2006

Usability in Requirements Engineering

Sisira Adikari
University of Canberra, Sisira.Adikari@canberra.edu.au

Craig McDonald
University of Canberra, craig.mcdonald@canberra.edu.au

Neil Lynch
University of Canberra, neil.lynch@canberra.edu.au

Usability in Requirements Engineering

Sisira Adikari
Craig McDonald
Neil Lynch

School of Information Sciences and Engineering
University of Canberra
Canberra, ACT
Email: {Sisira.Adikari, Craig.McDonald, Neil.Lynch}@canberra.edu.au

Abstract

Usability is a determining factor in information systems acceptance. Despite the presence of various usability engineering techniques to produce usable information systems, usability issues are still identified late in the development process during final system/product testing, or worse, after the system is live. One of the critical reasons for this usability failure is the inadequacy of current requirements engineering practice to transform usability perspectives effectively into software requirements specifications. In this paper, we propose a design-oriented framework incorporating user modelling and usability modelling to produce user-centred software requirements and a research design to verify its validity.

Keywords

User-centred software requirements, user modelling, usability modelling

INTRODUCTION

Usability has been recognised as an important factor in information systems acceptance. For last two decades or more, there have been various user-centred design techniques developed to produce usable information systems. There are no "usability issues-free" information systems yet and usability related deficiencies are still detected late in the development process, during testing and deployment (Folmer and Bosch, 2004, p.62). Some of the identified reasons for the presence of poor usability in products are that usability requirements are often weakly specified, usability requirements engineering techniques have only a limited ability to capture all requirements, and usability requirements may change during development (Folmer, Gurp, & Bosch 2004, pp.321-339). According to Bevan (1999, p.764), the traditional approach to systems design is to build systems that meet specific functional requirements without a sufficiently detailed understanding of the cognitive and physical capabilities and expectations of the intended users. He further argues that traditionally, requirements engineering concentrates on functional requirements and ensuring that the developed products meet these requirements, rather than other non-functional requirements, which are considered less important. Usability has been classified as a non-functional requirement in Requirements Engineering (Sommerville, 2004, p.122). Keeping the usability as a rather less important non-functional requirement may cause designers to pay less attention during the requirements definition stage. From an information systems perspective, one of the obvious reasons for usability issues in end products is the insufficient specification of usability in software requirements specifications. Information systems development is based on software requirements specifications, hence information systems built on less usability focused specifications are likely to propagate usability issues into end products. As a result, end-user experience and satisfaction are directly affected.

In practice, the usability of a product is created by the designers of the product. Jokela(2006) identified that usability engineering techniques and design actions may not be adequate for making products usable if the results of the design evaluations do not impose an impact or influence on the design solutions. To address this issue and as a possible solution, he further emphasised that one of the most-effective means for bridging the gap between usability and design as cross-functional teamwork involving designers and usability practitioners to work together going through usability issues (Jokela, 2006). However, the final product of the information system is realised by the developer of the product based on software requirements specifications. Therefore, there is a real need in practice to translate usability aspects effectively into software requirements specifications. In relation to the software life cycle, inclusion of usability specification as part of the requirements specification will contribute to the usability of the product (Dix et al. 2004 p.237).

In this paper, we propose design-oriented requirements modelling based on user modelling and usability modelling as an effective means to transform usability aspects into software requirements specifications. We explain the user-centred design process and how to integrate user-centred design into a typical software development life cycle. Finally, we present a research design which can be used to test our proposal to determine if user modelling and usability modelling make a real difference in software requirements specifications.

SOFTWARE DEVELOPMENT PROCESS

The main objective of software development process models is to provide the guidance for requirements definition, analysis, design, development, and release of software products. Although each model has its unique capabilities, most of the models have similar objectives, tasks, and follows a particular life cycle. The requirements definition or the understanding of the problem domain is an essential and important activity in every software development process models. Some of the common and widely used software development process models are; the waterfall model, evolutionary development, component-based software engineering, and rational unified process (Sommerville, 2004, pp.66 - 71). For small or medium-sized systems, agile methods embody good software engineering practice (Sommerville, 2004, pp.396 - 398). However, according to a survey published in 2003, the waterfall model is still popular among software engineering community (Neill and Laplante, 2003). The waterfall model is illustrated in Figure 1.

