

Assessment of Software Interfaces using a Usability Evaluation Based Software Model

Saad Masood Butt, Wan Fatimah Wan Ahmad
 Computer and Information Sciences Department
 Universiti Teknologi PETRONAS
 Tronoh, Perak, Malaysia
 saadmasoodbutt668@yahoo.com
 fatimhd@petronas.com.my

Abstract-- The usability evaluation of the software interfaces is one of the prominent concepts in Human Computer Interaction (HCI). In order to increase the usability, the design of software interfaces becomes an important task for HCI experts. This paper describes a new HCI based software development approach called the Usability Evaluation based Model (UEM). The UEM consists of five processes that help Software Engineers work with HCI experts from the beginning until the deployment of the software. However, this model not only bridges the gap between the SE and HCI experts, but it also links the AI experts to make the development process more intelligent. One of the processes in the UEM is the Cognitive Analysis of the Software Interfaces (CASI). CASI is an expert system that helps designers and software developers to evaluate software prototypes in an intelligent way based on user perception and evaluation rules. This paper also presents a case study on the development of the university online classroom booking system. The system was built using the UEM. The results mentioned in this paper show that with the help of the UEM, more usability problems in the software interfaces can be detected. Hence, enhancing the usability of the software interfaces by an automated UEM is feasible.

Index Term-- Cognitive Science, Software Interface, Software Engineering, Artificial Intelligence, Expert System, Usability Evaluation, Usability Engineering, User Interface, Interaction User Prototyping, Human Computer Interaction, Cognitive Analysis of Software Interface .

I. INTRODUCTION

An effective source to exchange information and interaction between a user and a computer are software interfaces. Designing a software interface that is easy to use, easy to learn and easy to memorize are the attributes of the software usability evaluation [1]. Therefore, the software usability evaluation is an important concept in the HCI.

In designing the software interfaces, the SE and HCI need to understand the user behaviours, user familiarity with different features of the software interface and the user expertise while working with other software interfaces. The HCI deals with social, cognitive and interaction phenomena. Where the social layer focuses on how people interact with each other as well as with technology.

In the HCI, usability engineering plays an important role to achieve user goals in an effective, efficient and satisfying way. It is a discipline that helps to achieve usability during the design of software interfaces. Usability engineering itself is a vast topic but usability evaluation is the part that contains various techniques like heuristic evaluation, guideline reviews and the cognitive walk-through [2].

In this paper, a model for the software development is presented to help SE, HCI and AI experts to work together to produce high interactive interfaces in a software system to achieve the user's goals. The most vital task of this model is the development of the expert system called CASI [3]. CASI enables SE and HCI experts to produce an interactive interface that can meet user requirements. This paper is divided into a few sections. Section 2 is on the literature review' section 3 describes different software models, section 4 describes the UEM, section 5 focuses on the Expert system CASI and section 6 discusses the case study of the UEM. In the end, section 6 shows the results and future work.

II. LITERATURE REVIEW

In order to ensure the requirements that were mentioned in the requirement engineering (RE) are fulfilled by the software, the HCI and SE need to work together in the interaction layer of the software development. It is not clear how the HCI and SE experts work together when there is a need to provide a high level of UI Usability.

According to [4], a web based tool is recommended to find usability problems in the HE. Such type of tools are beneficial to use as they are accessible especially when distant assessment becomes significantly popular. They also support common problems of interest among with different evaluators, developers and researchers scattered to work in different locations.

As mentioned in [5], "Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services". Today automation is required to perform daily routines and repetitive work. It is also important to automate those software processes that take a considerable amount of time and contain a cycle between various processes. As discussed in [6], the HE evaluators feel that it's difficult to a make report on paper, which is time-consuming and cumbersome. So, there needs to be some AI based interface evaluator system, which is discussed in Section IV.

