Human Persuasion Integration in Software Development Lifecycle (SDLC)

Muhammad Shakeel Faridi[1], Tasleem Mustafa[2] and Fahad Jan[3]

[1] MS(CS) Student of Computer Science, University of Agriculture Faisalabad, Punjab, 38000, Pakistan

[2] Chairman/Assistant Professor, Department of Computer Science, University of Agriculture, Faisalabad Punjab, 38000, Pakistan

[3] Lecturer, Department of Computer Science, University of Agriculture, Faisalabad Punjab, 38000, Pakistan

Abstract

Of course, people have different opinions about what is attractive. They have different views of what's attractive; designers need to understand the aesthetics of their target audiences when creating a persuasive technology product. The more visually attractive the product is to its target audience, the more likely it is to be persuasive. Similarly during development software product, software engineers are facing lot of problems. End user feels reluctant to provide the whole his persuasion during requirement gathering phase. In this paper the author tried to propose a human persuasion (HP) model that will embed during the software development lifecycle (SDLC). In this way a powerful solution can be achieved within available resources.

Keywords: *Persuasive Technology, Software Product, Software Engineering, Requirement Gathering, Software Development Lifecycle*

1. Introduction:

Human persuasion techniques are now well recognized as important milestone in software development. E-commerce, Graphic User Interface (GUI), mobile devices oriented application are the most common examples of human persuasion techniques. The study of users' attitude and behavior has a long history in Information System research. Theories from users' psychology have been widely used for predicting users' intentions and behavior [1].

Studies and research have shown that 80% of total maintenance costs are not related to technical bug it is just related to user's problems with the system [2]. Among them, 64% are usability problems [3]. Human-centered design (HCD) philosophy and related usability engineering (UE) methods provide powerful solutions to solve the poor requirements, lack of user involvement, requirement incompleteness, changing requirement, unrealistic expectations and unclear objectives [4-6]. Knowledge and theory is still limited about how to efficiently and smoothly incorporate UE methods into established software development processes. While standards such as ISO 13407 (Human-Centered Design Processes for Interactive Systems) provide a detailed description of the major Users Centered Design (UCD) activities as well as strategies to assess an organization's capability to adopt HCD practices, they lack guidance on how to effectively integrate usability in a specific development team, project or context.

2. Why Human Persuasion Software Engineering

Interactive information technology designed for changing users' behavior or attitudes is called persuasive technology [7]. Traditionally persuasion means human communication designed to influence the autonomous judgments and actions of others [8]. Human Persuasion (HP) emphasize that the end product is designed as per user requirement. It focuses in system development towards the goals, needs and wishes of the users on top priority.

The process associated with software development, the methods used to analyze, design and test computer software, the management techniques associated with the control and monitoring of software projects and the tools used to support process, methods, and techniques[9].

2.1Requirements from the user's side

Users are precious and frequently underused resources in the software development process. Users are the only ones who really have the potential of explaining how

they interact with the system and how they use the system as a support to achieve their desired tasks.

Active contribution of the users can improve the user's ability to develop their understanding of the possible of the technology and also helping them to see how their tasks could develop and change due to the impact of the new technology. Several studies show great benefits in involving the users actively in the design process [10-11].

2.2 Requirements from the developer's side

Developers have for a long time been suffering from the limitations in existing tools to turn requirements into designs that will work under the limitations that existing technology provides.

Developers want to do a good job; they want to produce a system with the highest possible level of usability. But developers also work under restrictions: restrictions in time and budget, and restrictions imposed by the limitations of the technology and the methods used. Therefore it is natural that developers can express some hesitation towards too much of their time being consumed by cooperation with the users [12-13].

3. System Development Lifecycle in Software Engineering.

The aim of software engineering is to produce a quality work that will construct according to the stakeholders' requirement.

The systems development life cycle (SDLC) is a abstract model used in project management that describes the stages concerned in an information system development project, from an initial feasibility study through maintenance of the desired application.

