

An Overview of Software Models with Regard to the Users Involvement

Saad Masood Butt¹, Wan Fatimah Wan Ahmad²
Computer and Information Sciences Department
Universiti Teknologi PETRONAS
Tronoh, Perak, Malaysia

Abstract

The rapid growth in the field of software development process is taking place by integrating it with other fields of computer science. Human Centered Software Engineering (HCSE) is one of them that bridge the gap between Human Computer Interaction (HCI) and Software Engineering (SE). An important aspect in HCI is Usability Evaluation (UEV) that improves software quality. In this paper, involvement of HCI discipline in software development processes is discussed and on the basis of issues and gap a new approach for HCSE is introduced. Keeping HCSE as stepping stone a framework for intelligent software development life cycle is proposed that provides a complete and intelligent development infrastructure in which HCSE and UE coexist in complementary roles.

Keywords: *human computer interaction, software engineering, human centered software engineering, usability evaluation*

I. INTRODUCTION

Design of software is an important task of Human Computer Interaction. Human Centered Software Engineering is the latest approach in HCI that link SE and HCI discipline based on commonalities. For the last two decades both HCI and SE discipline are working in their own domain then later it became obvious that their integration would benefit the development of interactive software applications [1].

Human Computer Interaction (HCI) is a multidisciplinary subject where HCI experts play an important role to design a usable system, so HCI experts must be knowledgeable in understanding of human nature, requirement modeling and user interface design [2]. User Interface Design (UID) is one of the important aspects of HCI, which focuses on ease of use, ease of learning, user performance and user satisfaction. On the other hand Software Engineering (SE) focuses on requirement especially functional requirement that must be translated into the running system [3].

In order to overcome the gap HCI [3, 4] and Agile SE [6], development of an integrated model is an optimum solution that can integrate HCI and SE. CRUISER Cross-Discipline User Interface and Software Engineering Lifecycle is a new approach that integrates HCI and agile approaches of SE [2].

Lack of user involvement in software development not only affects product quality but also resulted in user dissatisfaction. From the study, it seems users are not satisfied due to the problem in the software interface and unhelpful error messages that affect product quality. This is because users were not properly involved in the entire software development process. User involvement takes place at the early stage when collecting the requirement and at the later stage of the development in order to validate and verify their requirements.

Software development shows user involvement only in the phase of collecting functional requirements. Only few users' involvement seems at collecting requirement or at the development phase of software. The question arises "why users' involvement is still negligible during the design and development stage" answer is due to the users' lack of technical knowledge in software development [7]. This Paper will first summarize literature review. Then provide an overview of related work along with the limitation of previous and popular various software models. In the end, the paper will present a complete overview of the proposed framework, its validation and verification aspect and future work.

II. LITERATURE REVIEW

The problem with current scenario that occurs both in SE and HCI, and why research is needed is because HCI focuses on UID issues where as SE is conscious about the requirement to translate them in the running system. As both SE and HCI plays a very important role in producing quality software.

In order to ensure that the requirement that was mentioned in requirement engineering (RE) and the product fulfills it, HCI and SE need to work together in the interaction layer of Software development. It is not clear how HCI and SE experts work together when there is a need to provide a high level of UI Usability.

As described in [8] about the use of user stories in Extreme Programming [6]. Scenarios in SE they take it as a sequence of events triggered by the user and are used for requirement gathering. HCI uses Scenarios in different contexts like users, users' role, task and interaction [8]. Prototypes in SE are used

to verify the functional requirements. Agile software process recognizes prototypes as a small release [6, 9]. HCI on the other hand apply Prototypes for iterative UID [10]. To overcome these small gaps Cross-Discipline User Interface and Software Engineering Lifecycle [2] is the best possible and practical compromise.

The proposed model mentioned in [11] providing a methodology that helps software developers to modify their different product versions in order to meet the maximum user interaction. The success of the propose model will only be possible once it fulfills user needs in a true manner. More ideas can be implemented in order to achieve maximum user satisfaction.

Memmel et al. describe a software lifecycle CRUISER [2] which is one of the steps taken to bridge HCI and SE based on common features of both fields. CRUISER is very close to XP [6] but don't have agile aspects. The author tried to integrate the important discipline in one lifecycle. Increase the involvement of user and stakeholder by using prototype and scenarios [3].