Nielsen [7] developed 10 heuristics but later 12 heuristics were developed against the original 10 heuristics. Research shows that the modified heuristics are more efficient and capture more defects that were missed by the old heuristics. Despite these benefits, some research shows the pitfall of the HE. It shows that the HE does not find as many defects as other Usability Engineering methods. The single evaluator may be able to find only a small percentage of defects, so it is useful to involve more than one evaluator

and later their results are aggregated [8]. According to [9], a software development team needs to focus on the functionality of the system as well as increase the usability of the software during the SDLC. One of the methods used in Usability Testing is the Heuristic Evaluation (HE). The HE is a good method to find major and minor problems in the software interface. The HE's main goal is to find usability problems in the software interface so that they can be attended as part of the software design process.

J. Brown [10] discussed the role of the HCI in the Information System development (IS). A new methodology named the Human Centered System Development is proposed which is the combination of the system development life cycle (SDLC) with the human computer interaction (HCI) in information systems (IS). As in the traditional development lifecycles of the IS, the role of the HCI is too low only at the design phase or at a later stage that affects the overall development. Thus, there is a gap found between the HCI and the SE, and in order to remove this gap the human-centered IS development approach was introduced.

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

The experimental model in [12] shows that adopting only one technique for the requirement elicitation is not appropriate. An integrated based approach for the requirement elicitation is much better and helpful to get the correct requirements. The experimental model consists of two folds: (1) to encourage the business analyst not to restrict themselves to the standard approaches of requirement gathering and (2) getting incomplete requirements is due to adopting only one technique for requirement elicitation; the best way is to adopt an integrated based approach for the requirement elicitation.

Soundararajan [13] proposed a framework which is the combination of traditional and agile software development, and it helps to handle rapidly changing requirements in building large-scale systems. The framework has two parts: (1) an agile approach of soft structured requirement gathering and (2) a development process for small and large systems.

Davis et al. [14] discussed the various requirement gathering techniques. Interviews are one of the techniques in the RE used to gather requirements. But interviews are not an effective way of getting requirements; moreover, this will not help to get clear requirements. Interviews only help to give a clear understanding of a particular topic.

III. OTHER SOFTWARE MODEL

There are a number of Software models that are used in Industry. This paper will review the following five models: the Waterfall model, Iterative model, V-shaped model, Spiral model and Extreme model.

The waterfall model is a sequence process model in the field of software. This model is still used in various companies. This model focuses on planning in the beginning, it guarantees design faults before they are created [17].

The iterative model is one of those models that fall's in the category of agile software development models. It focuses on initial requirement gathering. In the iterative model, if any requirement is left behind during the requirement gathering phase it will be covered in the next phase of an iteration. These models produce good results as long as they do not focus on a user interface (UI) [18].

The V-model is the traditional software engineering process and it starts coding at a very early stage as soon as a few requirements have been gathered from users. Each phase must be completed before the next phase begins. Implementation of the code takes place in small increments and iterations. More testing is required in this model as compared to the waterfall model. In the V-model, the customer is supplied with a small release after each development cycle [20].

The spiral model is one of those models that combines the best features of the waterfall model and the prototyping model. The spiral model contains four phases: Planning, Risk Analysis, Engineering and Evaluation. The project repeatedly passes through these phases in iterations.

Another software model in the software development discipline is the Extreme Model. The Extreme Model helps software engineers to develop and deliver small increments of functionality to users. Therefore, the continuous involvement of users helps to improve the codes in the software development process.

The weaknesses mentioned in various software models, either traditional or agile, are the source of the product failure. Some major problems need to be highlighted such as:

- The 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 Evaluation

Hence, it seems that there needs to be a cross discipline software development that can fulfill the agile approach of the SE as well as the infused HCI approach. The proposed UEM proves to be a complete model that consolidates both the SE and HCI approaches.

IV. USABILITY SOFTWARE MODEL (UEM)

Gathering requirements for the agile software development is very crucial. Agile focuses on completing a task in a less amount of time and providing complete functionality that was stated by the user during the requirement gathering phase. For the past few years, it has seemed that the expert system may help the SEs to complete their tasks in less amount of time in an efficient way. Therefore, The Usability Evaluation Based model (UEM) has been developed to automate the software development process. The UEM consists of five processes: Requirement Gathering, Interaction User Prototyping, Expert System CASI, and Development and Deployment as shown in Figure 1. There

is a cycle between the first four processes of the UEB. Anything unclear at any phase needs to go back to its previous phase in order to fix the problem.