To accomplish the desire task, software developers are definitely followed SDLC methodology. These methodologies have been developed to guide the processes involved, including;

1. Waterfall model
2. Spiral model
3. Iterative model
4. V-shaped model
5. Extreme model
6. Rapid application development model

Most familiar and probably used is waterfall model which is mostly used in many fields of life like education, agriculture, commerce, medicine and industries etc.[14-16].

These process models have different advantages and disadvantages but their features are most corresponding in software development program.

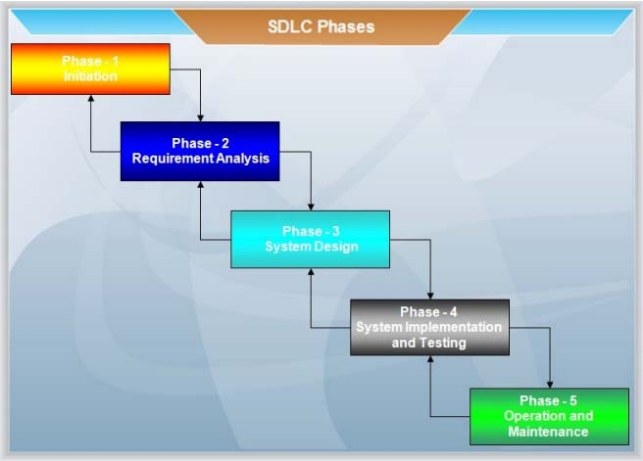

Fig. 1 SDLC Phases

The phases involved in Fig. 1 described as following;

Phase 1: Initiation

Preliminary investigation is conducted in Phase-1 where alternative solution, costs, benefits for the organization, organization's objectives and abstract scope of the project finalized.

Phase 2: Requirement Analysis

In Phase-2 the goal of the project, functions and operation of the ongoing application and also analyzes the functional requirement of the system. In this Phase we also study the end-user requirements.

Phase 3: System Design

According to the requirement elicitations, the system goes toward design phase. In this Phase the design of screen layouts, business rules, activity diagrams, detail operations of the project, pseudo code and other documents are designed.

Phase 4: System Implementation and Testing

In Phase-4 the designed system enter into different tests like performance testing and stress testing then developed software is implemented for required output.

Phase 5: Operation and Maintenance

In operational phase of the life cycle the users utilize and operate the final product for required output. The developer provide the maintenance where required until the final product accepted by the user.

Proposed Model

Human interactions with the system are most important process that leads the system towards success. There are certain systems where persuasive technology can be useful like web application, e-commerce application, mobile technology etc. are strongly dependent with the human persuasions because user can operate it easily. Moreover, persuasive systems motivate end users to achieve their desire goals. Software which is inconsistent, poor quality, unreliable, dissatisfaction of end-user, and developers consuming much time and cost to build as per user requirement leads the system towards failure. Literature and review shows that while developing system users usually focus the function/business requirements and are often neglecting and poorly understood the user requirements.

There is a need to develop a system which is fulfilling the function, non-functional and human persuasion requirements. Fig. 2 proposed model will help the Project Managers regarding human persuasion while developing the system.

Fig. 2 Human Persuasion in SDLC Phases

Information technology always manipulates people's feelings and performance. Without human persuasion a system cannot be successful. Successes of the projects are fewer than failure of the projects. However, studies about persuasion, relying on credibility of the product for occurrence, are less recognized. Our goal is to promote the introduction of persuasive model in software development through psychosocial theories, especially in the Business Intelligence area.

In Fig. 1 the author discussed the overview of SDLC phases which are commonly used in software development. Fig. 2 shows the proposed human persuasion in SDLC Phases. Human persuasions consists of several aspects like business rules (production life cycle, cultural and political environment), quality attributes (performance, reliability, accessibility, maintainability), external interfaces (user interface, color combinations etc.) and development constraints (time, cost and human resource) that do not describe the functionality of the system.

Conclusion and Future Work

While developing the software projects, project managers ignore the target audience and their persuasion about the project. Similarly software engineers are facing lot off problems while developing the project. End users don't accept the project easily because the project doesn't fulfill his/her requirements. These all aspects go the projects towards failure. In this paper the author tries to propose a model which helps the project managers while studying the project requirements. The project managers chop up the requirement phase into two phases 1) functional requirements 2) non-functional requirements. In non-functional requirements, he will collect and embed the all human persuasion requirements which will lead the project towards success and the failure of the project will be minimize.

