New standard software product for administrative work, approached in the light of “The usability engineering life cycle”
Table of content

Introduction .......................................................................................................................... 3
Requirement Analysis ......................................................................................................... 4
User Profile ......................................................................................................................... 4
Contextual Task Analysis .................................................................................................... 4
Usability goal setting .......................................................................................................... 5
Platform Capabilities and Constraints ............................................................................. 5
General design principles ................................................................................................. 5
Requirement analysis ......................................................................................................... 5
Design/Testing/Development .............................................................................................. 6
Level 1 .................................................................................................................................. 6
  Work Reengineering ......................................................................................................... 6
  Conceptual Model Design .............................................................................................. 8
  Conceptual Model Mock-ups ......................................................................................... 9
  Interactive Conceptual Model Evaluation ...................................................................... 10
  Conducting the tests ....................................................................................................... 12
Level 2 .................................................................................................................................. 13
  Screen design standards ............................................................................................... 13
  Screen design standards prototyping ............................................................................ 13
  Iterative screen design standards evaluation ............................................................... 14
  Intermediate style guide development ......................................................................... 15
Level 3 .................................................................................................................................. 16
  Detailed user interface design ....................................................................................... 16
  Iterative detailed user interface design and evaluation ............................................... 16
Installation .......................................................................................................................... 17
  User feedback ................................................................................................................. 17
Conclusion ............................................................................................................................ 18
Introduction

According to Deborah J. Mayhew who is the author of the book Usability engineering lifecycle the best way to approach this kind of project is to follow the lifecycle (picture 1) that is shown here underneath and divide the project development into the different topics shown in the picture. Our assignment is to present how we would organize a project constructing a new standard software product for administrative work using the method described in the book by Deborah J. Mayhew. We will not design a fully solution but merely explain how the process would turn out. Our interpretation of her technique applied on this assignment is as follows:

We will accordingly to the book divide the assignment in different subgoals and explain how we should implement them on solving this assignment.
Requirement Analysis

During this first part of the process we define the fundamentals of the whole project. In the following chapter the tasks of producing user profiles and a task definition is done.

User Profile

In this phase we are supposed to identify the different categories from which we later on should define the different work tasks. In each and every category the actors must be identified, defined and a frame created to work within. There are numerous things you need to validate for each category such as religious aspects, experience and education of the user, psychological characteristics and physical strains.

During the user profile phase it’s important to get management feedback to check whether the collected data is accurate. Once all the data is collected and validated it should be interpreted and defined in a definite User Profile, this will be fed right into the Contextual Task Analysis and have direct impact on all design tasks. As for creating a new standard software product for administrative work it’s hard to predefine the actors since they basically varies from case to case, but the task they will preform should be similar. We assume that the service done by the users should be performable on different stations and places, with small differences on each location. The time for this phase takes approximately 3 weeks of team effort and 2 weeks of user involvement, these periods is always counted in man hours.

Contextual Task Analysis

By following the intended users work assignments we’ll be getting the necessary info on how to create the Contextual Task Analysis. And also by giving a query we can implement changes that the user want to change in the present system. Using contextual observations in a combination with interviews or queries we can develop a description of users work tasks. The persons being interviewed should be the ones with most knowledge about the task that should be preformed. Preferable the interviews should not intervene with the work being done and therefore may take some time to collect the data. Interviews often take more time then queries and such but present more accurate answers, taking this in consideration we chose to have interviews with the users when creating the Contextual Task Analysis.

Also in this phase it’s important to distinguish between different actors and Use cases. If the task is preformed by different actors with different subtasks each subtask and actor needs separate data collecting. Different Use cases also needs to be defined and investigated. The outcome of this phase is a model over users task and is directly fed into the Usability Goal Setting and will be documented in the product style guide. Depending on the size of the company that uses the product this should not take no longer then a month work for our team and should not take up more time from the working staff then one and a half week.
Usability goal setting

By referring to the earlier obtained user profile and contextual task analysis we start developing a usability goal containing an overview of what the actual task imply. An important step of obtaining the usability goal setting is collecting the goals of the task to be done which can be divided into quantitative and qualitative usability goals. Typical qualitative goals for this task could be that the design should be conformal enough for previous users to adapt to the new system without extensive instructions. Where quantitative goals could be that the user should be educated in preforming complex tasks before the new system is put to use.

