VFAST Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2014 ISSN(e): 2308-8168; ISSN(p): 2411-6335
Volume 2, Number 1, January-December, 2014

pp. 15-20

A METHODOLOGICAL APPROACH: FORMAL SPECIFICATION OF QUALITY ATTRIBUTES MODELING APPROCAHES IN THE WATERFALL PROCESS Model

UMER RASHID
Department of Computer Science, Quaid-i-Azam University
umerrashid@qau.edu.pk

Received October 2013

ABSTRACT. *The fields like Software Engineering (SE) and Human Computer Interaction (HCI) are considered dissimilar.. SE based process model mostly discuss modeling of functional requirement while the HCI based approaches are mostly concerned with the modeling of quality attributes. The quality attributes are mostly discussed during late phases of software development. The non-functional requirements as quality attributes can be integrated in software products by considering quality or non-functional modeling approaches during all of the phases of software engineering process model. The separation of SE and HCI concerns restricts formal specification of quality attributes during all of the phases of SE process model. The software systems or products are generally less user centered because SE process models can't address formal specification of quality attributes in SE process models. In this research a methodology for the formal specification of approaches that model functional requirements and quality attribute during SE process model is proposed. The proposed methodology is based on waterfall SE process model. It can be utilized in design and development of users centered software products. Our proposed methodology also bridges gap between SE and HCI fields.*

Keywords*:* Functional Requirements; HCI; Non-functional Requirements; Quality Attributes; User Centeredness; SE; Usability; Waterfall Process Model; Methodology.

1.**Introduction.** Human Computer Interaction (HCI) is observed as one of the most growing and emerging computer science field or discipline [1]. HCI grasps concepts from Software Engineering (SE), Cognitive Sciences, and Behavioral Sciences [2] [3] [4]. The filed is highly emerging and evolved over the years. The SE and HCI discuss system design. The difference is that HCI is mainly focused towards modeling of non-functional requirements or quality attributes, while the SE is focused towards modeling and specification of functional requirements.

Although both disciplines discuss design phases but their concerns are dissimilar [5] [6]. It is revealed that a gulf exists between these two important computing disciplines. This is the main reason that human need as quality attributes or functional requirements are fully or partially ignored in SE process models. Analysts, designers and stakeholders are mostly unaware about the quality attributes specification mechanisms during the early and late design phases of SE products. Special attention is required for the formal specification of functional requirement along with the quality attributes during all of the design phases of software products.

The integration of HCI based approaches in the SE development process model will provides standard mechanism for analysts and developers to slot-in quality attributes along with the functional requirements in the software during early and late design phases. This may improve user centeredness in software engineered products [7].

In this research a methodological approach that formally specify approaches for the modeling of functional attributes and quality attributes during the all of the design phases of SE process model is proposed. The waterfall software process model [8] [9], is one the oldest and most primitive process model. It is considered as a reference model for the integration of quality attributes during the early and late design phases of the SE. In this study the approaches that models quality attributes during the early and late design phases are identified and integrated in the existing waterfall process model. The integration of quality attribute modeling approaches provides one-to-one correspondence with the existing phases of waterfall process model.

In section 2 of this article, we will propose our methodology. The significance of the proposed the proposed mechanism is discussed in section 3. In section 4 we conclude our discussion and discuss some future work.

2. Proposed Integration Methodology. The requirement elicitation, analysis, Design, implementation and testing are the phases that are linearly connected in waterfall SE process model [8] [9]. Model can be utilized as a reference model for the design and development of software systems belonging from multiple divergent domains [10]. In this research we have proposed an enhancement in the existing waterfall process model. Our proposed enhancement will integrates the quality modeling approaches in the existing phases of waterfall SE process model.

In the HCI discipline approaches are available, which are commonly exploited in requirement gathering, analysis, design, implementation, and testing. These approaches can be organized in linear order such that they provide modeling of quality attributes at the various phases of SE process model. Consideration of these approaches at various development phases provides user centered design and integrates quality attributes in the software products [11]. The proposed adaptation in the existing waterfall SE process model is represented in figure 1 represents.

