A Framework for Integrating Software Usability into Software Development Process

Hayat Dino
AFRICOM Technologies, Addis Ababa, Ethiopia
hayudb@gmail.com

Rahel Bekele
School of Information Science, Addis Ababa University, Ethiopia
rahtesf@yahoo.com

Abstract

This paper reviews the status of software development companies to what degree they adapt formal software process model and perceive & integrate usability throughout the development process. The researchers believe that usability must be tightly integrated in the software development process and makes its objective to examine how usability can be integrated more tightly into a specific software development process model that is being applied by the majority of software development companies. Process models proposed by software engineering and some of user-centered design models proposed by usability engineering are discussed. Moreover, literature study, related works, survey and its findings are discussed. Finally, a framework is developed for use by software development teams wishing to integrate user-centered design practices with a particular software development process model by adding some additional features.

Keywords: Software Development; Software Process Model; Usability; Software Engineering; Usability Engineering; User-Centered Design

1. Introduction

As many research works show, usability should be considered throughout the development process, from the requirement gathering phase to the deployment of the product. It might also be considered further. This is because product of good usability can only be achieved if its development process integrates usability. Integration of usability should be considered by the all members who are involved in the software development process. From software development company perspectives, developing software which is usable can increase sales and productivity. From customers' point of view, it makes the end-user satisfied, also increases the productivity. Bear in mind that companies developing software have to make sure that their development process meets usability requirements. Customers should not only demand software which provides the main functionality but also they have to take into account usability as a critical feature.

Along with the concept of usability, there is the Usability Engineering (UE) discipline that is concerned with the question of how to design software that is easy to use (usable). UE is an approach to the development of software and systems which involves user participation from the outset and guarantees the efficacy of the product through the use of usability specification and metrics [1]. As an integral part of software engineering, UE seeks to understand the client's, customer's, and users' wants and needs; translating these needs into system requirements; and designing a system to satisfy these requirements and testing to help ensure their realization in the final product. However, at each level of software development process, the two disciplines have many differences such as levels of iteration and evaluation, terminology, requirements representation, activities, techniques, timelines, scope, roles, procedures, and focus as reported in [2]. Many researchers have designed UE process models which help to develop software which is usable and user-centered. Some of the models will be discussed in the next section.

All software applications developed based on SE and UE lifecycles are tools that help users accomplish certain tasks. However, before developing a software product or designing UI, one should consider information about the people who will use the tool such as who are the system users? What will they need to accomplish? What will they need from the system to accomplish this? How should the system supply what they need? [3].

2. Statement of the Problem

Research results [2, 4, 5, 6] in the past few years recommend for software organizations, developers, user interface designers, and other professionals involved in developing software to integrate software usability on software development lifecycle.

Rubin and Chisnel [7] also indicated that usability is a quality that many products possess, but many, many more lack. The lack of usability may lead the end-user to need further assistance and having bad perception about the application or the software system, in essence that the end-user becomes less satisfied to use the application. As an example, let us take mobile devices, in which people sometimes lack knowledge about how to use those devices. For instance, one might not know where to set an alarm clock, how to browse the Internet using the device, even how to switch off can be difficult. Hence, these kinds of usability issues should be considered in another case as well.

If the end-users find the software too difficult to understand, then an otherwise excellent product could be ruined to failure. The system may not be usable if the software system works as intended by the developer but it cannot support the task, or the user cannot figure out how to do it, or doesn't like the system, or the user interface is inconvenient or hard to understand and is not easy to use [8].

The main reason for not testing usability is because it is not considered as a critical feature of an application. Not only software developers in organizations but also students in schools do not include usability as one of the quality attributes of applications to be developed. In most of software organizations, usability is not integrated or considered unless it is a critical feature from the organization's perspective, although many researches indicated that usability should be integrated in the software development process.

The situation is the same for the case of Ethiopia. Based on the information and experience the researchers have, developers concentrate on the main functional requirements of the applications and do not care whether the system is interactive with the user or not. The development team spends days and weeks figuring out features and specifications without one mentioning the user; to mean that developers are not focusing on information about users.

