements needed to serve the professionals but at the same time allow for novice usage.

**Used references**

We used the book “Software for use” by Constantine and Lockwood. In supplement to this we also used their homepage and especially a paper on “Usage-centered engineering for web applications” written by the authors of the book.
Preliminary steps

1. Project starts (the first day)

As an initial step in conducting this project there should be a meeting/workshop where all parties in the project are present. This is also the first time everyone is introduced to one another. This activity is seen as the start of the project and should not last longer than one day. The activity in this session is to brainstorm or fantasize without any preconceptions or limitations in mind. The purpose is to “think outside the box” and come up with as many ideas as possible. Having an initial session like this helps the creative process of finding all types of features that might be interesting.

To execute this, a meeting between the developers, customers, and users should take place. The project manager holds the meeting and is responsible for the resulting documents. The participants should be given a very short introduction, typically 1-3 sentences about the project. Then the brainstorm sessions should be divided into two mixed groups. The sessions should focus on innovation and creativity. The groups should think about features, facilities, content, and capabilities. The output artifacts should be maps/drawing/text (let the participants have some flexibility) showing the result. The result should be seen as a surplus of possible things to implement where individual parts can be chosen for implementation later on in the project. This session also serves as an opportunity for developers to get an understanding of the usage and contextual situation of the system. This first step provides everyone within the project an understanding of the overall purpose.

2. Requirements documentation

As there by now is a greater understanding of the purpose of the system we can start formulating the requirements. About four persons working full time should start formulating the initial requirements and it should take at least two days. In this group there should at least be customers and developers. At the end of those two days there should be a first draft to a essential requirements specification that will be improved later.

The next step in this stage of the development should be to have a meeting with the other actors that shouldn't take more than a day. These actors include the human factor expert, the customers (perhaps more people than in the development of the requirements) and the different types of end users. At this meeting everybody should give their input and comments
on the essential requirements. This is to get a wider view and to pick up on things that were missed in the first step.

After gathering the input, the system developers analyze and evaluate the new information to pinpoint a new refined requirement specification. This session should last for no more than a day to end up with these refined requirements. Now an iterative process starts to present these rewritten requirements for the other actors and they can once again give their input and comments on them. These activities will continue until the developers end up with an approved artifact, in this case a suitable requirements specification document. The reason for conducting these meetings is to get a more user based idea of the requirement aspects as well as making the customer satisfied. This will help the project to take a more usage-centered approach in general. Although this is the final artifact in this step, it should be changeable in future steps due to the iterative approach that will refine the requirement. It is the project manager that is responsible for the requirement document and that the requirements are addressed later.

3. Exploratory modeling

To finalize the preliminary steps in this project there should be a session where the newly written requirements specification document is analyzed to identify possible issues. This can be done by the developers as a meeting for half a day. The objective is to go through every requirement and identify if there is anything unclear or uncertain. There should also be a risk analysis to show what parts of the projects have the highest risks. The project leader has responsibility over the risk analysis and it should be presented to the customer. The analysis is just to make everyone aware of the risks and not primarily avoiding them. It is a document to widen the understanding of the project to all parts involved.

This process is important for the developers because they have to make sure the requirements are clearly stated and well defined. When this meeting is over there should be a document identifying requirements that might need to be reformulated or even new requirements that needs to be added. The questions and issues raised by the developers should be talked over with the customer to clarify what they really want. The resolved requirements list should therefore also contain input from the customer. After this the developers should have a requirements specification that they fully understand and are able to start working from.
The iterations begin

Now the preliminary steps are complete. After this the iterative process starts that will continue on throughout the project. The customers and users are less active and the work will mostly be done by developers. The iterations are focused around the three essential models to be used. These models are according to the authors used to model usage in a project and should be seen as artifacts to be used later on. The focus is not to observe the user but more to describe what they want to do in an abstract way. In each iteration the models are reviewed and altered if necessary. The first iteration is usually a bit heavier and takes longer time then the following ones.
4. Role modeling

Identifying the users

The initial face of this step is about identifying and defining the different users that will play an active role in the system. Here active really means interactive because an end user that plays a part in the system interacts with it. The users are actors in the project and the roles are what parts will be played. A single user can play several roles as well as a role can be played by several users.

To identify and understand the users we must find what kind of users that will interact with the system. As an initial part of finding the users there should be a corporation between the developers and the customer. The work could start with identifying the users of the old system and then try to identify possible new users.