As mentioned in [12] almost thirty-seven to fifty percent of effort in software development are related to the system's user interface. Because of this reason human-computer interaction (HCI) affects the overall process of software engineering (SE). One of the major issues discussed is the role of HCI in traditional software development. If the HCI is ignored then there will a great chance the problem will occur either in testing or in the maintenance stage. If the HCI is treated in later stage it will be very expensive.

The paper [13] describes a design process that helps to link both SE and HCI processes. The scenarios presented in this paper serve as to link between the two disciplines. In the end a tool was discussed name Scenic Vista that works as a prototype to link design artifacts of SE and HCI.

The methodology mentioned in [14] discussing about the integration of modern systems development life cycle (SDLC) with human computer interaction (HCI) in information systems (IS). As in the traditional development lifecycles of IS, role of HCI is too low only at design phase or at a later stage that affect the overall of development. Thus there is a gap found between HCI and SE and in order to remove this gap human-centered IS development approach is introduced.

On the whole various methodologies trying to link HCI and SE, some design process helps to integrate usability in software development process. In order to manage all issues and gaps, an efficient development model must be proposed. So development of intelligent HCSE process takes place to remove all the gaps and issues between HCI and SE.

III. RELATED WORK

In the section an overview of previous work mentioned in [2] is highlighted. In 2007, Memmel et al. from Human-Computer Interaction Lab Germany proposed a software development life cycle in his paper "Agile Human Centered Software Engineering" later published by British Computer Society. The authors of this paper confess that they did not implement this development lifecycle practically. But the proposed lifecycle is based on the facts of industry and research experience. They believe by the use of this development lifecycle in practical projects not only helps to build better software but can provide software with high functional in less time with significantly lower costs.

Further details of that proposed software development life cycle were published in [3]. Motivation of the paper was to bridge HCI and SE based on the common attributes in both HCI and SE.

It is important to provide some highlight of CRUISER lifecycle in relation to the proposed framework. CRUISER contains five phases as shown in Figure 1.

As mentioned in [2] the first phase of CRUISER is an initial requirements up-front IRUP. In this phase real users are considered rather than stakeholders. The needs of users can analyze by Role model and Task Models. A Model based technique which is proposed by [15] is to ensure the true meaning of agile by the help of index cards. The user role is first prioritized and use cases are developed. Similarly task models are also prioritized and analyzed which task is important first to implement. During IRUP phase, users, HCI experts, SE experts and business support to get final requirements. This will help to final IRUP phase that consist of outline of various mock-ups or prototypes. IRUP phase helps to get the real needs of users that can be described with the help of user needs and task goals.

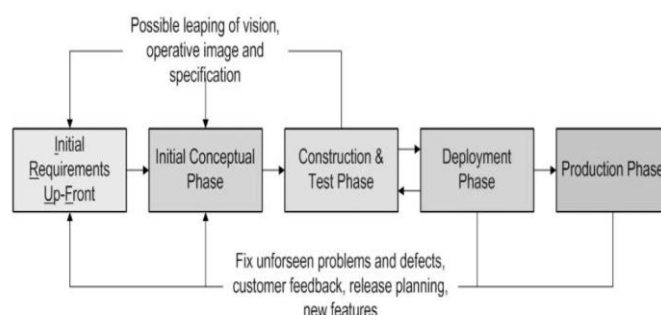


Figure 1: Phases of CRUISER [1]

The second phase of CRUISER is initial Conceptual Phase (ICP). In order to speed up the development process User Interface (UI) and the architecture of the system build in parallel. Task cases and scenarios help to identify the dependencies between UI and architecture of the system. The ICP helps to narrow down toward the solution through scenario refinement.

The third phase of CRUISER is Construction and Test Phase (CTP) in which coding will start in parallel as soon as the prototype is finalized by ICP. The CTP will work in the form of incremental and iterative manner very similar to extreme programming. If any component of UI having great impact needs to be developed first and later refine it during the iteration process. CTP end with small release. Each small release must be evaluated [17] before going to the next iteration. If any issue or defect is identified they must be noted on defect card and corresponding task case also reviewed. CTP prioritizes all defects and review them in the earlier or later iteration.