On the other hand, not adapting usability engineering lifecycle in the software engineering lifecycle is one problem though some software development companies are not practicing specific software process model. This is because usability is wrongly identified to be a Graphical User Interface issue which can be addressed after the main part of the functionality has already been developed and lack of knowledge about the overall principles of usability engineering discipline.

The other reason for lack of developing a usable product is the issue of usability testing. Based on the information the researchers found, during the testing phase, some end-users participate in order to test the software system whether or not it fulfills the functional requirements; they are not there to verify the software ease of use, learnability, the time to complete a specific task and other usability criteria. This is because there is lack of knowledge as mentioned earlier. Hence, developers could not figure out errors and problems, which happened because of lack of usability since they do not test the software by considering the major factors of usability. In this way, it becomes very difficult to find software that fails because of usability issues in the Ethiopian context.

Since Ethiopia has many nations and nationalities with different cultures and languages, the researchers intended to work on in this emerging research area by addressing some research questions. One of the research questions is adapted from another study that was conducted in Norway [9]. This would help to practice development of usable applications whether web-based or desktop by taking into account some critical issues that make a difference on using a system.

Based on the above statements of the problem, the main objective of this research is developing a framework that integrates software usability into the software development process model. Under this, the following specific objectives have been tried to be achieved:

- to find out which process models have been used,
- to find out which process models are being applied currently,
- to examine the degree to which usability is integrated into software process model,
- to study the problems of those process models from the view point of usability,
- to analyze the level developers and project managers understanding about usability engineering,
- to investigate the major challenges of integrating usability into software engineering lifecycle, and
- to develop a framework that guides the way of integrating usability into a particular software development process.

3. The Proposed Solution

In order for developing a framework, a survey research was conducted to analyse how software development companies perceive software process model as well as usability throughout the development process. Furthermore, a particular process model that is being applied by the majority of the software development companies is selected to come up with the framework. Thus, waterfall process model is found to be the mostly applied approach. Some of the additional components added to the traditional waterfall model in the proposed framework (see Figure 1) will be discussed as follows.

3.1 Requirement Phase

User Profile: There are some types of information that are important to put into user profile. These include (i) characteristics of the real users who are going to use the system (e.g., age, culture, language, social issues, disability issues, etc.); (ii) what they want to do with the system; (iii) how they are going to perform specific tasks using the interface; (iv) users' experience for a certain task; (v) users' motivation, and so on.

Task and context Analysis: Analysis of tasks helps for understanding the user's tasks at both a high-level and detailed level. According to [10], during task analysis, the problem to be solved and user's tasks should be documented. Otherwise it would not be enough to design a solution for undocumented problems. In the case of context analysis, three critical aspects should be explored and documented. These are usability expectation of users, their needs, and feasibility (i.e., specifications including costs and resources required to design the interface and constraints and assumptions regarding the user interface to be designed).

Usability Goals: The product should have an agreed set of qualitative and quantitative usability goals that can be used to measure how useful the product is.

After identifying the above mentioned aspects, evaluation will be made whether or not they are identified based on users' needs.

3.2 Design Phase

Interaction and Interface Design: the interaction design is the look and feel of the user interface as well as the design of the individual interaction. The interface design is about the design of the actual software that drives the interaction. These components help the developer to come up with a good user interface that can be accessible by any user.