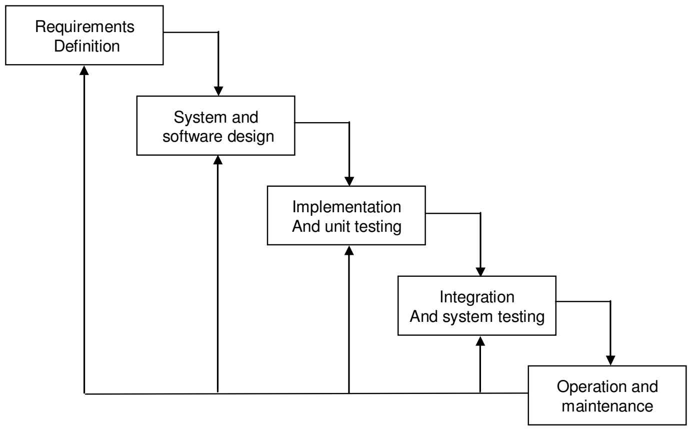

Figure 1: The waterfall model life cycle (Sommerville, 2004, p.66)

One phase proceeds to the next phase only when its preceding phase is complete and achieved defined goals. The advantages of the waterfall model are that artefacts and documentation are produced at each phase and it fits with other engineering process models (Sommerville, 2004, p. 67). The main criticism of waterfall model is about its inflexibility to respond to changing requirements. However, use of waterfall approach for system development is evident in many organizations. (Kruchten, 2004).

USABILITY ENGINEERING PROCESS

Usability engineering is the paradigm for developing usable products and it is also called human-centred design or user-centred design (Jokela, 2003). The International Standards Organization (ISO) has developed a number of standards on usability in two major categories; product-oriented standards, and process-oriented standards. ISO 9146 and ISO 145988 are considered as product-oriented standards while ISO 9241 and ISO 13407 are considered as process-oriented standards. The ISO 13407 standard intends to provide guidance of best practice in usability engineering (ISO 13407, 1999). As indicated in the standard ISO 13407, the rationale for adopting a user-centred design process are to make systems easier to understand and use, improve user satisfaction and reduce discomfort and stress, improve the productivity of users and the operational efficiency of organisations and to improve product quality, appeal to the users towards competitive advantage. The basic principles in usability engineering will involve user participation, iterative design process in a multi-disciplinary and collaborative teamwork. The usability engineering processes that have been defined in ISO 13407 standard are illustrated in Figure 2.

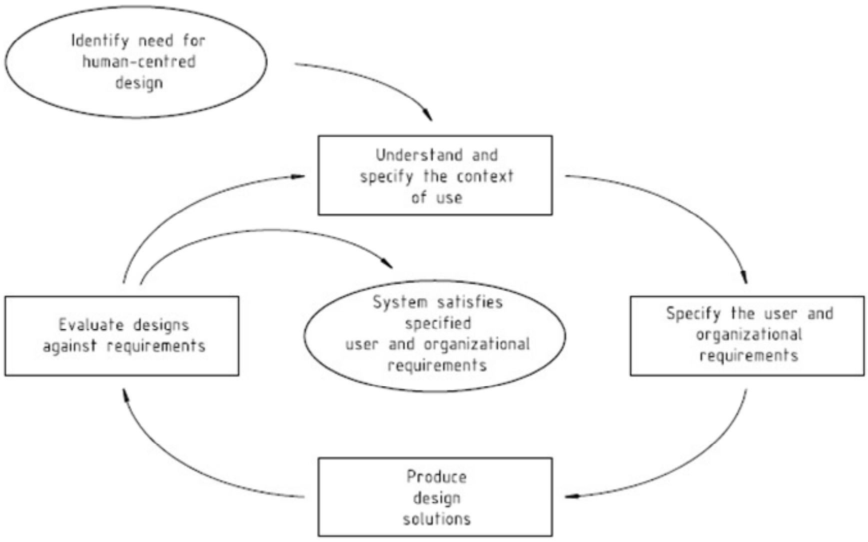

Figure 2: Usability engineering processes (ISO 13407, 1999 p.6)