The developers, the customer and users from the old system gather for a meeting and here an understanding of those will lead to a way of finding possible new users. This work of finding new users starts with sending out some questioners to people that were the outcome from that meeting.

The next sub face in this step is to collect the questioners and evaluate them and identify who the new users are. The evaluation face is conducted by the developers, the customer and the human factor expert and shouldn’t take more than one week. Analyzing the result of these evaluations is the next face in the work of identifying the users. This analysis is supposed to answer questions like user categories and types. The analysis should answer the question if there are more users of relevance. The result of categories and types of users has to be shared at a new meeting where new evaluations take part in order confirm these types and activities and also answering the question if there are more users of relevance. If the answer to the last question was yes then a new questioner is send out to a more general and broader public. After this all processes of this step is being performed again. When all parties are satisfied with identifying the users the result of this step is an understanding of all different types of users. The most considerable aspect of this understanding is finding the focal user and that is the most common and particularly important user.
**Defining the roles**

“A user role is an abstract collection of needs, interests, expectations, behaviors and responsibilities characterizing a relationship between a class or kind of users and a system.” (Constantine and Lockwood 2006). Also according to Constantine and Lockwood this role has to be played by whom who has an assumption of particular relationships to the system.

To identify the roles that will be played by users in the system there should be a meeting including all participants in the developing process. That means that the developers, the human factor expert, the customers and the users gather for that meeting. The goal with this meeting process is to define the behaviors, needs and expectations played out within the system. It is these factors that built up the user roles. The result of these meetings will be the user roles and it will take at least 2-3 days, but if the result is not satisfying enough it has to be done iteratively and therefore take longer. Maybe a need to go back to the previous step, identifying the users, is required and therefore it takes even longer. The main responsibility of the project manager here is to decide when this section is complete.

The most important and considerable roles within in the system is played by the focal user and that is the focal roles. From the result of identifying the users section, the focal roles are identified, from the focal users. The important part of the system is the interface and that has to be shaped with great consideration to the focal roles.

**Mapping the roles**

The role mapping is a way of capturing the overview of the system and it shows who will use the system and how. This map consists of relationships between all rules in different ways like affinity, classification and composition (Constantine and Lockwood 2006).

As the user roles are indentified in the previous step it is now time to conduct the role maps. This work starts with the developers meeting up for a day to identify the relationships between these roles. The result of this meeting should then be presented to the others participants in the development of the project. At this second meeting the customers, human factor expert and the users give their input on the role mapping in order to make it more precise. This input could lead to removal, adding and replacement of relationships in order to redefine the role maps. This process can also be iterative; it is up to the project manager to decide when the mapping is done.
In the process of identifying these relationships it can occur that some of the roles are similar. If this occurs there is a need to go back to identifying the roles section and reconstruct the rules in order to deal with these similarities. This process of handling the similarities shouldn’t take more than one meeting since the roles should be quite clear and therefore pinpointing the similarities won’t be a problem.

At the end of the last meeting in the role modeling section the project manager decides if the role modeling is done within the range of the first iteration. The project manager does this by trying to establish if the participants has reached a certain satisfaction level. When this is done the artifact from this whole role modeling process is the role models. These artifacts consists the different user types, roles and mapping of the roles. According to Constantine and Lockwood, These artifacts are the primary resource for building the task model which is the step. But they also have direct and indirect influence on various aspects of the user interface design itself.

5. Task modeling

Now there should be an understanding of the users and who they are. The next step according to the authors is to identify what work they need to do. Here the role modeling can help to identify what tasks that needs to be modeled. This is done by modeling the tasks in the system. Tasks are essentially descriptions of what the different users want to do and how they interact with the system. The purposed method here is a more abstract way of identifying the tasks then in traditional task modeling.

Identifying tasks

The first step should be to write essential use cases. An essential use case “is a structured narrative, expressed in the language of application domain and of users, comprising a simplified, generalized, abstract technology free and implementation independent description of one task”. The important thing is that the essential use case is based on the intention and purpose of the user and not the steps required to do a task.

Users and developers should write the essential use cases together, users are allowed in this process to help developers understand their real intentions with using the purposed
system. It might be a good idea to involve expert users more than novice users since it is likely they have a better idea of the tasks that needs to be done. The output of this is to produce tables that describes the intentions of the users and how to system responds. When starting to come up with these essential use cases it is a good idea to let the participants start from the identified roles and ask questions such as “What are users in this role trying to accomplish”. To write essential uses cases for this project the first time should take about 1-2 days. In the following iterations the process should be quicker since most information stays the same.