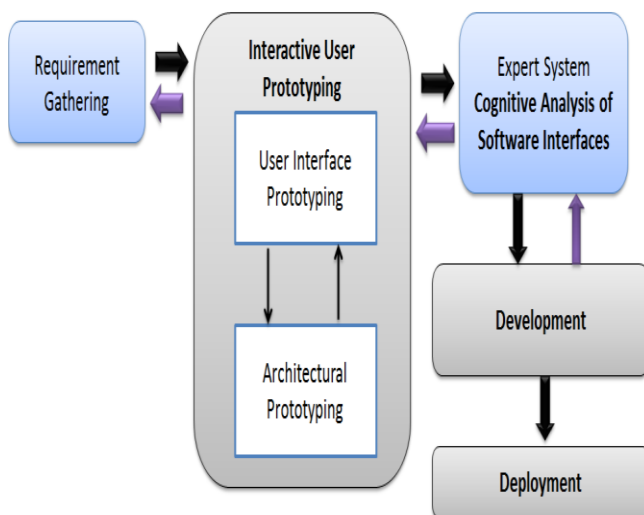


Fig. 1. UEB Model

Requirement Gathering

It is important to get a stable set of requirements before the system design and implementation starts [15]. In this phase, the requirement from the user needs to be documented. Gathering the right requirements are an important phase in software development. For the proposed UEM model, open end questions will be asked of the user to gather the initial requirements.

Interaction User Prototyping

The second phase of the UEM is the Interaction User Prototyping (IUP). The IUP consists of two parts: the User Interface Prototyping (UI) and the Architectural Prototyping (AP). The IUP helps to design prototypes both at the user level and the architectural level. In the user interface prototyping, while making prototypes of the User Interface features, it does not consider the functionality or architecture. Whereas in the Architectural Prototype, it does not focus on the User Interface, instead the prototype is built to focus on the hidden architecture. After getting the requirements from the first phase of the UEM, the development of the prototypes will start. Missing features that are found during the prototype development can be solved by going back to its previous process.

Expert System CASI

The third phase of the UEM is an expert system CASI. Section V describes the functionality of this expert system. The expert system evaluates the usability of the interface per prototype, which are produced from the result of the IUP. CASI contains a series of Rules defined either by the user to evaluate the prototype or the system itself defines Rules to evaluate the prototype. If the prototype evaluation result is not up to the user's rules or the system's rules, then the prototype needs to be revised.

Development and Deployment

The fourth phase of the UEM is the development of the software. After passing from the expert system CASI, the

development of the software starts based on the prototypes. Sometimes, new requirements or modifications in the existing requirements are requested by the user. So, the UEM can handle new requirements or modify any existing requirements.

At the end, the software is deployed at the user side and the UEM processes are completed.

V. EXPERT SYSTEM CASI

The expert system evaluates the interface per prototype and works based on the concept of inference [16]. In this expert system, there are some defined Facts and Rules. The Facts are like inferences and on the basis of these Facts, some Rules are defined; these are then stored in an Inference Engine. The Rules are defined by the user and are stored in the Inference Engine. The Rules are either self-defined or system defined. The self-defined Rules are based on a user's interest whereas the system defined Rules contain a combination of Heuristic and Cognitive walk through. These Rules help to evaluate the user prototypes and architectural prototypes.

In this paper, the author discussed a case study of an online classroom booking system and focused on user defined Rules. The expert system CASI contained three phases.

- a. Facts and Rules
- b. Decision Tree
- c. Results

- a. Facts and Rules

For this system, five Rules are defined:

Rule A: Go back to the previous Process, i.e., IUP

Symbol: RA

Rule 1: Easy to use

Means that the prototype makes the task easy to use.

Symbol: R1

Rule 2: Easy to learn

The task is easy to learn and the next time the user performs the same task easily without thinking much.

Symbol: R2

Rule 3: User perception

The interface was designed according to user perception.

Symbol: R3

Rule 4: Easy Mastery

The interface provides enough information that the user does not need to study the Help file.