The last phase of CRUISER lifecycle is Deployment. It is very common in software that new requirement are requested or any design issues occurring while using it. The CRUISER lifecycle will return to an earlier phase to accommodate all new requests and design issues.

IV. PROBLEMS IN VARIOUS SOFTWARE MODELS

The classic software engineering process like V-Model, coding is start very early stage as soon as little requirement gathered from users. Implementation of code takes place in small increments and iteration. And the customer is supplied with a small release after each development cycle. During its requirement analysis phase development team writes user stories to describe user needs and roles. The people interviewed need not necessarily to be the real user of the software product, hence the software fails due to lack of coordination with real users and fails to collect real user data. So coding phase starts on the bases of assumed users' needs [17].

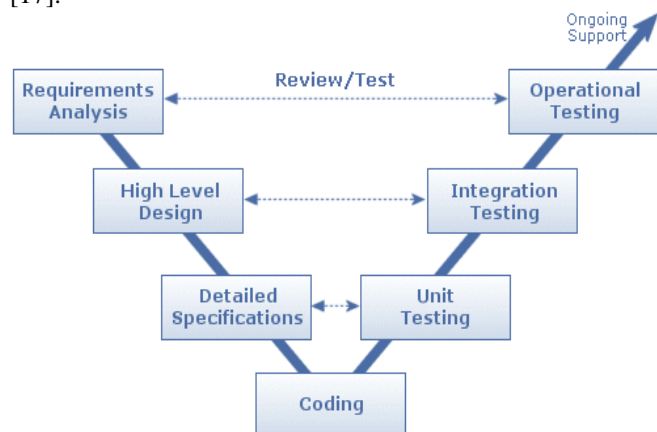


Figure 2: V-Model [17]

Describing the traditional software development like waterfall model it contains a series of phases that must be completed in sequence. All requirements must be completed properly before going to the next phase. If any new requirement comes or change in existing requirement makes it too expensive and difficult to handle. The Problem occurs in one phase may propagate to the next phase and so on. The output of one stage serves as input for the succeeding stage. The user involvement

in Waterfall model is only at the requirement gathering and design phase. The waterfall model is not good to handle rapid change in requirement and is not good for large projects. Only users or stakeholder having a clear vision about their project need this model [18].

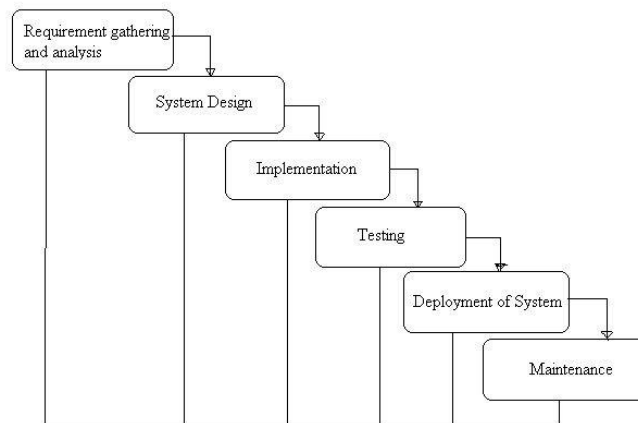


Figure 3: Waterfall Model [18]

Spiral Model is one of the Software development models and based on software development life cycle model. It integrates the characteristics of waterfall model and prototyping mode. Spiral model is good for large projects only because it's a very costly model for software development. High expertise is required to handle risk and uncertainties in the project. Such project successes are dependent on handling such risk and uncertainties [19].

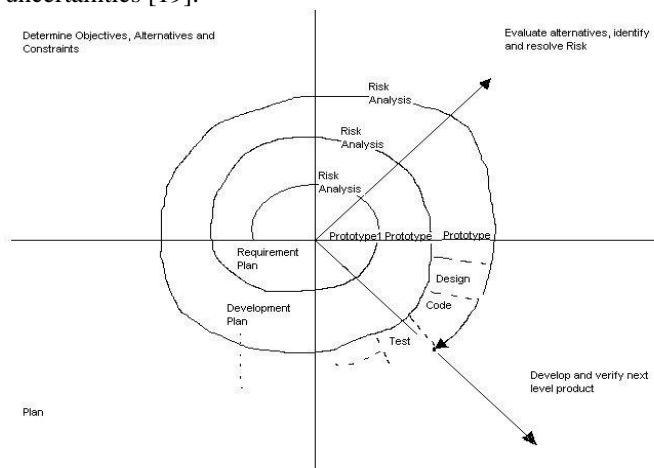


Figure 4: Spiral Model [19]

Agile development process is about iterative and incremental development, it focuses on initial requirements-gathering. So the missing requirement will cover in the next phase of an iteration. But the agile development process work properly as long it do not focus on user interface (UI). But if the agile process starts focusing on user interface it will fail to quality user centered design (UCD) [2].

