Volume : 2 | Issue : 9 | September 2013 • ISSN No 2277 - 8179

Research Paper

Process Life Cycle of Usability Engineering

Computer Science

KEYWORDS : Human Computer Interaction, Life Cycle, Usability Engineering

Sonam Mittal — Computer Science & Information Technology, B.K. Birla Institute of Engineering & Technology, Pilani, Rajasthan, India

Reena Saini — Computer Science & Information Technology, B.K. Birla Institute of Engineering & Technology, Pilani, Rajasthan, India

ABSTRACT

Usability Engineering is a field that is concerned with Human Computer Interaction (HCI). Usability Engineering is used explicitly to judge the success of a product in terms of its usability and user friendliness. A user friendly interface is one that allows user to effectively and efficiently complete the tasks for which it was designed for. This paper presents usability engineering process that can be taken as a product development process and aims at effective and efficient working of the interface.

INTRODUCTION

Usability Engineering concerned with HCI. HCI, Human Computer Interaction, is the combination of three Human, Computer and Interaction. By Human we may mean an individual user, a group of user working together, each dealing with some process. By Computer we mean any technology ranging from general desktop computer to control system or an embedded system. By interaction /interface we mean any communication between a user and computer. A key aim of HCI is to understand how human interact with computers and represent how knowledge is passed between the two. Usability Engineering is an approach to performed development i.e. based on customer data and feedback. Usability Engineering begins in the conceptual phase and contextual inquiries to understand the functionality and design requirements of the product. Usability is iterative design and evaluation to provide customer feedback and usability of product's functionality throughout the development cycles. Usability Engineering emphasizes direct observation including- knowing your customers, knowing other products, customer feedback and throughput of development cycle and direct observation of customer. Usability Engineering offers many benefits: reduction of failed product, reduction in development time, cost reduction in training and increased sales and increased productivity.

PRINCIPLES TO SUPPORT USABILITY ENGINEERING

Principles are abstract designed rules with high generality and low authority. Principles are derived from knowledge of the psychological, computational and sociological aspects of the problem domains and are largely independent of the technology. More abstract designed rules are general principles which can be applied to design of an interactive system in order to promote its usability. There are some principles which are broadly used in usability engineering. These principles are first divided into three main categories: Learnability, Flexibility and Robustness.

A. Learnability

It is the ease with which new users can begin effective interaction and achieve maximal performance. There are certain principles that affect Learnability:

1. Predictability: It is the support for the user to determine the effect of future action based on past interaction history.
2. Synthesizability: It is the support for the user to assess the effect of past operations on the current state.
3. Familiarity: It defines the extent to which a user's knowledge and experience in other real world or computer based domains can be applied when interaction with the new system.
4. Generalizability: It is the support for the user to extent knowledge of specific interaction within and across application to other similar situations.
5. Consistency: It describes the likeness in input-output behavior arising from similar situations or similar task objectives.

B. Flexibility

It defines the multiple ways in which the user and system exchange information. There are various parameters that define flexibility:

1. Dialog Initiative: It is the freedom of user from artificial constraints on the input dialog imposed by the system.
2. Multithreading: It defines the ability of the system to support user interaction pertaining to more than one task at a time.
3. Task Migratiability: It is the principle which defines the ability to pass control for the execution of the given task so that either it becomes internalized by the user or by the system shared between them.
4. Customizability: It is the modifiability of the user interface by the user or the system.

C. Robustness

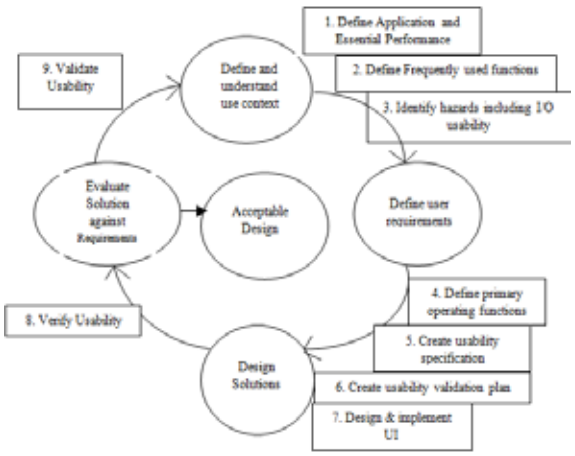
It defines the level of support that is provided to the user in determining successful achievement. Certain parameters define the robustness of the interface developed by the user, which are:

1. Observability: The principle which allow the user to evaluate the internal state of the system from its perceivable representation at the interface.
2. Recoverability: The ability of the user to take corrective action once an error has been recognized.
3. Responsiveness: It measures the rate of communication between the system and the user variation in the response time will impede anticipation exploited by motor skill. Since the purpose of an interactive system is to allow a user to perform various tasks in achieving certain goals within a specific application domain. [5]

LIFE CYCLE OF USABILITY ENGINEERING

A process model provides the structure for usability engineering to be crafted and tailored to the needs of a specific project as an instantiation of the model [2, 7]. *Figure 1* describes the different phases in life cycle process of usability engineering starting from defining and understanding user context to evaluation of solution against requirements. In the next phase user requirements are defined and Usability specification is created like software requirement specification. Now create the validation plan for it and develop the user interface, verify and validate the usability of design solutions against requirements.