Symbol: R4

Rule 5: Provided Functionality

All these functionalities are available that were stated by the user during the requirement gathering phase.

Symbol: R5

b. Decision Tree of CASI

Rule R1, R2, R3 and R4 are stored in the Inference Engine. The expert system evaluates the output (that comes from the IUP phase) by R1 . If R1 proves to be correct, the then prototype will move to R2 for the evaluation. If it fails at any Rule then the flow will move towards RA. RA is a state to improve the prototype according to the self-defined or system defined Rules.

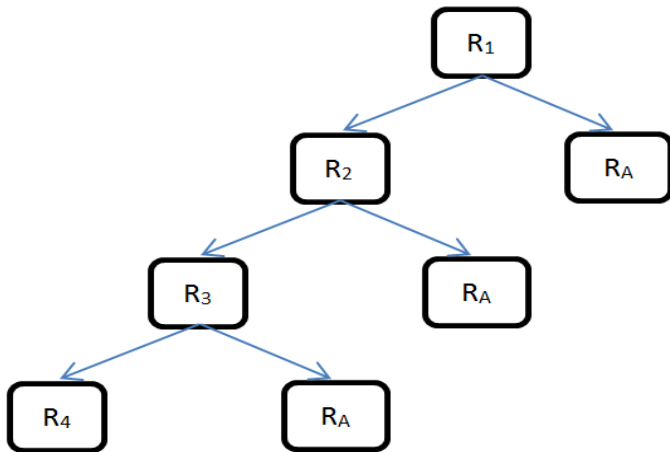


Fig. 2. Decision tree of CASI

c. CASI Process

The CASI contains four element's, namely, the Process, Knowledge Base, Inference Engine and Database. Figure 3 depicts the clear understanding of the flow of the process between these elements.

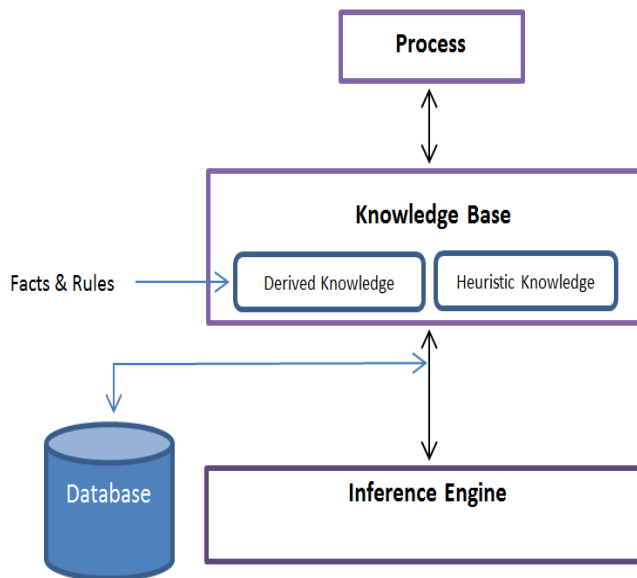


Fig. 3. CASI Process

VI. EXPERIMENTAL MODEL

In this section, the authors discuss the case study which is the development of the university online classroom booking system that was built by using the UEM. Each prototype is

tested by the expert system CASI. Further improvement is noted where the expert system cannot evaluate according to the user's perception.