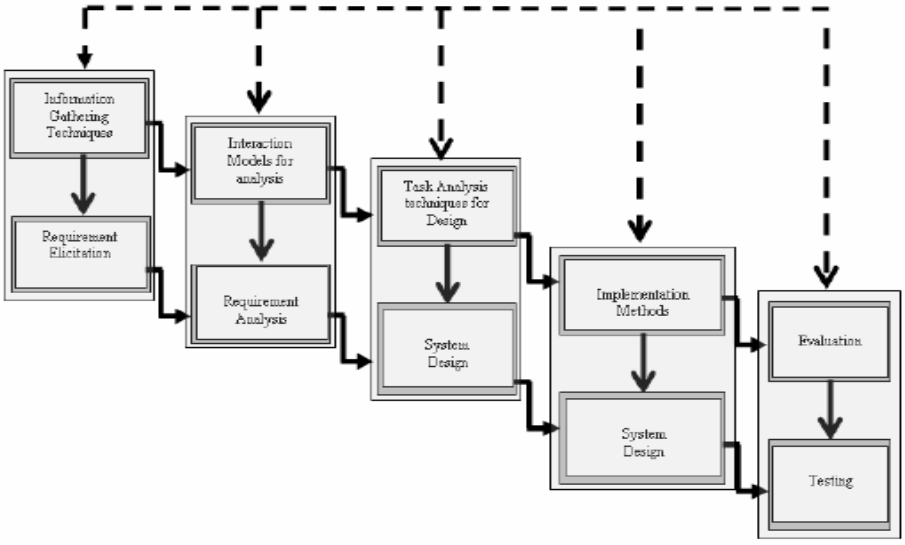

Figure 1. Specification of HCI approaches in Water Fall process model

It is shown in figure 1 that HCI process model runs over the waterfall model and it has one-to-one correspondence with requirement gathering, analyses, design, implementation and testing phases of the waterfall SE process model. The represented model not only provides formal specification of functional requirements, but it also covers the quality attributes specification and modeling during early and late design

phases. This model is an enhancement or adaptation of the waterfall process model. This model includes all of the feature of original waterfall model along with quality modeling aspects of computing in the software development process. In the following sections we will discuss formal specification for quality modeling approaches in the proposed model.

2.1. **Requirement Gathering**. Non-functional requirements are also called quality attributes. They are not given by the users, but lacking of these attributes in software products, negatively effects end users satisfaction [12]. Non-functional requirements can be efficiently gathered by utilizing HCI theories.

In waterfall SE process model functional requirements are identified, specified and communicated with users. HCI provides common practices for the human as well as computers. These practices provides pre-phase of requirement gathering. Surveys, interviews, and questionnaires provide non-functional requirements gathered mechanism. These mechanisms can be designed in such a way that, they will gather end user non-functional requirements [13]. Equation 1 represents the integration of standard HCI based information gathering methods in requirement elicitation phase of selected model.

$$Phase\_1_{output} = f_{RG}(m_1, m_2, ..., m_n) + Nf_{RG}(n_1, n_2, ..., n_n) \quad (1)$$

Where $Phase\_1_{output}$ be the output of the requirement gathering phase. This output is usually in the form of two sets which are: $Set_1 = \{SQA, USQA\}$ and $Set_2 = \{definitions, specifications, communications\}$

Set-1 represents outcomes of nonfunctional requirements gathering activity conducted in the water fall process model. Outcome will be in the form of software quality attributes and unspecified non-functional attributes. Quality attributes may be known in advance because they are in the form of system efficiency, utility and performance etc.; while unspecified quality attributes are unknown in advance and can be gathered using various specified methods. Set2- represents outcomes of the functional requirement gathering phase which will be in the form of definitions, specifications, and communications.

f_{RG} and Nf_{RG} be the functions adopted for the elicitation/Gathering of functional as well as non-functional requirements. f_{RG} Employs set of methods $m_1, m_2, m_3, ..., m_n$ which may includes functional models, use cases, and functional prototypes etc. Nf_R captures non-functional requirements or quality attributes using methods $n_1, n_2, n_3, ..., n_n$ which are usually surveys, questionnaires recordings, and low fidelity quality driven paper based prototypes.