Volume : 2 | Issue : 9 | September 2013 • ISSN No 2277 - 8179



Software Development Life Cycle

System Development Life Cycle (SDLC) is the most well-known of the so-called phase models. A system development lifecycle (SDLC) is a general process for developing information systems. It provides a framework for the tools, techniques and methods needed to build and implement systems that adequately address business needs. A SDLC defines the phases and tasks that are essential to systems development regardless of the size or scope of the impending system. While there are many versions or models' of the SDLC, they differ only in terminology and are more often alike than dissimilar. Figure 2 shows the different phases of SDLC.

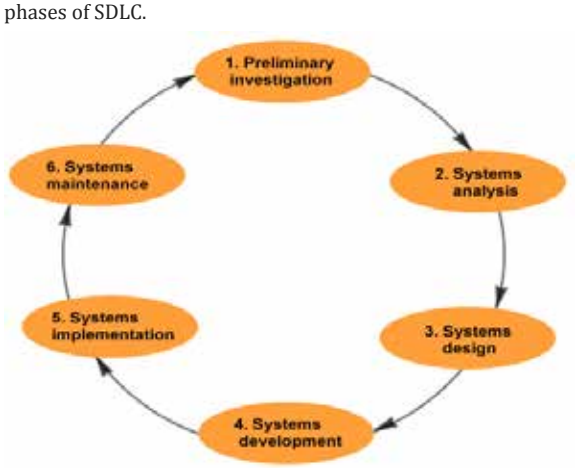


Figure 2

Difference between SDLC and Usability Engineering Process Life Cycle

SDLC consists of different phases beginning from requirement phase. At the end of each phase evaluation is done. On the basis of evaluation, usability of the product is determined. SDLC is a process for developing software. At the end of each phase of SDLC, usability of the product is measured. On the basis of that, the successful development of the project can be determined. User-centered design is just another word for usability engineering. Software engineering is far from being a mature discipline; it certainly needs further research and development as regards user- orientation. User- centered design is a philosophy opposed to the system-driven development philosophy that is the traditional way of seeing and doing things in software development. Usability engineering has become the way of thinking about user-centered design in the software engineering community. Usability engineering focuses on requirements and evaluations preserving, perhaps, a technical, engineering oriented attitude to software development. User-centered design, on the other hand, addresses designing with the users.

The differences between human-centered and Technology - driven development

Software Development Life Cycle philosophy	Human-centered development(Usability Engineering) philosophy
Technology/developer-driven	User-driven
Component focus	Solution focus
Individual contribution	Multidisciplinary teamwork including users, customers, human factors experts
Focus on internal architecture	Focus on external attributes (look and feel, interaction)
Quality measured by product defects and performance (system quality)	Quality defined by user satisfaction and performance (quality in use)
Implementation prior to human validation	Implementation of user-validated solution only
Solutions are directed by functional requirements	Understanding the context of use (user, task, work environment)

Benefit of usability engineerng

Enhanced reliability and efficiency of the system will have economical effects by Decreasing user support costs, and time-consuming training investments. Usability Engineering will decrease technology development time and costs: late discovery Of serious flaws to a system will render the necessary re-engineering time- and Cost-intensive. From Usability Engineering Users will-

- Experience satisfaction instead of frustration
- Achieve goals more effectively and efficiently
- Not waste time and energy
- Easily learn to handle the system

Providers/producers/developers will benefit from –

- Reduced financial costs
- Efficient design that adds value, not frills
- Fewer revisions
- Reduction of support costs
- Increased productivity
- Increased accessibility to maximize the potential audience
- Increase in use
- Happy and loyal customers
- Reduced development times
- Avoidance of unnecessary features

Problem with Usability Engineering

The problem with usability metrics is that they rely on measurements of very specific user actions in very specific situations. When the designer knows what the actions and situation will be, then she can set goals for measured observations.

Another inherent limitation for usability engineering that is it provides a means of satisfying usability specifications and not necessarily usability. The designer is still forced to understand why a particular usability metric enhances usability for real people.

CONCLUSION

The world is full of useless and frustrating software with poor functionality and user interfaces. It can be improved with inclusion of usability aspect so that users do not suffer needlessly.

REFERENCE

1. Shackel, B. 1990, Human factors and usability. In Human-Computer-Interaction: Selected Readings (Preece J. and Keller L., eds) Hemel Hempstead: Prentice-Hall. | 2. Nielsen Jacob, "Usability engineering" Boston Academic Press, 1993. | 3. Nielsen J., "Usability engineering life cycle", IEEE computer society, Volume 25, Issue 3, March 1992 Page(s):12 – 22. | 4. Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, " Human-Computer Interaction: Overview on State of the Art" international journal on smart sensing and intelligent systems, vol. 1, no. 1, march 2008. | 5. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human Computer Interaction", third edition. | 6. Jean Scholtz, Ben Shneiderman, "Introduction to Special Issue on Usability Engineering" Empirical Software Engineering, 4, 5–10 (1999). | 7. J. Whiteside, J. Benett and K.Holtzblatt. Usability Engineering our experience and evolution. In M. Helander, editor, Handbook of Human-Computer Interaction. North Holland, Amsterdam, 1988.