In this particular case the stability and security is highly prioritized while things like easy to learn could get lower priority since the user is supposed to be fairly experienced in the area and looked at as an advanced user. Estimated time for this section is to be about three weeks for the team and about one week for the users.

Platform Capabilities and Constraints

This sections purpose is to define the platform which the task is to be conducted on. Also if the task should be able to be preformed on multiple operative systems, different resolutions or contain multiple languages options. As for the operative system issue to be taken in consideration is whether another operative system could be suited for preforming the task in a easier way. The resolution and display should be able to preform on different hardwares and stations and should at each location be adapted to give a comfortable display of content. Also to be taken in consideration is whether a wish for adding, subtracting or modifying any type of platform currently existing. This information could all be obtained by research of the present system and by adding additional interviews about how to adapt the compatibility to a satisfiable level. This part should not take more then one week to finish for the team and no more then a couple of days for the users.

General design principles

By consulting the relevant Usability Engineering literature we can establish the general design principles, creating a detailed High-level style guide to be used during the whole development. This phase should decide the design standard throughout the whole project. External resources may be a smart thing to add. For example a team for iterative usability testing where a navigational correct prototype is presented and users can test it and make comments about it. May also demand that relevant usability experts are consulted an involved in the project, preferable with experience of the similar kind of project earlier. This sections demands no user interaction and should be fairly short on merely three or four days of team effort.

Requirement analysis

Combining the results obtained in previous segments; user profile, task analysis, platform compatibility/constraints, general design principles, we have created the Requirement analysis for this given task and therefore have all the information we need to start development. When we have reach this far we have to evaluate the previous processes and the outcome, and depending on whether it is a complete solution we can move on to the design development. If we notice that something is missing or need to be modified we institute a iterative process where we preform all previous steps with the modifications and repeat the process until the usability goals are defined.
Design, testing and development
This is the main part of the whole life cycle process, where the actual design and development is conducted.

Level 1
The initial level of the life cycles is done in purpose of developing a initial approach to the design work.

Work Reengineering
Following on from the analysis which will have determined exactly how the organisation currently functions the next stage in the usability engineering life cycle given by Mayhew is work reengineering. As can be expected from the name of this stage it seeks to change the current work models so as to best incorporate the new system being designed to improve the work.

The reason for this is that to simply exactly map the current work structure (particularly if paper based) onto a new system would defeat a lot of the point in a new system- of making work more efficient and effective. This is especially true if the pre-existing system was a purely paper based one. At worst direct mapping of the old system onto the new one could even give a reduction in work capacity.

The work reengineering stage is divided into a fairly simple three step process; first being the actual reengineering, second validation of this and third documentation. Reengineering is achieved via examining the goals of work actions and looking for opportunities to improve them. This should seek to both generally streamline the processes to make them more efficient and to overcome the specific usability goals that have earlier been identified.

As mentioned above automation often requires drastic changes to the paper based standard of working in order to be useful however in automation we must be careful to incorporate every action possible in the real world or else users will be forced to work with the system and seek out work arounds; the opposite of the help and improvement that it should present.

To make sure that the reengineering has been done successfully the book states that validation is necessary. This validation involves using examples of each user type. We must make index cards labelled with all the groups and subgroups of work tasks that have been identified then asking users to sort index cards of low-level tasks into the greater hierarchy.
If the users place the low level tasks in the correct place in the hierarchy then the design has been a success, if the many don’t then it requires re-examining; Interviewing the users to ask where they would place such ‘misplaced tasks’ would seem the obvious immediate solution here.