Each of these tables should be short and not go into details, only focus on high level goals. The reason for doing it in this way is not to try to get stuck early on with preconceptions of how the tasks are preformed. It tries to ensure that both users and developers have a more open mind to different design decisions later on.

<table>
<thead>
<tr>
<th>User intention</th>
<th>System responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get information</td>
<td>Retrieve the information from DB</td>
</tr>
<tr>
<td></td>
<td>Display information</td>
</tr>
</tbody>
</table>

**Mapping the use cases**

As a result of these essential use cases there should be several documents describing the tasks that users want to do. So far the basic tasks have been described and the next step is to identify the connections and overlaps between the essential use cases. To visualize all the essential use cases and the relationship between them a use case map is constructed. The idea behind a use case map is to show the whole system functionality and from that construct a simpler overall model. This map shows the abstract architecture of what tasks are needed to be completed.
The construction of the use case map should be done by the developers since they are the ones with best understanding of the relations in the system. Once all the essential uses cases are done the time spent on making a map should be no more than two days for the first iteration. They are also the ones responsible for the map itself and that it is used in the later step when drawing the architecture of the system. Other reasons to make such a map are to identify and visualize hierarchy of tasks. The developers should identify what tasks are specializations of other tasks and what tasks are extensions of other tasks. An extension task is a task that might be needed to complete a different task. For the developers this kind of insight on to how the tasks relate to each other is crucial in understanding the whole system. These classifications should in the end lead to an identification of the key tasks within the system. These key tasks are used in what is called focal use cases.

Write narratives

At the end on the session when the use case map has been completed the participants should identify where the focus of attention or the central part of the interface should be. These central essential uses cases are the focal use cases that show the most important representative uses of the system. The focal use cases should usually support the focal users that were identified in the role modeling.

Alongside or before the activity of making the use case map it is also a good idea to write use case narratives. They are essentially an abstract description of a scenario and should be very short. The main purpose of writing these is to identify the purpose of the use case. This activity can easily be made in collaboration between developers and users. The task of writing these narratives are relatively strait forward and easy for users to understand. This activity is not to complex and can be done in relatively short time. As a result there should be several of these use case narratives that are the responsibility of the developers to maintain.
6. Content modeling

The purpose of this modeling is to clarify where in the interface the user is operating and what he or she does there and also which tools he or she is using. To perform this rather large modeling the task modeling should be used as foundation. Here the significant point is that some activities that can be implemented by single interaction contexts and some of them by many. This is the essential work for the developers in order to design a dynamic system. For example that one task can be supported by several different tools and in several spaces within the interface. This is what Constantine and Lockwood calls the context or the interaction spaces.

Modeling the contents

The developers meet and start to develop evaluate the use case narratives derived from the task modeling. This is done line by line to determine for each use case what tools and materials (content) will have to be supplied within a given interaction space (context). For instance the content is tools, drop down menus, buttons, check boxes etc. and the context are windows, work spaces etc. So every use case is here separately examined to establish which tools are used to perform that task and also where in the system the task is being performed. As said before, this does not have to be by one tool in one place. The length of this meeting process is of course based on the number of use cases and their complexity.

The result from these activities should be visualized in a clear way, according to Constantine and Lockwood good model visualization is by post-it notes and paper in order to present the content model for the customers in a good way. In the book this is referred to as abstract prototyping. So the next step in the content model is for the developers to present the model for the customers. After this the developers and the customers will have joint meeting where the customers give their comments on the models. Then the developers will change the models taking the customer comments under consideration, the ultimate decision on what changes should be made lay on the project manager.

Context navigation

The context navigation map is about visualizing the complexity of the system in order to simplify it. The map is derived from the use cases and their respectively tools and contexts in order to make a clearer view of the overall architecture by modeling the relationships between different contexts.
This has to be done by a developers meeting based on the post-it notes and paper visualization. In this meeting the developers use those results consisting the tools and interaction spaces in order to sequentialize each use case and define the relationships between different contexts. This step is estimated to take at least one week depending of the number of use cases and their complexity. After finalizing this step of activities the artifact should be a document that contains the whole content modeling with its tools, contexts and their relations etc. In other words all that’s needed to start the actual implementation.

7. Implementing the design

Prototyping

The first step of implementation is prototyping and modeling the layout. All artifacts from the content modeling section such as post-it notes are useful here. Developers should have a meeting or design session to transfer all post-it notes and papers to some passive prototypes. This passive prototype consists of the appearance of the final interface however nonfunctional. Paper is also useful in this section to show some parts of the software for non paintable. They may appear after some code and cannot be painted by visual tools.