The summary of activities of the ISO 13407 is:

- *Understand and specify the context of use.* Identify the characteristics of the user, tasks of the user, and the environment in which the user intends to use the product.
- *Specify the user and organisational requirements.* Identify the requirements of user task performance, user interface, work design and organisation towards user-centred design goals.
- *Produce design solutions.* Apply existing knowledge to the design and make design solutions more concrete using simulations until user-centred design goals are met.
- *Evaluate design against requirements.* Evaluate the design involving real users carrying out user tasks to ensure that user and organisational requirements have been met.

Jokela et al. carried out an in-depth interpretive analysis of ISO 13407 and identified that the standard provides limited guidance for designing usability, describing user goals, usability measures and producing the various outcomes (Jokela et al. 2003). They commented that ISO 13407 alone was not providing adequate guidance for using the standard definition of usability and it should also emphasize in detail how valid and effective outcomes can be achieved.

Although the ISO 13407 provides a general guidance to user-centred design activities, our analysis of the standard yields two important aspects which are not clearly visible;

- how the evaluation feedback can be *effectively* used to improve the design as well as requirements.
- how the process model altogether can be used to support a typical software development life cycle.

EVALUATION FEEDBACK TO IMPROVE THE DESIGN AND REQUIREMENTS

In ISO 13407 process model shown in the Figure 2, the output of the activity 4 (Evaluate design against requirements) feeds back to activity 1 (Understand and specify context of use). In practice, there are situations where it is required to apply the outcome of the evaluation immediately to the design to improve the design prior to the next iteration. In producing effective design solutions, we argue that it is quite important to feedback the output of the activity 4 (Evaluate design against requirements) *also* into activity 3 (Produce design solutions). Our suggested variation is shown in the Figure 3.

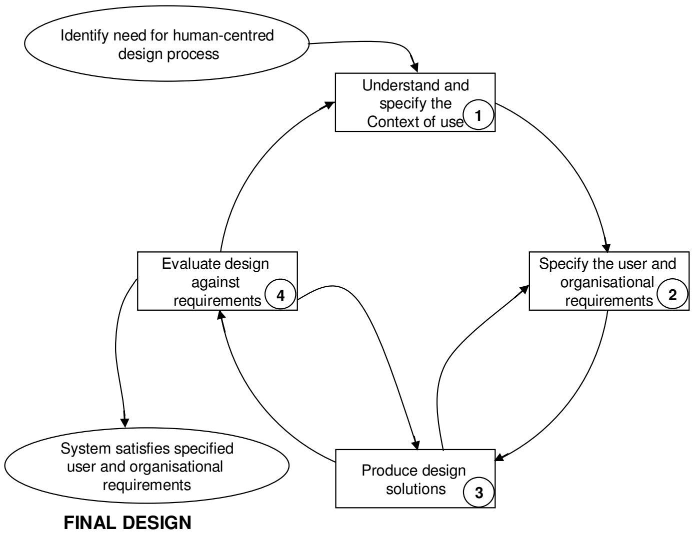

Figure 3: A variation to the ISO 13407 incorporating evaluation feedback to improve the design

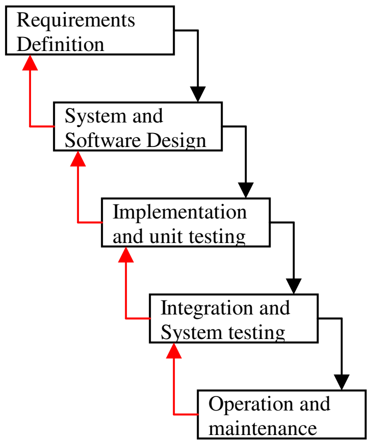

Figure 4: Waterfall model showing iteration with the preceding and succeeding phases

Figure 4 presents Royce's original illustration of the waterfall model which shows the iteration with the preceding and succeeding phases (Royce, 1970). There is a similarity between Figure 3 and Figure 4. In Figure 4, the feedback from testing phases to the system and software design phase, and the feedback from the system and software design phase to the requirements definition phase resemble in Figure 3: feedback from evaluation to design and then to the requirements. That is, our suggested variation to the ISO 13407 is in accordance with the foundation model of the software development process illustrated in Figure 4.