2.2. System Analysis. HCI theory proposes models of interactions. They can be exploited in the analysis of requirements. GOMS and OAI (Shneiderman) Model, Foley & van Dam Model, Abowd and Beale framework, Donald Norman's model, and OAI Model are basic interaction model of HCI [14] [15] [16] [17] [18]. The mentioned interaction models provide requirement analysis for human centered software products or systems. These models can be exploited in analysis of non-functional or quality requirements. Equation 2 shows the specification of interaction models in analysis phase of the waterfall model. These models provides the investigation, analyzing, and integration mechanism for quality attributes in the software products developed using proposed SE process model.

$$Phase\_2_{output} = f_{RA}(m_1, m_2, ..., m_n) + Nf_{RA}(n_1, n_2, ..., n_n) \quad (2)$$

Where $Phase\_2_{output}$ be the output of the requirement analysis phase. This output will be usually in the form of two sets which are: $Set_1 = \{SQAM\}$ and $Set_2 = \{DM, FMl, BM, DD\}$. Set-1 includes software quality attribute models which provide formal modeling of the quality attributes and set-2 includes Data Model, Functional Model, Behavioral Model and Data Dictionary.

f_{RA} and Nf_{RA} be the functions adopted for the analysis of functional as well as non-functional requirements. f_{RA} includes set of methods $m_1, m_2, m_3, ..., m_n$ which may includes data modeling, functional modeling and behavioral modeling approaches. These approaches will be adopted for the analysis of functional requirements while Donald. Norman's model, Abowd and Beale framework, Foley & van Dam Model, GOMS and OAI (Shneiderman) Model are the set of method $n_1, n_2, n_3, ..., n_n$ adopted by Nf_{RA} for the analysis of non-functional requirements.

2.3. System Design. HCI theory discusses Task analysis techniques. These task analysis techniques include Hierarchical Task Analysis (HTA) and Cognitive Task Analysis (CTA). HTA is considered best for designing and analyzing device dependent interactions, while the CTA is considered best for modeling human activity present in users mind [20][21]. In HTA requirements are transformed into hierarchies which are further transformed into implement-able plans, while in CTA model human activities which are omni-present in user's mental model.

Equation 3 formally represents specification of standard HCI based system design mechanisms in the corresponding system design phase of the waterfall process model.

$$Phase\_3_{output} = f_{SD}(m_1, m_2, ..., m_n) + Nf_{SD}(n_1, n_2, ..., n_n) \quad (3)$$

Where $Phase\_3_{output}$ is the output of the system design. This output is- usually in the form of two sets which are: $Set_1 = \{Acction - Seq, Resource - Organization, Non - funtional - Design\}$ and $Set_2 = \{Data - Structures, Algorithms, architecture, interface - design\}$. Set-1 represents outcome of non-functional system design activity conducted using HCI based methods. Outcome will be in the form of action sequence, system resource organization and quality or non-functional design. Set2- represents the outcome of the functional design which may be in the form of data structures, algorithms; software architecture and interface design representations.

f_{SD} and Nf_{SD} be the functions adopted of the design of software functional and non-functional aspects respectively. f_{SD} use the set of methods $m_1, m_2, m_3, ..., m_n$ which may includes entity relationship model, architectural design, and information flow model. Nf_{SD} can be gathered using the set of methods $n_1, n_2, n_3, ..., n_n$ which are usually Hierarchical Task Analysis, Cognitive Task Analysis and Resource Model.

2.4.System Implementation. HCI proposes various prototyping based implementation methods. The low fidelity and high fidelity prototyping implementation can be efficiently employed in the implementation phases [23] [24]. Paper based prototyping, GUI based story boards, screen Sketches, Post-it' notes are are common examples of low fidelity prototypes. The prototyping techniques support brain storming sessions and provide basics implementation start. In later stages of development high fidelity prototyping methods may be engaged in implementation of the software products. High fidelity prototyping methods can be categorized into horizontal and vertical prototyping methods [25] [26]. Equation 4 formally show that how are standard HCI based system design mechanisms may be integrated in the corresponding system implementation phase of the waterfall process model.

$$Phase\_4_{output} = f_{SI}(m_1, m_2, ..., m_n) + Nf_{SI}(n_1, n_2, ..., n_n) \quad (4)$$

Where $Phase\_4_{output}$ is output of the system design. This output is usually in the form of two sets which are: $Set_1 = \{Functional - Code, Documentation\}$ and $Set_2 = \{Non - Functional - Quality - Code, Interface - Design\}$ .Set-1 represents outcome of non-functional system implementation activity conducted using HCI based methods. Outcome will be in the form of functional code and functional documentation. Set2- represents the outcome of the non-functional implementation which may be in the form of data non-functional quality code and interface design. f_{SI} and Nf_{SI} be the functions adopted of the implementation of software functional and non-functional aspects respectively. f_{SD} use the set of methods $m_1, m_2, m_3, ..., m_n$ which may includes pseudo codes, algorithms, and information flow etc. Nf_{SD} can be gathered using the set of methods $n_1, n_2, n_3, ..., n_n$ which are usually low fidelity and high fidelity software based prototypes.