Coding

The results of this section are some papers and visual sketches that look like real interface but nonfunctional and useful for programmers to make the real interface by coding. It’s time for the developers to implement parts of the design. They should now draw knowledge from all the previous steps in the project and utilize the artifacts and models in the implementation. They should also focus on their knowledge in human computer interaction and graphical design. The implementation can start directly from the content modeling using tools that allow visual design while programming. The implementation can focus on different parts of the system during different iterations this allows developers to complete independent sections quickly. It is a good idea that what is done in the implementation reflects with the models during that iteration. It is only the developers who are active in this step and they should have at least 2 weeks (in every iteration) to complete their work. The output of this should be a part of the system that is useful for testing in the next step.
8. Testing

In order to improve the software’s user interface some user interface design metrics are useful. These metrics make it possible to evaluate and compare design quickly and economically without first constructing a functioning system, simulation or working prototypes. These metrics are some mathematical formulas to measure different aspects of the software by comparing some values. After doing these measurements developers can improve their interface by using these values and changing different parts of the interface in order to reach better result from these metrics.

The next step is to test the final version in usability lab. In this testing laboratory the final version will presented to different users with diverging backgrounds. This will give the developers some ideas about how most people work differently when they are using the system. This is valuable for them to improve their design in order to generalize the usage.
**Reflection of the method**

The process this report proposes is an application of the usage-centered design approach described by Constantine and Lockwood. As mentioned in the introduction the focus of this approach is the use of models and to describe things in an abstract way. The argument is that models are better at describing how systems should look instead of user input. We feel this approach is a good way to design systems in a top-down approach. The approach should enable new ways of thinking about how to use the system because of the level of abstraction. We feel this approach is more exploratory in the sense that it takes a problem and makes a system that solves the problem in a good way for the user. The more classical user-centered approach focuses more on how to make a system that supports the user in its work.

There are a few drawbacks of this approach according to us and most of them are due to the abstraction level. There is probably sometime in the beginning where the participants in the modeling feel that it is difficult to think technology free and very high level, this might be especially hard for developers since they are used to specification. We also feel that the step from the three models into an actual interface is less clear. The abstraction in the models might be a disadvantage when doing the implementation, sooner or later things need to be specified. The feeling that the process is more engineering driven is shown through the lack of testing. There is naturally testing involved but much less than in a traditional approach. It is somehow assumed that when using these models the result will be good. Another issue is the fact that users are not involved to much in the process, or rather the process is not focused around the users. This might make some users of the system feel not involved and that their opinion is not too important. In the proposed project of the website these users might be the researchers that use the system in their working environment. They might be a lot happier with the end result if they feel that they were an important part of it.

The book is more specific about designing Microsoft Windows application and less about website designing. Therefore we used a document that we found on the book’s webpage that was about designing a webpage. In our report we used both the book and the website to conduct our project. We used the ideas of the book together with the webpage design document on the website and tailored it to fit the conditions of our project. This was a necessary in order conduct this project within the frames of the assignment.
**Estimation of cost and time**

**Cost**

Employees 700,000

Office and equipment 500,000

User activities 50,000

Our report 50,000

TOTAL 1,300,000 SEK

**Time**

At least 130 working days

<table>
<thead>
<tr>
<th>Activity</th>
<th>First Iteration</th>
<th>Following Iteration 1</th>
<th>Following Iteration 2</th>
<th>Following Iteration 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>The first day</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Requirements documentation</td>
<td>8</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Exploratory modeling</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Identifying the users</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Defining the roles</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mapping the roles</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Identifying tasks</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Mapping the use cases</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Write narratives</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Modeling the contents</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Context navigation</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Prototyping</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Coding</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Testing</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>52</strong></td>
<td><strong>26</strong></td>
<td><strong>26</strong></td>
<td><strong>26</strong></td>
</tr>
</tbody>
</table>
Time Line

First Iteration
- The first day
  - Exploratory modeling
  - Defining the roles
  - Identifying tasks
  - Write narratives
  - Context navigation

- Mapping the use cases
- Mapping the roles
- Identifying the users
- Requirements documentation
- Coding
- Prototyping

Following Iterations
- Identifying the users
  - Mapping the roles
  - Mapping the use cases
  - Modeling the content
- Prototyping
- Testing
- Coding
- Defining the roles
- Write narratives
- Context navigation

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52