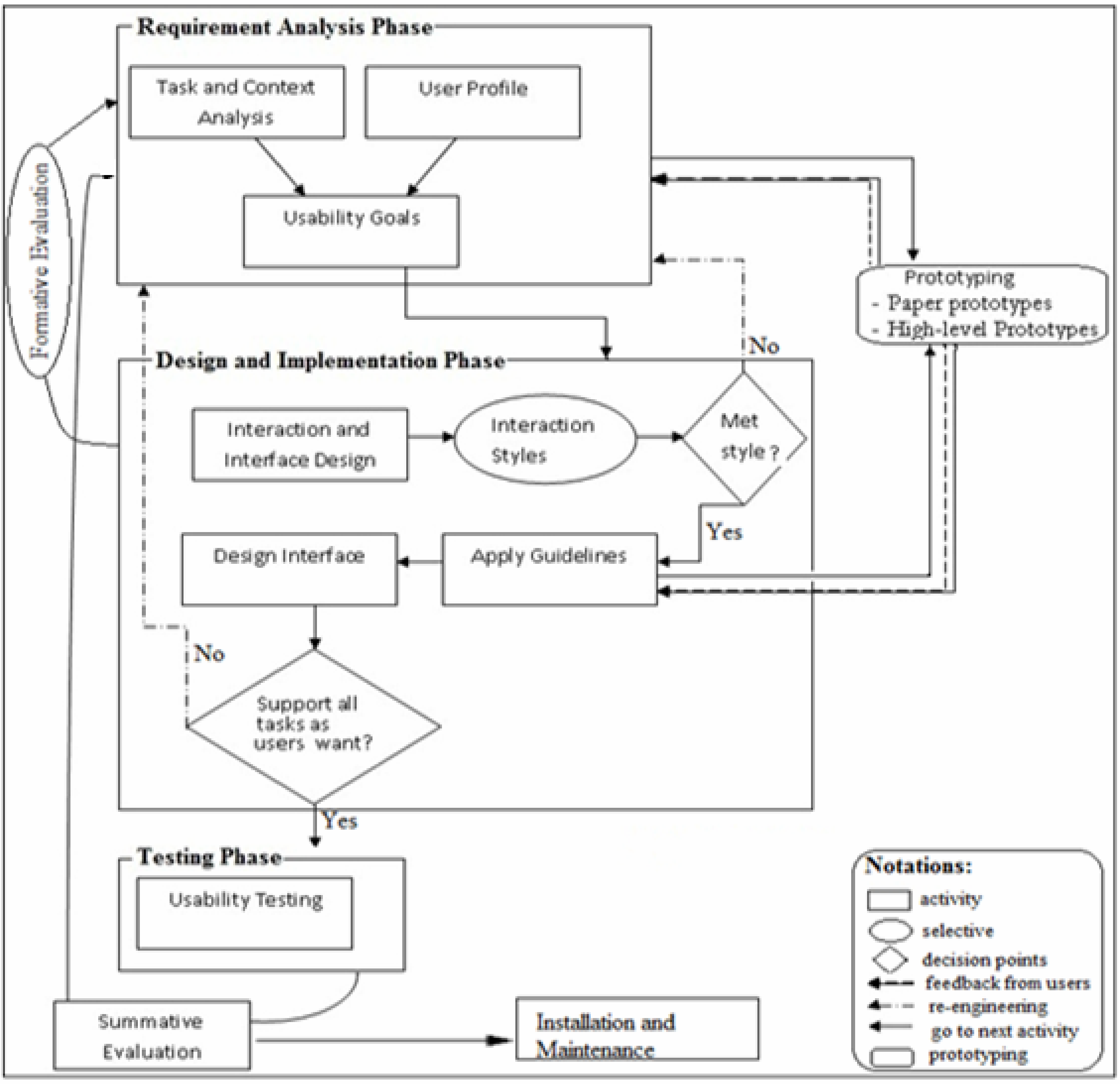

Figure 1: The Proposed Framework that integrates usability into the waterfall process model

Interaction Styles: help to identify how a user interacts with the interface from user's perspectives. Interaction styles are chosen depending on the task and user characteristics. If the interaction styles are not selected depending on these, at the end of the day, the user may not be comfortable with the design.

Guidelines: these help to improve the design of the interaction so that one should apply guidelines for developing individual interactions within a given interaction style.

3.3 Testing Phase

According to the proposed framework, usability testing must be conducted rather taking it as an optional testing type like others since the purpose of the framework is for integrating usability issues into the waterfall model and developing usable products. Hence, usability testing is the most critical part to be considered. Software development companies or individual software developers can adapt appropriate testing techniques based on the project and their characteristics.

Mainly there are three types of usability evaluation methods such as Testing, Inspection, and Inquiry. In the case of usability testing approach, representative users work on typical tasks using the system (or the prototype) and the evaluators use the results to see how the user interface supports the users to do their tasks. Testing methods include the following techniques: Coaching Method, Co-discovery Learning, Performance Measurement, Question-asking Protocol, Remote Testing, and Thinking-Aloud Protocol. The Inspection methods include Cognitive Walkthroughs, Feature Inspection, and Heuristic Evaluation. The scope of the research does not allow to briefly describe the mentioned usability evaluation methods. For the case of formative and summative evaluation, testing techniques which have been previously mentioned can be applied.