2.5.SystemTesting. The HCI field provides standards software testing/evaluation approaches. These approaches include model based evaluation approaches/techniques [27] [14] [28] [29], which may be user based or expert based. In user based evaluation approaches, the sample of the intended users use the applications. This type of evaluation is useful in exploration of user requirements about interface design in the early stages [29]. User testing is significant only when performed by engaging large samples in the evaluation phase. Expert based system evaluation methods includes Heuristic based evaluation and Cognitive Walkthrough [30] [31]. The Model based evolution methodology exploits predefined standards in the evaluation like ISO [32]. These approaches provide the rigorous estimations regarding usability and can be performed on interface specifications. Equation 5 formally show that how are standard HCI based system

testing mechanisms may be integrated in the corresponding system implementation phase of the waterfall process model.

$$Phase\_5_{output} = f_{ST}(m_1, m_2, ..., m_n) + Nf_{ST}(n_1, n_2, ..., n_n) \quad (5)$$

Where $Phase\_5_{output}$ is output of the system design. This output is usually in the form of two sets which are:

$Set_1 = \{functional-evalauartion\}$ and $Set_2 = \{Quality-Evalaution\}$. Set-1 represents outcome of non-functional system testing activity conducted using HCI based methods. Outcome will be in the form of functional evaluation. Set2- represents the outcome of the non-functional implementation which may be in the form of quality or non-functional evaluation. f_{ST} and Nf_{ST} be the functions adopted of the testing of software functional and non-functional aspects respectively. f_{SD} use the set of methods $m_1, m_2, m_3, ..., m_n$ which may includes unit, regression, white box and black box testing. Nf_{SD} can be gathered using the set of methods $n_1, n_2, n_3, ..., n_n$ which are Quality evaluations usually conducted by heuristic evaluation, cognitive walkthroughs and model based testing.

3. Significance of the Integration Mechanism. Quality attributes and functional requirements in the proposed SE process model has one-to-one correspondence. The one-to-one correspondence makes the adaptation in waterfall SE process model for the analysts and developers to apply the proposed methodology. HCI techniques or approaches captures and models non-functional requirements or quality attributes, while the original waterfall SE process model provides gathering and modeling of functional requirements. The systems that will be designed and developed by using the proposed methodology will functionality and user centered. A Software product always desires a user center design. Massive work has been conducted on the user centered design, but the intergradations of HCI and SE based approaches in the design and development of software products is perhaps less explored area; this area is exploited in this research. The proposed methodological approach for the formal specification of quality attributes modeling approaches in the waterfall process model acts like blueprint and provides bases for modeling quality attributes in the SE process models.

4. Conclusion. The separations of HCI and SE concerns make them distinct domains. The SE process models mostly considers functional requirement, while the HCI is more focused towards quality modeling issues or non-functional requirements. It is needed that that The HCI based quality modeling approaches must be integration in SE process models. HCI based techniques are there; they provides quality modeling during requirement elicitation, analysis, design, and implementation, and testing phases of SE process models. The modeling of quality attributes during development provides user centered design. In this research an HCI and SE based methodology for the formal specification of quality attributes modeling approaches in the Waterfall Process Model is proposed. This approach makes waterfall SE process model a quality driven models that not only considers functional requirements but also cater non-functional requirements or quality attributes. In this research only theoretical prospective of the proposed model are discussed. In future we are intended to conduct case studies, discussing the comparative evaluation of the solutions that will be engineered using the primitive and proposed approach.

.

REFERENCES

[1] Carroll, J. M. (2002). Human-computer interaction in the new millennium. New York: ACM Press

[2] Carroll, J. M. (1997). Human-computer interaction: psychology as a science of design. Annual review of Psychol. 48(1), 61-83.

[3] Wilson, T. D. (2000). Human information behavior. Informing science, 3(2), 49-56.