With the time scale given in the book suggests two staff members in compliance under this part, giving about three weeks of work effort in total. Including reengineering of work models, validation, document work models and then another day or two for validate the total work.
Conceptual Model Design

Based upon the findings of the work reengineering stage the first step in conceptual model design is deciding whether the model should be ‘product’ or ‘process’ oriented. The example of a product based application given in the book is in Microsoft office with the different applications (word, excel, etc…) each being there for the editing of a specific product (file type). A process oriented model meanwhile is needed when there are no clearly identifiable separate products made in the work with the point of the program being to support the work process. An example of this is a product designed to support customer service.

Our task would seem most likely to be a process based system; though administrators would need to produce spreadsheets, documents, etc… their actual work in managing things doesn’t have such a strictly identifiable outcome with the company running smoothly being the goal and one that is not actually done by the system. Administrators would have some products unique to them such as time sheets however the actual scheduling that these time sheets are merely representative of is not such a ‘product’. Likewise managing stock; a stock list is a product however it is not the desired outcome of the system in itself; that is keeping track of stock,

The second step with a process based model is in identifying the task hierarchy. This is taken from the previous stage of work reengineering.

The third stage is designing presentation rules for the processes. Starting with the hierarchy of tasks, the rules for how each level of the hierarchy will be represented visually should then be developed. There should be a fairly uniform presentation across our product especially with regards to different options on the same level; variation here is extremely detrimental to the design of a product. For example the top level could perhaps use icons to select, the second level having a top drop-down menu and the third level a selection within this.

Step four is designing the rules for how the processes should behave. For instance how long a menu will be displayed if not actively selected, how it can me shrunk and closed, etc…

Step five is in identifying major displays. This refers to whether information will be in different windows or all in one window or whether there will be a main window and secondary windows, etc… The example given in the book is for word where the document is in the primary window with the secondary window taken up by various options related to the manipulation of the content of this primary window.

Step six is designing the navigation. i.e. how will you pass up and down the hierarchy and the rules about how you go side to side within it.

The final step is in the documentation of all of the above with various alternate designs. These should take the form of both sketches and the explanatory notes to support them. This is an iterative process and only intended to be a very simple overview with the full design of what the intended product should look like not coming until the next stage.


**Conceptual Model Mock-ups**

According to the book the user interface designer takes the lead role in this task, with those team members which were part of the requirements analysis, work reengineering of conceptual model design tasks providing input and feedback in its development.

Step one of the task is to select the functionality
This means picking a subset of the functionality based upon the most important issues identified in the conceptual model designs. According to Wilson and Wixon (1997), this selection should be based upon:

- Parts that all users will use
- New Features with high visibility
- Features with mission critical outcomes, even if infrequently used
- Older features that have been updated
- Parts of the product interface that the team has concerns about.
- Features involving safety or liability concerns
- Features that will be highlighted in marketing efforts

Step 2 is to sketch the user interface design.
For the functionality chosen a design for a mock up should be drawn. A point is made of saying that this should include the screen design detail only when its essential so as to establish enough context for the user to understand each step in the process of navigating through the interface. Empty windows with some explanatory text are more than acceptable in this stage and are the recommended way of doing things generally, a detailed exact design is not the main focus here however laying out some display content is sometimes fine so as to give a rounder picture if something particularly special is planned there.

Step 3 is the actual building of these mockups. These can be either paper based or actual running prototypes. The book recommends if using paper to use one piece to represent the screen with separate pieces representing specific windows, dialog boxes and pull-down menus which can then be laid over the screen-sheet and each other simulating the system running. If it is an actual running prototype that is created then showing how navigation will work is the key and not the design of the screen or processing of input.

An example level of effort given for the CMM stage is:

Step 1: 8 hours
Step 2: 24 hours
Step 3: 48 hours

80 hours total of work for team members.

This is said to be for a complex application however it involves only two designers.
A low-fidelity paper based mockup would seem to tend to be the best use of time with it accomplishing most of the things a working prototype does but being far easier to make.
Interactive Conceptual Model Evaluation

The purpose of this stage is to get quick feedback from the model design mock-ups. This allows for an iterative design process without the major effort that such would involve at a later level of development.