INTEGRATION OF ISO 13407 INTO SOFTWARE DEVELOPMENT PROCESS

In order for a system to be usable, that has to be designed with a focus on usability and the design need to be the basis for the succeeding implementation. The prime objective of user-centred design practice is to provide design solutions which satisfy the user and organisational requirements as the basis for usable systems development. Usability engineering or user-centred design does not clearly give the guidance as to how the design solutions can be used to implement a usable system. How system developers use the design solutions to implement a usable system is an open question which still has no a clear answer. The software requirements specification is the official statement which is used by system developers to implement the system (Sommerville, 2004, p.136). In implementing usable systems, we argue that usability focused design solutions need to be the basis for software requirements specifications. In Figure 2, when the system satisfies user and organisational requirements (Final Design), the requirements at stage 2 (Specify user and organisational requirements), can be considered as the requirements of the best design solutions of the product. Hence, we argue that if these requirements are fed into the requirements definition stage of a typical software development life cycle, requirements definition process will be more user-centred from the beginning. The Figure 5 illustrates our suggested integration of the ISO 13407 process model into a typical software development life cycle, in this case, the waterfall model. As Royce's original illustration of the waterfall model suggests iterations with the preceding and succeeding phases (See Figure 4), our suggested integration model is of iterative nature. Similar integration is possible with other software development processes such as prototyping. The information flow between requirements stages of the software development life cycle (SDLC) and the ISO 13407 is two way, and it benefits each other. There are many significant advantages of integrating the ISO 13407 process model through *requirements* into a typical software development life cycle.

- Requirements definitions are more user-centred and task oriented.
- Requirements definition phase can be completed fairly quickly.
- System and software design phase can be driven with concrete design solutions leading to lesser turnaround time.
- Testing phase can more readily be user-centred as the requirements specifications are user-centred and the readily availability of design solutions to devise usability-focused test specifications.