[4] Phillips, C., & Kemp, E. (1998, January). The integration of HCI and software engineering. In Software Engineering: Education and Practice, International Conference on (pp. 399-399). IEEE Computer Society

[5] Sutcliffe, A. G. (2005). Convergence or competition between software engineering and human computer interaction. In *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle* (pp. 71-84). Springer Netherlands.

[6] Bagert, D. J., Barbacci, M., Budgen, D., Lethbridge, T. C., Suryn, W., & van Vliet, H. (2002, October). Thoughts on Software Engineering Knowledge, and how to Organize it. In *Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on* (pp. 24-35). IEEE

[7] Constantine, L. L., Biddle, R., & Noble, J. (2003, May). Usage-Centered Design and Software Engineering: Models for Integration. In *ICSE Workshop on SE-HCI* (Vol. 3, pp. 106-113).

[8] Sommerville, I., software engineering, Pearson Education, 2004.

[9] Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.

[10] Sun, Z. (2004). A waterfall model for knowledge management and experience management.

[11] Seffah, A., Vanderdonckt, J., & Desmarais, M. C. (2009). Human-Centered Software Engineering: Software Engineering Architectures, Patterns, and Sodels for Human Computer Interaction. In *Human-Centered Software Engineering* (pp. 1-6). Springer London.

[12] Bishop, R., & Lehman, M. M. (1991, October). A view of software quality. In*Designing Quality into Software Based Systems, IEE Colloquium on* (pp. 1-1). IET.

[13] Kuter, U., & Yilmaz, C. (2001). Survey methods: Questionnaires and interviews.*Choosing Human-Computer Interaction (HCI) Appropriate Research Methods*.

[14] Dix, A. (2009). *Human-computer interaction* (pp. 1327-1331). Springer US.

[15] Foley, J. D., & Van Dam, A. (1982). *Fundamentals of interactive computer graphics* (Vol. 2). Reading, MA: Addison-Wesley.

[16] Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction.

[17] Shneiderman, B. (1986). *Designing the user interface-strategies for effective human-computer interaction*. Pearson Education India.

[18] Rashid, U., Niaz, I. A., Amin, M. W., & Bhatti, M. A. (2009, October). Designing interactions using OAI model: A new interface modeling paradigm. In*Emerging Technologies, 2009. ICET 2009. International Conference on* (pp. 422-426). IEEE.

[19] Crystal, A., & Ellington, B. (2004). Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. *AMCIS 2004 Proceedings*, 391.

[20] Annett, J. (2003). Hierarchical task analysis. *Handbook of cognitive task design*, 17-35.

[21] Spillers, F. (2003). Task Analysis through Cognitive Archeology. *D. Diaper y N. Stanton. The Handbook of Task analysis for Human-Computer Interaction, Eds, Lawrence Erlbaum Associate, London*, 279-290.

[22] Norman, D. A. (2002). *The design of everyday things*. Basic books.

[23] Virzi, R. A. (1989, October). What can you learn from a low-fidelity prototype?. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*(Vol. 33, No. 4, pp. 224-228). SAGE Publications.

[24] Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *interactions*, *3*(1), 76-85.

[25] Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *interactions*, *3*(1), 76-85.

[26] Horizontal and vertical prototyping: Retrieved on July 29, 2009 from http://www.great-web-design-tips.com/web-usability/115.html.

[27] Dillon, A. (2001). The evaluation of software usability. Encyclopedia of Human Factors and Ergonomics.

[28] Wania, C. E., Atwood, M. E., & McCain, K. W. (2006, June). How do design and evaluation interrelate in HCI research?. In Proceedings of the 6th conference on Designing Interactive systems (pp. 90-98). ACM.

[29] Nielsen, J. (1994, April). Usability inspection methods. In Conference companion on Human factors in computing systems (pp. 413-414). ACM.

[30] Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994, June). The cognitive walkthrough method: A practitioner's guide. In Usability inspection methods (pp. 105-140). John Wiley & Sons, Inc.

[31] Toshihiro, K. O. M. I. Y. A. M. A. (2008). Usability Evaluation Based on International Standards for Software Quality Evaluation. NEC Technical Journal, 3(2), 27-32.

[32] Dumas, J.S., User-based Evaluations. In: Jacko, J.A. & Sears, A. (Eds). The Human-Computer Interaction Handbook: fundamentals, evolving technologies, and emerging applications. Lawrence Erlbaum Associates, Inc., New Jersey, (2003).