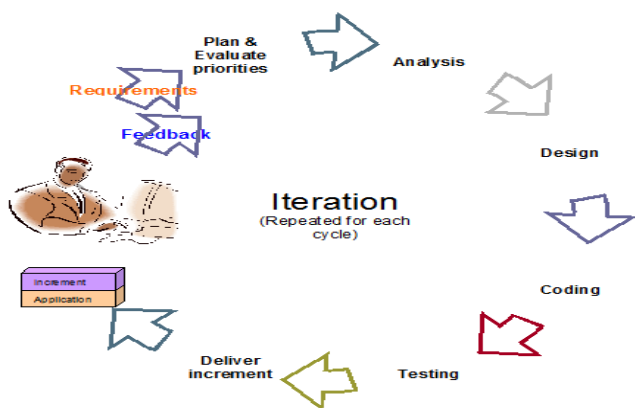


Figure 5: Agile Model [20]

Software Development Life Cycle (SDLC) shows the user frustration and its effect on the software products. This frustration is due to ignorance of HCI considers in software development methodologies. Zhang et al. [22] propose new SDLC emphasizing on a human centered approach called Human Centered System Development Life Cycle (HCSDLC).

According to the Microsoft Research an article was released in March 2008 named as “Human Computer Interaction in the year of 2020” [23] discussing the role and importance of HCI in Human Life and how the world will change by the new paradigm of HCI in 2020.

The problems mentioned in various software models in either traditional or agile method are the main source of product failure. Some major problem needs to be highlighted may include:

- Development process is not flexible
- Lack of User involvement
- Lack of focus on User Interface
- Unable to handle rapid change in Requirements
- Lack of Software Usability

So it seems their need a cross discipline software development that can fulfill agile approach of SE as well as infuse HCI approach. Agile Human Centered Software Engineering [2] proves to be a complete model that consolidates both SE and HCI approaches.

V. PROPOSE FRAMEWORK

The main focus is to automate Initial Conceptual Phase (ICP) of CRUISER life cycle [1] or come up with intelligent approach that can reflect the definition of “Automated Human Centered Software Engineering” as show in figure 2.

Usability needs to be tested once the software is developed by CRUISER model. So the development of a framework on “Mental Examination of Software Interfaces” is important to test the usability of the interface.

The objective is to find an appropriate way to develop an Intelligent HCSE lifecycle (CRUISER) that fulfills agile practices in software development which can reduce the pressure of time and can build software in less time in an intelligent way.

The second objective for the development of a framework is to evaluate a methodology for “Mental Examination of Software Interface” to assure Usability of software developed by CRUISER model. In order to achieve the objective the proposed framework gives a clear picture of all processes need to be done.

As ICP phase falls in CRUISER lifecycle, clear understanding of CRUISER lifecycle is important. In ICP a development of intelligent technique can be addressed to automate ICP phase of the CRUISER life cycle. As soon as automation of ICP phase is done it need to be tested practically. Any defect identified the cycle will go back to intelligent technique and fix it.

The next phase is to evaluate the software interfaces by the help of “Cognitive Examination of Software Interfaces”. This will examine the user mental model regarding software interface. As mentioned in [11] about the proposed model tries to get maximum user interaction.