Prototyping: this component helps to prepare a prototype in each phase of the proposed model. A prototype can enable not only the development team to better understand the requirements of the desired system –both the functional and usability - but also to clarify users' needs iteratively and early in the development process. Also prototyping at the early stage of the development process helps users to express their needs to be considered, problems they want to be solved, and their expectations from the system. It also helps to decide whether to move forward in the development process or to iterate back to the previous state. Furthermore, it is stated in [10] that the use of prototypes can potentially improve usability and user satisfaction and may reduce the resources needed for development effort. It is also identified that for supporting iterative development, software developers must be able to generate prototypes more quickly than a finished project [10].

It is mentioned in [10] that prototypes may be distinguished by the amount of detail that they include and the goal of the prototype. Thus, different scholars divide prototypes into different ways. For instance, some of them divide it into three types namely: scenario/storyboard prototypes, demonstration prototypes, and version 0 prototypes. The others divide prototypes into two common types: (i) Low-fidelity prototype – useful in the early stage of the development process and it can be presented as paper-based prototype. It helps to evaluate limited functionality and user interactions of the system, and (ii) High-fidelity prototype – this is complete and or nearly complete functionality and is interactive. It also helps the designers to gather realistic feedback from users.

The above mentioned components found in each phase of the framework are to be considered iteratively until the development team gets the right design as well as the users' needs are fulfilled as they want it to be. Furthermore, the framework forces the development team to involve users from the customer side. This will make the final product easy to use by any users. Besides, users will have high interaction with the development team throughout the development process.

4. Related Work

Many researches have been made regarding integration of usability. Most of the researchers believe that integrating usability in the software development process lifecycle is very crucial for user satisfaction, effectiveness, efficiency, and accessibility of a software system or an application.

Pyla *et al.* [2] presented a model based framework for integrating usability engineering in the software engineering lifecycle. They discussed that interactive design requires specialized UE knowledge, training, and experiences in topics such as human psychology, task analysis, visual perception, cognitive load, and so forth. They consider UE and SE as two different lifecycles and each requires their own experienced professionals. Even if the application's user interaction is tested by usability engineers and came up with some usability problems, the software development organization to which the system is tested for will not be willing to improve the problems because of budget and time constraints. They also reported that lack of coordinating the two development lifecycles leads to conflicts, gaps, misinterpreting the users' needs, lack of provision for change, lack of communicating among different development roles, and the like.

In order to overcome the mentioned problems they proposed process model as a solution that helps to integrate the two development lifecycles and also enables the software engineers and usability engineers to work together without affecting or conflicting to one another.

Nebe and Zimmermann [6] proposed an approach that identifies integration points between software engineering and usability engineering on three different levels of abstractions. Before identifying the integration points they analyzed the two usability engineering standards DIN EN ISO 13407 and the ISO/PAS 18152 in order to acquire detailed knowledge about usability engineering activities, methods, deliverables and their quality aspects. They identified that ISO/PAS 18152 defines detailed base practices that specify the tasks for creating usable products. These base practices have been used as a foundation to derive requirements that represent the ‘common activities’ from the usability engineering perspective. The quantity of fulfilled requirements for each activity of the framework informs about the level of compliance of the software engineering model satisfying the base practices and therewith the usability perspective of activities. DIN EN ISO 13407 introduces a process framework for the human-centered design of interactive systems. Its aim is to support the definition and management of human-centered design activities.

Seffah *et al.* [11] reported statistical evidence that shows the importance of user involvement throughout the software development process, the cost-benefits of integrating usability and how a product can be less usable because of lack of user involvement.

Juristo *et al.* [12] mentioned that software usability should be integrated into the software development process before the design phase is started. They implicitly claimed that usability of software should be considered as one quality attribute.

Ferré *et al.* [3] proposed research regarding how to integrate usability in the software development process lifecycle by reporting some usability techniques that can be most valuable for integration purpose. They suggested from those usability techniques a particular set of techniques that can be selected based on a specific characteristic of the software development process model followed by organizations and the application being developed.