The Usability Engineer should take lead in this task with the user interface designer acting as an assistant. Other team members can also help in planning and all should be present in the observation of the test. Everyone should be involved as this makes it harder for them to just ignore any results. If test results are bad then new mock up must be created to try and fix flaws. Testing using users in their workspace, asked to think aloud. Major problems identified by users isolated and summarized.

The first decision to be made is whether the important factor in analyzing the design is its ease of use or its ease of learning or if a test for each must be done. If its ease of learning then minimal instructions should be provided to novice users whilst with the latter expert users recruited and given some instruction of where the operations they are used to are located.

The second decision is in identifying the type and range of users to include in the test group- should a broad sample from everyone who will use the program be taken or a tight selection of the most typical users? Representative sampling within these groups however constrained they are must be followed however.

Also to be considered here is what tasks should be focused on in the test- the primary question being a choice between those tasks that are performed most frequently and those tasks that are most important when they are performed (even if this is scarcely).

Wilson and Wixon’s recommendations are again used for the test process as they were in design; most common features, most visible features, etc… should be concentrated on in testing.

Part 3 of testing is the designing of the tasks. The conceptual task analysis performed earlier will be users to give scenarios that can be adapted for tests, the use of a high end user to help design the tests is a recommendation that we should follow given that we know nothing about administrative work. The book identifies two types of task- results and processed based. With results based the users are told the desired end result and given a set starting point but achieve this end result by whatever means they can find to do so. With process based tasks however the task is divided into subtasks with users led down a set path towards a goal. Results based tasks have the advantage of being more realistic; achieving a goal is the reason people use systems, not the process itself. For testing purposes however this could create hard to analyze data as different people have different styles of working and with a system that is new to them things will be especially inaccurate with regards to how they would work should they be face with this system every day. Process based testing though far less naturalistic and perhaps in some ways a bit of a cheat with guiding users down one set task does allow for data comparable between different users to be collected and specific areas of the new design can be tested.
A mixture of process based and results based tasks is the best way in my view. This allows for both a general insight into how users would prefer to operate and a more specific guide to differences between users and the system’s ease of use.

Another key choice to be made is in whether the tasks should be dependent or independent on one enough; that is should one task lead into another or not. Dependant tasks are more realistic however they have a downside in that if the user fails one task they will have failed every subsequent task before they have even started. Again a mixture would seem the best method; though dependent task chains should not be made too long lest one of the early tasks proves difficult thus rendering the latter ones redundant.

The fourth step outlined in the book for the conceptual model evaluation is the testing and designing of the materials to be used in the test and planning exactly what will happen during the test. It is recommended here that a test should take between one and three hours however elsewhere it says tests should take no longer than two hours, lest the user grow tired and also forget what happened early on in the test rendering feedback less useful. The materials to be provided here are a introduction for the test subjects and the observers, instruction cards for the different tasks, the pre and post test questionnaires, permission forms, data collection sheets and if the test is of ease of use a basic training manual for the user to examine before hand.

The fifth step is around organizing the actual test environment. As mentioned earlier doing it in the standard working environment of the finished product is desirable however here it says to do it in a lab; doing the early attempts in a lab and the latter, more exact stages in the work environment seems the best course of action for us to take.

The sixth and seventh stages are the recruiting of pilot users then the running of pilot tests with them (test tests so to speak) The purpose of the pilot test is to make sure you’ve set up your test well and that it will give you accurate data. Step eight is then using the pilot test data to refine the test and then step nine recruiting and scheduling the actual test subjects.

One thing the book makes a large point of is not underestimating the time and effort required in arranging test subjects; something which I know to be true from past experience in usability testing. The effort can be reduced if you are working within a company, this will provide you with a set of users from the start. If you do not have this though things are hard and methods such as offering incentives must be used.
**Conducting the tests**

What the book says about running the tests covers much the same ground as what it says to plan to do in the planning stage. Stick to set procedures to try and provide a neutral test to each user and do not help them when they struggle.

Observations of everything the user does should be recorded on data recording sheets; more than just whether they are successful or not this should also cover how long it takes them, how they do it and the ease with which they do so.