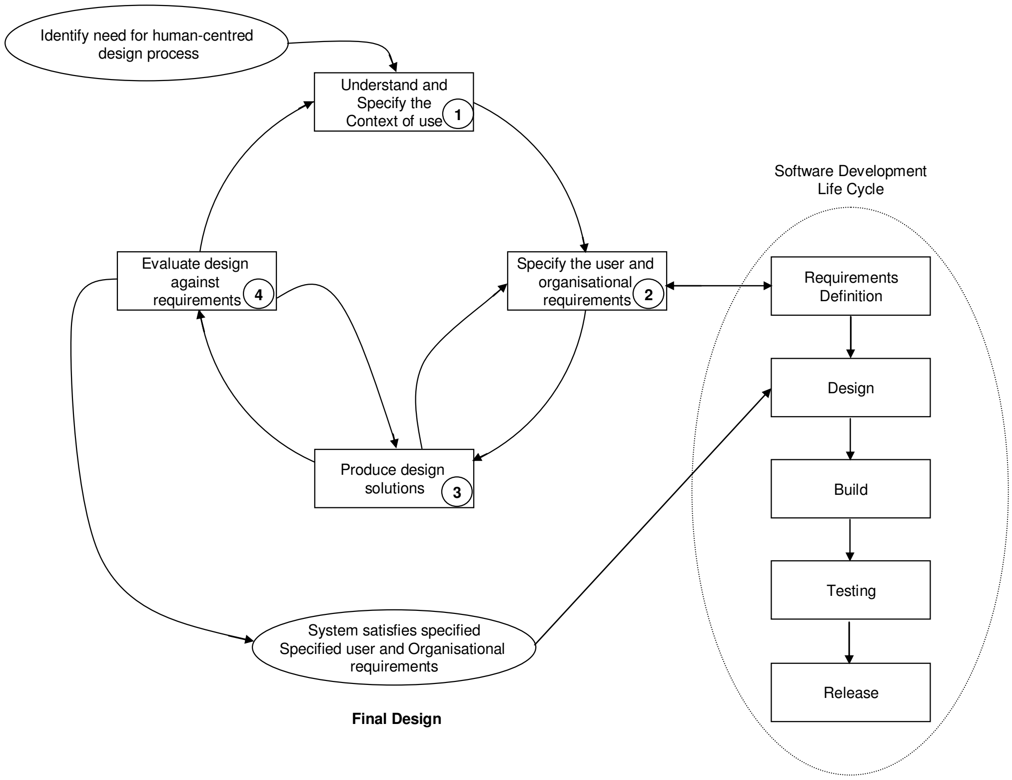

Figure 5: The integration of ISO 13407 process model into software development cycle

The key challenge of any user-centred design process is to communicate the user-centeredness effectively to the designer as well as to the developer. The ISO 13407 lacks a clear guidance for designing usability or producing outcomes.

DESIGNING USABILITY INTO REQUIREMENTS SPECIFICATIONS

To address the deficiency, we propose usability modelling and user modelling as important aspects that need to be integrated into any user-centred design process. If the final product meets the functionality requirements, user requirements, and usability requirements altogether, such a product should be very usable.

Conceptual Usability Attribute Model

The term "Usability" has been defined in ISO 9241 part 11 as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Table 1 shows usability attributes derived from an analysis of a wide range of sources in the HCI literature:

Usability Attribute	Definition
Learnability	The ease with which new users can begin effective interaction and achieve maximum performance (Dix et al. 2004, pp. 260-270)
Memorability	The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again (Nielsen, 1993)
Functional Correctness	The system should correctly perform the functions that the user needs (Brink et al. 2002, pp. 2-3)
Efficiency	Expert user's steady-state level of performance (Nielsen, 1993, p.31), Effectiveness of task accomplishment in terms of speed and errors (Shackle, 1991)
Error Tolerance	System should have low error rate as well as easy error recoverability (Nielsen, 1993, pp. 32-33.)
Flexibility	The multiplicity of ways in which the user and system exchange information (Dix et al. 2004, pp. 260-270)
Attitude / Satisfaction	User's subjective opinion about the system in terms of tiredness, discomfort, frustration and personal effort (Shackle, 1991)

Table 1: Usability attributes and definitions

Figure 6 illustrates the conceptual usability attribute model to show that the usability is a combination of several attributes and each attribute is governed by several usability related measurable aspects of the system or product. The details of the measurable criteria for each usability attribute are detailed in Figure 7.