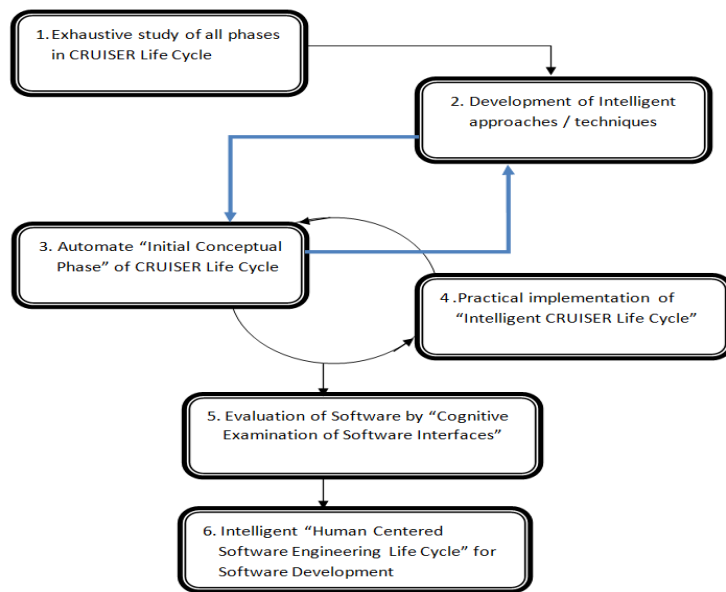


Figure 6: Propose Agile framework

The success of the proposed framework will only be possible once the user needs are satisfied. The last phase of the propose framework is the complete development process that can be called as “Intelligent Human Centered Software Engineering Life Cycle”.

VI. SIGNIFICANCE OF THE PROPOSE FRAMEWORK

The proposed framework has two significant roles. First role it’s a new approach in software development industry which will not only covers the loop holes of the software industry. The second role of proposed framework help to fulfill the needs of the development sector in an innovative way that will increase capacity building of people as mentioned in [21]. The proposed framework helps to automate the software development in less amount of time, able to handle new

requirement as well as rapid change in current requirements, remove the gap between HCI and SE and fulfill Usability goals.

VII. VALIDATION AND VERIFICATION

As a theoretical concept, this framework is being implemented in a closed environment with various parameter settings. The proposed framework work is implemented with the following assumption.

Real world data is dynamic and requirements with event to time change. Due to the organization structural changes completeness and correctness of software artifacts is essentially uncertain from conception.

VIII. FUTURE WORK

Author is a PhD student of Information Technology and the proposed framework is related to his PhD research work. During his PhD tenure, the author will evaluate the framework practically with real world projects. The proposed framework needs to be practically tested in order to evaluate its result with other software development models. But this framework is the integration of agile approaches of software development with usability aspects of HCI and development of the proposed model to meet maximum user interaction needs in a shorter amount of time.

IX. CONCLUSION

The software development process is tremendously changing from traditional to agile and now HCSE approaches. The paper proposed a frame work which is the further continuation of CRUSIER lifecycle [3]. The framework supports to build software development process in an intelligent way and assures the Usability of software.

By this framework, software can be developed in less time, fulfill user requirements and Usability of software can easily be tested in one development lifecycle. Propose model is the integration of agile approaches of software development with Usability aspects of HCI and development of methodology to meet maximum user interaction needs in a shorter amount of time.

ACKNOWLEDGMENT

Author of the paper would like to thanks to Universiti Teknologi PETRONAS and other staff members for their kind support.

REFERENCES

- [1] Seffah, Ahmed; Vanderdonckt, Jean; Desmarais, Michel C. (Eds.), Human-Centered Software Engineering: Software Engineering Architectures, Patterns, and Models for Human Computer Interaction. Springer. ISBN 978-1-84800-906-6, Springer Dordrecht Heidelberg London New York
- [2] Thomas Memmel, Fredrik Gundelsweiler , Harald Reiterer, Agile Human Centered Software Engineering, Published by the British Computer Society, Proceedings of HCI 2007
- [3] Thomas Memmel, Fredrik Gundelsweiler , Harald Reiterer, CRUISER: A Cross-Discipline User Interface and Software Engineering Lifecycle, Human Computer Interaction, Part I, HCII 2007, LNCS 4550, pp.174-183, 2007, Springer, Heidelberg (2007)