Once the test data is collected the first step is to summarize the data collected- i.e. the time the users took, the route they followed through the prototype, how many times each problem arose, etc…

Next this data should be analysed to show where the problems in the design are. e.g if users keep experiencing the same problem why is this?

Step 4 then follows directly on with the finding of solutions for these problems that have been identified. These changes that should be made to correct the program should be ranked in severity with those that are hard to implement and least critical being of a lower priority than those that are critical and can be easily changed. Different alternative solutions to problems should also be provided, especially for major problems. Often there will be one ‘quick fix’ which is easy to implement and should bridge the problem and a more definite fix which is difficult to implement but should utterly eliminate the problem.

The final stage of testing is documenting the results. This is mainly necessary in large projects where all team members aren’t present at the test and a way of easily showing them what was uncovered in the test is required.

Even if everyone on the project is involved however some record of the tests should be made at least for future reference. Enough is not known about the project that we are theoretically applying this book to so as to say how much effort should be put into documentation.

The level of effort outlined in the book for an example test is:

- Designing/making the test materials: 32 hours
- Designing/setting up the test environment: 8 hours
- Run pilot test: 8 hours
- Revise test materials: 8 hours
- Run tests: 32 hours
- Summarize test data: 16 hours
- Documenting results: 40 hours

Total: 144 hours.

Which shows testing to be a very large part of any project and quite understandably so; only through testing or enormous luck in getting everything right first time can a project be a success.
Level 2

Overall design of the user interface is done in this part of the process.

Screen design standards

In efforts to provide a consistent and simple graphical user interface to the we develop screen design standards. Thus providing the necessary tools for the actual developers the guidelines for building the software in a manor that the end user is familiar with the interface, and doesn't get confused when all of a sudden the interface changes its look and feel.

The screen design standards is a set of regulations that apply to all the displays and interactions under the graphical user interface. Some of the rules are directly an adaptation of norms applying to our specific interface design standards, which is typically presented during the task of presenting general design principles. These design principles could for instance specify the behavior of input components, color scheme and fonts. In this case these design principles could be derived from common standards, since the software needs to provide a familiar feeling to whomever might access the system.

Developing the screen design standards can be carried out by a single user interface designer. On the other hand, with the help of users we can get a more detailed version what design standards might be appropriate to include and exclude. However, the fact that we are developing a standard software the user input in this task will not change the screen design standards fundamentally, this because it most likely will be based on common standards. Not said that users cannot provide valuable input, on the contrary they can provide invaluable information about their work that could imply implementing a certain standard.

The work effort required by this task is dependent on the number of user involved. But as pointed out their participations will not cause that great effect, and is thus not needed in great numbers, so let us say two end users. Together with the interface designer that makes three people and they perhaps need a week together to compile a document specifying the screen design standards.

Needles to say, the product of the this task is the actual document specifying the different design standards. This document could be divided in to different categories, defining subsets of standards, which each specify behavior of components. As an example, in this case the standards regarding input for birth date could be specified to implement three drop down boxes representing year, month and day, accordingly. Now that standard must be implemented by all input screens, to ensure user familiarity.

Screen design standards prototyping

To ensure that we can evaluate our design standards, previously described, we create running prototypes illustrating the standards to provide feedback for further iterative design, described in the next part. These prototypes can be presented both to management and end users as to provide a actual result, and a sort of summary, of all the work up until this point.

The main guideline for this task is delimitation, since it is not reasonable to prototype all of the
screen design standards, we will only focus on one of the subsets describing a delimited part of the interface. In order to achieve the greatest effect with our prototype we try to illustrate a large task process with as few design standards as possible. There are of course different approaches to actually building the prototypes, and the appropriate method is chosen depending on the detail needed. If the need is great then a actual application should be appropriate, however if the demand for detail is low a simple drawing should be sufficient illustration of the design standards. In this case the need for evaluation, as well soon see, calls for greater detail. Hence a working prototype is important for coming evaluation.