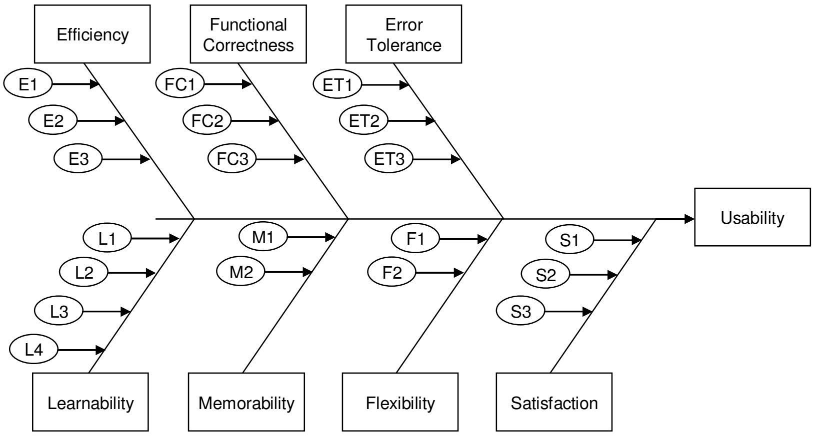

Figure 6: Conceptual usability attributes model

Efficiency	Functional Correctness	Error Tolerance
E1 – Task completion in minimum time	FC1 – Task completion in minimum time	ET1 – Appropriate error messaging for invalid conditions
E2 – User tasks are not misleading	FC2 – User tasks are appropriate, effective and match the user requirement	ET2 – Ability to exit error conditions or unwanted states
E3 – No workarounds are needed	FC3 – User spends minimal time on "Help"	ET3 – No workarounds are needed

Satisfaction
S1 – User desirability of the system and user tasks
S2 – User opinion about user experience
S3 - User opinion about frustration or confusion

Memorability	Learnability	Flexibility
	L1 – Clear visibility of current system status and a feel about what to do next	
	L2 – User tasks are not misleading	
M1 – No memory recall to carry out tasks	L3 – Task completion in minimum time	F1 – Multiplicity of ways to carry out user tasks
M2 – User spends minimal time on "Help"	L4 – User spends minimal time on "Help"	F2 – User control of task performance

Figure 7: Usability attributes and measurable criteria

We propose that each and every user task of a system needs to be designed based on a conceptual usability attribute model similar to shown in the Figure 6. Such an approach will ensure that the user interaction is efficient, functionally correct, error tolerant, learnable, memorable, satisfying, and meet the usability measurement criteria shown in Figure 7. These all together will lead to generate a higher level of positive end user experience and acceptance.

Conceptual User Model

User interface design is based on user models and descriptions derived from studies of able-bodied users (Keates, Clarkson & Robinson, 2005). A user model is a representation of information and assumptions about users (Kobsa, 1995, pp.155-157) and can be viewed from three perspectives: modelling user knowledge, modelling user plans, and modelling user preferences (Kobsa, 1993). Modelling user knowledge will involve the accurate estimation of users' background knowledge, skills, and experience. Modelling user plans aims to investigate about the sequence of user tasks required to achieve user goals. Modelling user preferences primarily focus on users' information needs and preferences. A user model can be used to improve the usability of computer interfaces and ideally the user model needs to match the design model (Norman, 1998, pp.189-190). In general terms, the attributes of a user model are context dependent and vary across the application domains, but there are certain user attributes, which are important in relation to a user model. The Figure 8 outlines our proposed conceptual user model with important user attributes.

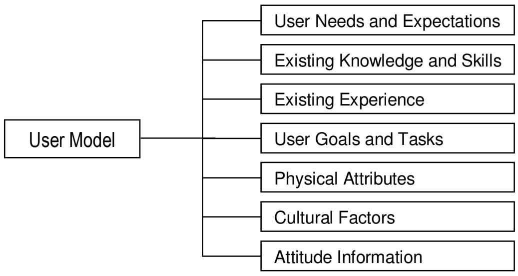

Figure 8: Conceptual user model

We also propose that user tasks of a system need to be designed based on a conceptual user model similar to the above to ensure that each user interaction matches the user needs and expectations, existing knowledge and experience, user goals and tasks, users physical attributes, cultural factors and the attitude. Our proposal will produce a new set of requirements in relation to user interaction and the interface. These new requirements can be incorporated with functional requirements to produce user-centred software requirements specifications on which the software development process can be progressed on.

RESEARCH DESIGN

The aim of the research design is to test the proposal and to determine the impact of user modelling and usability modelling on software requirements. Our proposed research design is illustrated in Figure 9.