- [4] Constantine, L.L.: Process agility and software usability: Toward lightweight usage centered design, Information Age, vol. 8(8) (August 2002)
- [5] Gundelsweiler, F., Memmel, T., Reiterer, H.: Agile Usability Engineering. In: Keil- Slawik, R., Selke, H., Szwillus, G. (Hrsg.) Mensch & Computer 2004: Allgegenwärtige Interaktion, pp. 33–42. OldenbourgVerlag, München (2004)
- [6] Beck, K: Extreme Programming Explained. Addison-Wesley, London, UK
- [7] Users' Frustration and HCI in the Software Development Life Cycle, International Journal of Information Processing and Management. Volume 2, Number 1, January 2011
- [8] Rosson, M.B, Carroll, J.M: Usability engineering: scenario-based development of human computer interaction. Morgan Kaufmann, San Francisco, Chapter 53 in J. Jacko & A. Sears (Eds.), The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Associates, 2002, pp. 1032-105
- [9] Blomkvist, S.: Towards a model for bridging agile development and user-centered design. In: Seffah, A., Gulliksen, J., Desmarais, M.C. (eds.) Human-centered software engineering– integrating usability in the development process, pp. 219–244. Springer, Heidelberg
- [10] Rudd, J., Stern, K., Isensee, S.: Low vs. high fidelity prototyping debate, Interactions, vol. 3(1), pp. 76–85. ACM Press, New York
- [11] Uzma Jami , Tasleem Mustafa, Cognitive Analysis of Software Interface, ISSN 1450-216X Vol.41 No.1 (2010), pp.99-108
- [12] SHAWREN, An Overview of Systems Design and Development Methodologies with Regard to the involvement of Users and other Stakeholders, SAICSIT 2003
- [13] J.brown, Sharing Human-Computer Interaction and Software Engineering Design Artifacts, Computer Human Interaction Conference, 1998. Proceedings. 1998 Australasian
- [14] Integrating Human-Computer Interaction Development into SDLC: A Methodology, Proceedings of the Americas Conference on Information Systems, New York, August 2004
- [15] Constantine, L. L., and Lockwood, L. A. D., Software for Use: A Practical Guide to Models and Methods of Usage- Centered Design. Addison-Wesley, Reading, MA, 1999
- [16] Gellner, M., Forbrig, P.: Extreme Evaluations – Lightweight Evaluations for Soft-wareDevelopers, In: IFIP Working Group 2.7/13.4, editor, INTERACT 2003 Workshop on Bridging the Gap Between Software Engineering and Human-Computer Interaction (2003)
- [17] Gundelsweiler, F., Memmel, T., and Reiterer, H. Agile usability engineering. In R. Keil-Slawik, H. Selke, and G.Szwillus (eds.), Mensch & Computer 2004:Allgegenwärtige Interaktion. München, OldenbourgVerlag, 2004, 33-42.
- [18] IBM Study on Software Models, "http://www.theserverside.com/tip/Waterfall-versus-Agile-methods-A-pros-and-cons-analysis"
- [19] Spiral-Model. <http://www.buzzle.com/articles/spiral-model-advantages-and-disadvantages.html>
- [20] <http://www.indiethreads.com/1439/quick-introduction-to-agile-software-development>
- [21] A free innovation guide for social enterprises and NGOs worldwide: <http://www.ideo.com/work/human-centered-design-toolkit>
- [22] Zhang, P., Benbasat, I., Carey, J., Davis, F., Galletta, D., and Strong, D. "Human Computer Interaction Research in the MIS Discipline". Communication of the AIS, vol. 20, no. 9, pp. 334- 355, 2002
- [23] BEING HUMAN: Human Computer Interaction in the year of 2020, "http://research.microsoft.com/en-us/um/cambridge/projects/hci2020"

AUTHORS PROFILE

Saad Masood Butt received his BS (Software Engineering) degree from Bahria University Islamabad, Pakistan in 2008. He completed his MS (Software Engineering) degree in 2010 from Bahria University Islamabad, Islamabad Pakistan. He is the recognized Engineer of Pakistan approved by Higher Education Commission and Pakistan Engineering council (PEC). He has got more than 4 years' experience and was associated with various organizations in Pakistan. Currently, he is pursuing his PhD degree in the

department of Computer and Information Sciences at Universiti Teknologi PETRONAS, Malaysia.

Wan Fatimah obtained her PhD from Universiti Kebangsaan Malaysia. She is currently an Associate Professor at Universiti Teknologi PETRONAS, Malaysia. Her research interests include topics on Multimedia, Human computer interaction, mathematics education, e-learning.