Alas, in this case it is problematic to choose a suitable task to illustrate, but it should be manageable if we have conducted a task analysis with brilliancy. Because then we can choose the most vital task which is in need to be implemented, in other words the task that will be used most often.

Only a couple of days work should be necessary to compile a prototype which complies with our chosen task and screen design standards. The main actor in this task is again the interface designer who provide input to a experienced prototype builder, that can be any of the existing personnel on the project or an external consulting engineering.

**Iterative screen design standards evaluation**

Both the screen design standards document and the prototype provide as input for the task of iterative evaluation. This is a means to validate the screen design standards and their actual impression on the end user. As in the prior task, we could not evaluate the whole prototype, or the whole design document for that matter. So, once again we need to delimit our scope of evaluation, and then mainly focus on either ease of use or ease of learning.

Since our subjected users will most likely perform the same tasks over and over again, the need for ease of use supersed the need for ease of learning. Therefor it will provide a better evaluation if the focus lay with the ease of use. This choice will also provide for easier measurements of the screen design standards, since we can measure the time spent doing a task. Then we can analyze and see if the interface is somehow prevent the user from performing the work task as efficiently as possible.

The evaluation process should be conducted along with a reference group of user, and the testing should be videotaped for further examination and analysis. Focus should be on a couple of core task that the user normally performs and they should be iterated to simulate a regular work environment. Hence it is very important to provide suiting test data, for instance if the intended users are working with booking of rental cars, then the test data provided to the user should be about that subject, and not about books.

During the evaluation it is important to orally ask the user about what they are doing, why they are doing it and how they think it is working. This is imperative to get the full picture of the user experience, since it is the closest we can get to the users thoughts. This form of questionnaire along with the video recording constitute a great material for analysis, where one could find flaws in the interface.

To maximize the result of the user evaluation one could conduct pilot evaluation along with a small number of users, lets say one or two. Here we get the opportunity test our approach to the testing,
this information will mainly provide the basis for small tweaks and fixes in our approach, maybe the length of the test or the questions asked.

Effort wise the evaluation is probably the most demanding task, next after the actual development of the standards. However, it is crucial to evaluate so that we can iterate and refine the standards to best fit the work task and the users.

At this point we need to evaluate if the usability goals, defined in the requirement analysis is met. If not, we need to iterate and improve the screen design accordingly. If yes, we can proceed to develop the detailed user interface.

**Intermediate style guide development**

Under this intermediate checkpoint we compile the documents developed so far, in total forming a style guide for the coming tasks. Included in this style guide we might include an overview of the desired functionality, that we found under the task analysis. Moreover we would include the user profile, platform capabilities/constraints, usability goals and the models design. Mayhem suggests that the contextual task analysis should be included. But this is rather redundant, and unnecessary, since the context is the framework that upholds the whole project and should be the red wire running through the rest of the artifacts created.

Since it is a intermediate style guide it will not contain a fully feathered version ready for implementing, as we will see in later tasks. The backbone of the style guide could contain high level definitions, along with examples and explanations of the advantages and disadvantages of the chosen approach. Moreover, this could be divided in to three separate categories, principles, guidelines and standards. Where the principles contain information about what information is underlaying all major design decisions. Guidelines present tips and tricks for conforming with the chosen design decisions and styles. Finally the standards are the enforced and more specific implementations that must be followed, and which is directly applicable.

The compilation should be done by the project leader and reviewed by all participants, then if accepted it will be the style guide for the coming work. It should not take more than a day to compile it and then another day for reviewing.
Level 3

During this last part, before actually deployment of the product, the detailed design of the interface is conducted. After this we have a final product, ready for delivery.

Detailed user interface design

Once we have a definite collection of screen design standards and the accumulated style guide we can start with the detailed design of the user interface. This task is designed to produce a complete user interface that ensure that our users optimize their work and satisfies users interaction with the system.

Based on the work flow we start by designing the pathways of dialogs, input queries and feedback. Then continue with the complete design of various menus and navigational tools. Since the major task for the user is administrative work, it is prone to contain process driven task. In other words it will often be several steps that need to be easily traversed and a high overview need to be present at all times. Hence the need for easy navigation is a large part which will need much attention during the detailed design. This is just a glean from what the detailed design is all about, but in general it is just what the topic says, a detailed design of the interface. In short based on the previous results and style guides.