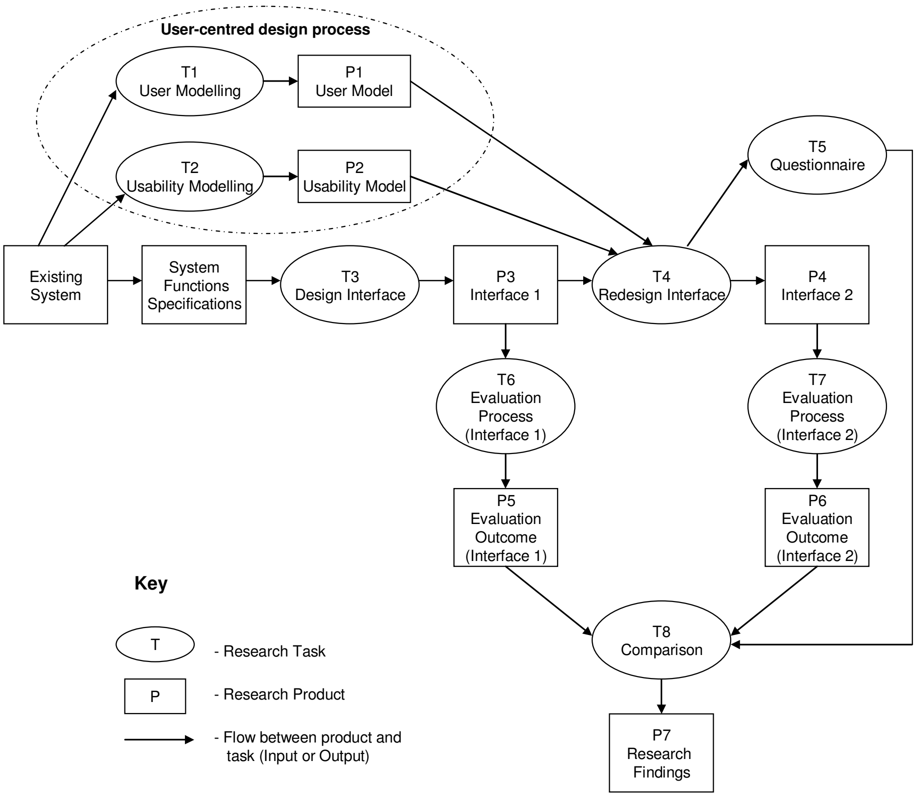

Figure 9: The research design

We provide system functions specifications of an existing system to designers so that designers will be able to design and produce the first interface. In another activity, user-centred designers will carry out an investigation into the context of use of the existing system and pursue a user-centred design process involving user modelling and usability modelling to produce two artefacts: user model and the usability model. This activity will involve active user participation and to develop a rich understanding of the context and knowledge about the user and user tasks. The outcome of the activity, the user model and usability model then will be provided to the same designers who produced the first interface to redevelop it and to produce an enhanced version of the interface. In this step, designers will be requested to complete a questionnaire to input their opinions, views, and experience they encountered during the design process. Following that, interface testers will carry out a usability evaluation process on both interfaces involving real users to determine the functionality and usability of the interfaces. Finally, outcomes of the evaluations and the questionnaires will be analysed to come up with the research findings.

Summary of activities in the research design:

T1 & T2
- User-centred designers carry out user modelling and usability modelling producing artefacts: user model and usability model.

T3
- Systems designers design and produce Interface1 based on system functions specifications.

T4
- Systems designers redesign the interface and produce Interface 2 using user model and usability model.

T5
- Systems designers fill out a questionnaire expressing their views, design experience and opinions on redesigning the interface.

T6 & T7
- Interface testers carry out an evaluation process against user requirements and usability requirements on both interfaces involving end users.

T8
- Outcome of the evaluations are compared to come up with research findings.

SUMMARY AND DISCUSSION

In this paper, we have outlined a framework for our research to integrate user-centred design capability into software engineering development through “requirements” and “design”. More design iterations in the framework aim to make the requirements and design right and user-centred. We have looked at usability from two perspectives: user modelling and usability modelling. We have proposed user modelling and usability modelling as two key requirements modelling approaches to enrich the design towards higher level of usability. Our framework links requirements and design from user-centred design process into software development process ensuring that requirements definition phase of a typical software development model is more user-centred at the first instance. This approach will facilitate to generate user-centred software requirements specifications effectively and efficiently. Moreover, the user-centred design artefacts will be a value-added aid to the build phase of a typical software development model. The research design is expected to verify the impact of user modelling and usability modelling on software requirements specifications. Altogether, the framework and the research design may make a contribution to knowledge by theory building and practice with effective and practical techniques to produce usable end products.