5. Conclusion and Recommendation

This paper focuses on finding ways for software developers and customers to integrate usability into the software development process in order to develop usable software. Based on theories on software development methods, user-centered design and related works, the paper has proposed a framework for integrating usability in the waterfall software development lifecycle. The waterfall method is selected because the majority of customers signed the contract under the activities of waterfall development lifecycle. Furthermore, software development team in a particular company apply waterfall combining with other methods.

The literature study documents that waterfall software development process model is not likely to take usability aspects and the user interface itself into account at the required level. It focuses on the functional aspects of software development and preparing documents about the overall structure of the project starting from requirement analysis. The waterfall method does not guarantee user participation throughout the development process as well as iterative process. So that, the framework is designed in order for making the waterfall method a short time span from the requirement phase to actually working software and enable it to respond to changes dynamically when identifying challenges and defects as the project evolves.

Comments from customers or end-users is very important and fundamental because it addresses functional challenges and errors. Moreover, it should be taken as a routine activity of the development process. This routine can give a way of how to apply valuable aspects of usability and how usability can contribute in the process for delivering it as feedback about challenges and errors in the user interaction activities. Thus, this paper has tailored the focus of enabling the software developers in developing products with integration of usability aspects in mind. That is, the software developers themselves should be able to bring a concerted action involving an interdisciplinary approach to software development including optimizing usability. Also in order to make effective use of usability, the paper examined user-centered design methods where user participation is considered absolutely central.

Numerous usability scholars consider some critical aspects of interaction design to be considered in order to come up with a usable and interactive product. So the issues to be considered in order for developing usable products are: who are the users, how do they use the software, what they want to achieve with the system, and how can one evaluate the software. By using these aspects of user-centered design, the proposed framework is more likely to provide an important knowledge about the users that proves to be crucial to creating a software product that is in fact usable and not only just functional. After all, the software developers themselves also agreed that the end-users should be involved in the development process starting from the initial phase till the end even though they do not involve them practically.

The researcher was intending to apply the proposed framework into real point of sale application system for validating the framework. Because of some unexpected problems that happened in the company (the one who developed the POS system and delivered for its customers), the validation process could not be completed. So that the researcher intends to precede the validation process as soon as the problems are overcome.

References

[1] K. Nebe, D. Zimmermann, and V. Paelke, “Integrating Software Engineering and Usability Engineering,” *Engineering*, Vol. 2, 2002.

[2] P. S. Pyla, M. A. Pérez-quiñones, J. D. Arthur, H. R. Hartson, V. Tech, and M. Hall, “Towards a Model-Based Framework for Integrating Usability and Software Engineering Life Cycles”.

[3] X. Ferré, N. Juristo, U. P. D. Madrid, H. Windl, S. Ag, and L. Constantine, “usability engineering Usability Basics for Software Engineering,” *IEEE Software*, February, 2001.

[4] H. Wagner, “Integrating Usability into the Software Development,” University of Hamburg, 2004.

[5] W. J. Giraldo and T. Granollers, “Integrating Usability Techniques into Rational Unified Process,” *Human-Computer Interaction*.

[6] K. Nebe and D. Zimmermann, “Aspects of Integrating User Centered Design into Software Engineering Processes,” *Engineering*.

[7] J. Rubin and D. Chisnell, Handbook of Usability Testing_ How to Plan, Design anad Conduct Effective Tests, Wiley Publishing Inc., 2008.

[8] S. Lauesen, “User Interface Design_A Software Engineering Perspectives”, Addison-Wesley, 2007.

[9] “Software Development Methods and Usability, Perspectives from a Survey in the Software Industry in Norway,” *Methods*, No. 2003, pp. 1-18, 2004.

[10] L. Leventhal and J. Barnes, “Usability Engineering_Process, Products, and Examples”.

[11] A. Seffah, A. Benyahmed, and M. Khalil, “Integrating Usability and Human Factors in Software Engineering Lifecycle and Organizations Software Usability Economics,” 2003.

[12] N. Juristo, A. M. Moreno, Maria-Isabel, and S. Segura, “Analysing the Impact of Usability on Software Design”, The Journal of Systems and Software, pp. 1506–1516, 2007.