Depending on the platform and the overview necessity of the work task dealt with this task might require as much as three or four weeks work, including a bunch of developers. These developers will produce the detailed user interface based on the style guide and the instructions given by the user interface designer, but mainly the documents provided. The number of developers are hard to grasp, but a there will perhaps be about five people, as an development group. This sounds like an awful lot of time, but keep in mind that this is the core task of the whole process of designing the interface and must not be ignored.

Iterative detailed user interface design and evaluation

In accordance to the nature of this whole model, this part is also iterative oriented. Thus leading to refinement and improvement, based on the feedback from users. The input from previous tasks and the rigorous evaluation assures us that the detailed user interface is the product expected and wished for. This leads to the conclusion that the evaluation, no matter if it leads to an iterative step or further to the next task, wont lead to any ground breaking insights. So in other words during this step we will not come to terms with the detestable fact that we have done it all wrong.

On the contrary, the flaws that we might encounter address functionality issues that has not been handled earlier in the process. And this is of course hard to seize these shortcomings earlier in the process, but as pointed out these problems may only refer to very minor changes. Furthermore, these minor changes of course apply to functionality and not the user interface as such, because again due to rigorous testing and evaluation. If minor flaws are detected we iterate and adjust these problems, first in the style guides, then in the actual application and finally the test plans and again this step of evaluation. However, if a major flaw is detected, the whole process is in some sense restarted. In this turn we focus on the requirement analysis and then sort of review the rest of the process to once again evaluate the work conducted. In other words it is just a matter of correcting
the error and to in the process verify that the changes are done accordingly and that this does not interfere with the already developed product.
But normally, as attempted to illustrate, this last step only brings attention to minor flaws, which are easily adjusted. Then the interface is ready for deployment, without any major issues.
This task should be include all team members and thus leads to significant level of effort. However it may be accomplished during the course of a week, maximum. Because everyone is familiar with the user interface, the detailed design, and the standards, guidelines and regulations behind it all. Therefor it will be easy for everyone to provide input ant commit the work to the final refinement.

Installation

Now that the development phase is complete the actual installation can begin. Since the system is brand new we cannot be a hundred procent sure until the system is in place and have been tested with the users who is going to use it. Therefore after the system is in place a new phase is being initiated where the users may provide feedback on the system and an investigation to check whether the systems provided a better solution then the last one.

User feedback

Once the product have been installed we need to get actual feedback from the persons who’s going to use the system. This is to be done after the systems been in production for some time and should be evaluated by experienced users who has been in contact with the system since production start. The method to get this information is preferable extensive interviews or queries. If we detect anything that needs to be changed we schedule a update and introduce the changes. The questionnaire made to test the usability should be thoroughly documented analyzed. This will take about two and a half weeks of team effort and about two weeks of user effort.
Discussion

The total work effort of the whole life cycle is estimated to be around twenty four weeks. It might sound like a long time but it is completely necessary to account for all the different steps in the process. Any negligence to some part of the process may be fatal to the overall output and user experience, which is completely contra productive in the sense that we then ignore the end user. Which in turn is against everything that the life cycle, and the whole purpose of user centered design, stands for.

All in all the process described by Mayhem fits the problem quite nicely, even so some modifications were needed, both crucial for making it work but also for improvement.

After the life cycles is completed we hopefully have a satisfied customer, both stakeholders and the end user. But as described in the all processes the extensive work ensuring and evaluating that our approach, models and work actually works and results in a fitting end result.

Since constructing a new standard software product for administrative work is a fairly open frame to work from our solution is very similar to the books strategy. The initial problem description can be applied on most problems and therefore almost all of the books solution strategies need to be covered.

The approximate time frame we followed should change a lot depending on project size. The one we covered here is an extensive develop phase of a large project, so if the project size is reduced the same goes for the design phase.