Usability engineering

Usability is not like a garnishing or topping that you can put on after the order is ready on the chef's table, neither is it like a paint which you can put on the building to hide the internal masonry faults. It is an all pervasive, all encompassing, rather a holistic approach to thing, something that has to be inculcated and practiced at every phase of the development process and even after.

In software terms it isn't something you add to a product once you are done with the development. It's something you should ***design*** into the product. A methodical approach to producing a user interface and putting in due thoughts to ensure that the end product has higher usability indexes is **usability engineering**.

The usability guru **Jacob Nielsen** has written a book (Jacob Nielsen: *Usability engineering*) on applying systematic methods throughout the development life cycle to increase ease-of-use for software, websites and other UI's. In his book Jacob talks about what comprises usability, key concepts of usability, some common jargons and mistakes associated with usability. He also talks about the various usability practices like thinking aloud, heuristic evaluation, usability testing etc. He also details the future road map for usability as a practice and as an idea in general.

The key word in designing usability is the **user** as without doubt s/he is the most important stakeholder in the entire development process and beyond. The user is the one who would be spending in the dollars to buy the product and they are the one who would be employing the services of the application for their needs. Customer is the king and he generally knows what s/he wants so you have to really *know* the user before making any decisions about the design. If you only assume what the users want, you might end up with a product that nobody needs. You need to spend a lot of time and effort in ensuring you have a true picture of the target audience and their expectations.

Usability engineering involves several methods, each applied at appropriate times.

*** Gathering requirements**

This phase/method include task analysis, user analysis and environment analysis.

In ***task analysis*** you determine cognitive and other characteristics that are required of users by the product. These can be, for example, pre required knowledge and cognitive (the psychological result of perception and learning and reasoning) loading. You need to understand the psyche of the user and their behavioral patterns and other aspects of their usage.

User analysis helps you describe who will use the product. You need to define the intellectual abilities, cognitive processing abilities, previous experiences and physical capabilities of the user. This would help in understanding what sort of decision making options should be given to the user as advanced settings and other options might/might not be understood by the user with limited technical grounding. Also the target audience physical capabilities would help you in deciding the input-output options to provide. For instance if the user is visually challenged you might think about speech recognition, Braille etc as a mode of communication etc.

Environment analysis makes you consider the physical, social and user support environment. In other words you need to determine what the user needs to do and find out the users way of doing things and the context the product will be used in. This would help you in deciding the kind/mode of information display and other criterion, for instance if the application is likely to be used in office environment you might consider the kind of media you want to play and at what volume etc.

*** Developing and testing prototypes**

This is just what it sounds like i.e. making prototypes based on the requirements and testing them. You need to find out if the prototypes work like wanted, if there are some features missing or if some features are misplaced. Prototypes can range from extremely simple sketches (drawings and post-it notes) to full systems that contain nearly all the functionality of the final system (low-fidelity prototypes/high-fidelity prototypes). It would be prudent to design and validate the prototype in tandem with the end user or a subset of the target audience if possible to minimize the element of surprise and shock at a later point in time. Prototype validation is indeed a key component of this exercise and something that is generally overloaded by the team.

*** Evaluating design alternatives**

This helps you find the usability problems in a user interface design. Evaluation can be done with heuristic evaluation, for example, where you go through a list of usability requirements and see if they are implemented in the design. The list can include qualities like visibility; learn ability, consistency and flexibility. An iterative approach to evaluating the design and looking at alternatives is the best way to go about it as this would ensure that the best ideas get included into the system and any variance is detected at an early stage thus minimizing the impact on cost and timelines.

*** Analyzing usability problems**

(C) Copyright Nokia 2014. All rights reserved.

This will (hopefully!) help you avoid the same problems in your next design. You have to get to the bottom of the problem. Why did it happen in the first place? What caused the problem? Understanding the common pitfalls and possibly documenting the best practices which can be used as a template for future work would reduce the rework effort for other engagements in the future.

*** Proposing solutions**

Every time you are encountered with a problem think of the solutions and don't delve on pin pointing the blame for the problem. Every problem should be approached of as an opportunity to try out something new and try out new solutions. Think of possible ways to solve the problems? Refine your design and test it again. If needed make another prototype and test the solutions with it so that you have a validation check done at that point itself. Such an approach would further minimize the need for change requests at a later point in time.

*** Testing an application with users**

This would give you valuable information about features that are too difficult to use, in a wrong place, not needed or perhaps missing. There are a wide variety of different testing methods to use: thinking aloud, field observation, questionnaires etc. The user group selected for doing this should not be involved in the development process in any way; this would ensure that you get an honest and impartial feedback.

Although Jacob's book is considered an almanac of sorts for usability it also has its share of bouquets and brickbats.

Many people thought found this book to be the answer to problems, however the book was written in the 'purest' environment. This book avoided the 'problem' customer who does not know what they want. Project management is ignored and the recursive 'loops' found in development.

The books main point's works well in small teams of three people, but this does not reflect the size of teams and the control needed in commercial projects. To be successful many people must test and give feedback on usability

Some additional resources about usability, its importance, the common issues in usability designing etc.

Link

Designing usability

Why Usability

General usability issues

---- Edited by Mayank on 17/06/2009 ----

(C) Copyright Nokia 2014. All rights reserved.