In future work the author will try to apply the proposed model in textile sector to ensure its usability and reliability in real scenario.

References

[1] Rawstorne, P., R. Jayasuriya, and P. Caputi. An integrative model of information systems use in mandatory environments. 1998: Association for Information Systems.
[2] Boehm, B.W., Software risk management: principles and practices. Software, IEEE, 1991. **8**(1): p. 32-41.
[3] Landauer, T.K., H.E.-B. Project, and A.C.o.L. Societies, The trouble with computers: Usefulness, usability, and productivity. 1995: Taylor & Francis.

[4] Norman, D.A. and S.W. Draper, User centered system design; new perspectives on human-computer interaction. 1986: L. Erlbaum Associates Inc.
[5] Vredenburg, K., Building ease of use into the IBM user experience. IBM Systems Journal, 2003. **42**(4): p. 517-531.
[6] Mayhew, D.J., Business: Strategic development of the usability engineering function. interactions, 1999. **6**(5): p. 27-34.
[7] Fogg, B.J., Persuasive technology: using computers to change what we think and do. Ubiquity, 2003. 2002(December): p. 5.
[8] Simons, H.W., J. Morreale, and B.E. Gronbeck, Persuasion in society. 2001: Sage Publications, Inc.
[9] Pressman, R.S., A Practitioner's Approach. Tata McGraw Hill Publication, 2005.
[10] Greenbaum, J.M. and M. Kyng, Design at work: Cooperative design of computer systems. 1991: CRC.
[11] Göransson, B., J. Gulliksen, and I. Boivie, The usability design process–integrating user-centered systems design in the software development process. Software Process: Improvement and Practice, 2003. **8**(2): p. 111-131.
[12] Boivie, I., et al., Why usability gets lost or usability in in-house software development. Interacting with Computers, 2003. **15**(4): p. 623-639.
[13] Gulliksen, J., et al., The role of patterns. 2nd Workshop on Software and Usability Cross-Pollination, INTERACT 2003' Conference, 2003.
[14] Lewallen, R., Software development life cycle models. Net Development, 2006. **20**(9): p. 61-72.
[15] Sommerville, I. and L. Prechelt, Component-based Software Engineering. Software Engineering, Addison-Wesley, 2006: p. 412-430.
[16] Easterbrook, S., Software Lifecycles. University of Toronto Department of Computer Science, 2001.

Authors Profile

Muhammad Shakeel Faridi is a student of MS (Computer Science) in University of Agriculture, Faisalabad, Pakistan. He has been published five research papers in different journals. He is the **Member of IACSIT**- Member NO. : 80342498 (International Association of Computer Science and Information Technology) Singapore, **Member of Editorial Review Board,** Journal of Information Technology Education, 131 Brookhill Court, Santa Rosa, California 95409 USA., **Honorary Peer Reviewer of GJCST** - Global Journal of Computer Science and Technology in Global Journals I Inc. U.S.A and **Member of the editorial board**, International Greener Journals (IGJ), Ikeja, Lagos Nigeria. His area of interest is software engineering, human computer interaction, databases and data warehousing and data mining.

Tasleem Mustafa received his MS(CS) from University of Agriculture, Faisalabad, Pakistan, MSC(CS) from University of Engineering and Technology, Lahore, Pakistan, MSC(Mathematics) from University of Punjab, Lahore, Pakistan and MIS from USA. He is working as a Chairman/Assistant Professor, Department of Computer Science, Faculty of Sciences, University of Agriculture, Faisalabad, Pakistan. His research of interest includes software engineering and database technologies.

Fahad Jan is a Lecturer in Department of Computer Science, Faculty of Sciences, University of Agriculture, Faisalabad, Pakistan. He received his MS(CS) from University of Agriculture, Faisalabad, Pakistan. His area of research is in the design and implementation of advanced programming languages, including functional languages, object-oriented languages and concurrent languages. Database and information systems; web data management; distributed databases; information integration; object-oriented database technology.