The main strength of the user-centred software requirements specifications is the clear visibility of usability aspects for developers and testers. Developers will be able to incorporate usability aspects effectively into systems and testers will be able to test systems effectively and efficiently to uncover functionality issues as well as usability issues so that any system that goes “live” will be with much richer positive end-user experience. We expect that the proposed framework is capable in transforming usability aspects effectively into software requirements specifications, hence minimising the current usability related issues in end products. The proposed framework is expected to fill the current gaps in Usability Engineering and Requirements Engineering in producing usable systems effectively.

REFERENCES

Bevan, N. (1999), Design for Usability, Proceedings of HCI International (pp. 762-767). Munich, Germany.

Folmer, F., & Bosch J. (2004). Architecting for Usability: A Survey, Journal of Systems and Software, Volume 70(1). Amsterdam: Elsevier. 61-78.

Folmer, E.; Gurp, J. & Bosch, J. (2004). Software Architecture Analysis of Usability, The 9th IFIP Working Conference on Engineering for Human-Computer Interaction (pp. 321-339). Hamburg.

ISO 13407:1999(E). (1999). Human-Centred Design Processes for Interactive Systems.

Jokela, T., The KESSU Usability Design Process Model. Retrieved July 28, 2006, from http://www.tol.oulu.fi/~tjokela/KESSUUDProcessModel2.1.pdf.

Jokela, T., (2003). The Assessments of Usability Engineering Processes: Experience from Experiments. Retrieved 26 July, 2006 from http://csdl2.computer.org/comp/proceedings/hicss/2003/1874/09/187490328a.pdf.

Jokela, T., N. Iivari, J. Matero, and M. Karukka. (2003). The Standard of User-Centred Design and the Standard Definition of Usability: Analyzing ISO 13407 against ISO 9241-11. Retrieved July 30, 2006, from http://delivery.acm.org/10.1145/950000/944525/p53-jokela.pdf?key1=944525&key2=4303774511&coll=portal&dl=ACM&CFID=74456598&CFTOKEN=24876680.

Keates, S., Clarkson, P.J. & Robinson, P. (2005). Developing a methodology for the design of accessible interfaces. Retrieved July 30, 2006, from http://ui4all.ics.forth.gr/UI4ALL-98/keates.pdf.

Kobsa, A., (1995): Supporting User Interfaces for All Through User Modeling, In Proceedings HCI International '95 (pp. 155-157). Yokohama: Elsevier.

Kobsa, A., (1993): User Modeling: Recent Works, Prospects and Hazards. Retrieved July 28, 2006, from http://www.ics.uci.edu/~kobsa/papers/1993-aui-kobsa.pdf.

Kruchten, P. (2004). Going over the waterfall with the RUP. Retrieved July 30, 2006, from http://www.-128.ibm.com/developerworks/rational/library/4626.html.

Nielsen, J. (1993). Usability Engineering (pp. 26-37). San Diego, CA: Academic Press, Inc.

Neill C. J., & Laplante P. A. (2003) Requirements Engineering: The State of the Practice, IEEE Software, vo1 20, no.6, pp 40-45.

Neill C. J., & Laplante P. A. (2003) "The demise of the waterfall model is imminent" and other urban myths. Retrieved July 30, 2006, from http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=110.

Norman, D.A. (1998), The Design of Everyday Things. The MIT Press. London, England.

Royce, W. (1970). Managing the development of large software systems. Retrieved July 30, 2006, from http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf.

Shackel, B. (1991). "Usability – Context, Framework, Definition, Design and Evaluation". In B. Shackel & B. Richardson (Eds.), Human Factors for Informatics Usability (pp.21-37). Cambridge: Cambridge University Press.

Sommerville, I (2004), Software Engineering. 7th ed. Reading MA: Addison-Wesley.

COPYRIGHT