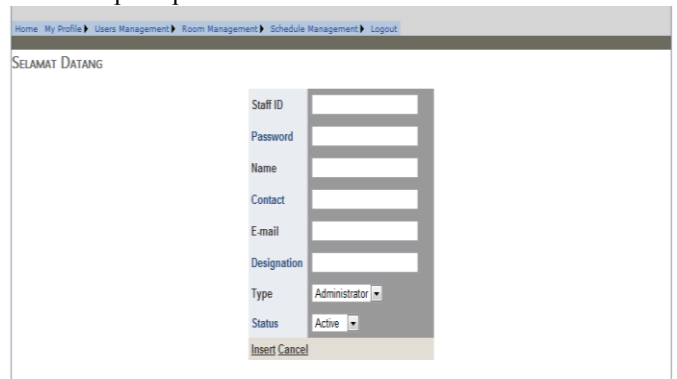


Fig. 4. Main Page

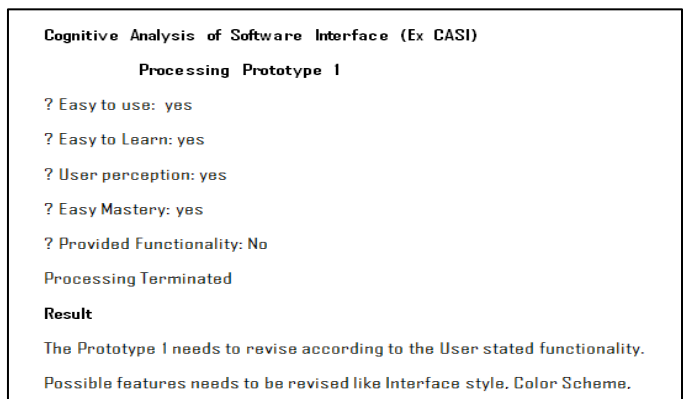


Fig. 5. Expert system CASI Evaluates Main Page

Figure 5 shows the results the of expert system CASI while evaluating the Main Page. Termination occurs where any Rule fails to achieve the user's goal. Similarly, Figure 7 shows the results of the Result page of the software.

Home My Profile Users Management Room Management Schedule Management Logout																	
Date	Staff Name	8 am	9 am	10 am	11 am	12 pm	1 pm	2 pm	3 pm	4 pm	5 pm	6 pm	7 pm	8 pm	9 pm	10 pm	11 pm
Sunday 20 May 2012	Assoc. Prof. Dr. Abbas B Md Said	ghghg															
Sunday 20 May 2012	Muhayudeen B Shaik Alaudin	sotsots															
Saturday 19 May 2012	Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli	ghghghj															
Saturday 19 May 2012	Assoc. Prof. Dr. Wan Fatimah Bt. Wan Ahmad	ghghghj															
Wednesday 25 Apr 2012	Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli	ghghghj	fhghf	fhghf	fhghf												
Monday 16 Apr 2012	Assoc. Prof. Dr. Wan Fatimah Bt. Wan Ahmad	fhghghj	fhgh	fhghgh													
Sunday 15 Apr 2012	Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli	dsds	sotsots														
Sunday 15 Apr 2012	Assoc. Prof. Dr. Wan Fatimah Bt. Wan Ahmad	xcxc	xcxc														

Fig. 6. Result Page

Cognitive Analysis of Software Interface (Ex CASI)

Processing Prototype 2

? Easy to use: No

Processing Terminated

Result

The Prototype 2 needs to revise according to the User Perception.

Possible features needs to be revised like Language, Interface style, Color Scheme etc.

Fig. 7. Expert system CASI Evaluates Result Page

Home My Profile Users Management Room Management Schedule Management Logout

SELAMAT DATANG

Staff ID	Password	Name	Designation	Extension	Email	Type	Status
112233	sf1	Safinaz Bt Hamzah	Administrator		safinaz@petronas.com.my	Administrator	Active
112244		Muhaijuddeen B Shaik Allauddin	Administrator		muhaijuddeen@petronas.com.my	Administrator	Active
112255	aa	Adeel Anani	Administrator	0163852717	adeel.anani@hotmail.com	Administrator	Active
112372		Dr. Rohiza Bt Ahmad	Senior Lecturer	7477	rohiza_ahmad@petronas.com.my	Staff	Active
115187		Helmi B. Md Rais	Lecturer	7478	helmi@petronas.com.my	Staff	Active
109721		Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli	Assoc. Professor	7430	roharam@petronas.com.my	Staff	Active
109734		Assoc. Prof. Dr. Wan Fatimah Bt. Wan Ahmad	Assoc. Professor	7475	fatimh@petronas.com.my	Staff	Active
110557		Assoc. Prof. Dr. Abas B Md Said	Assoc. Professor	7473	abas@petronas.com.my	Staff	Active
111100		Dr. Suziah Bt. Sulaiman	Senior Lecturer	7415	suziah@petronas.com.my	Staff	Active
111112		Assoc. Prof. Dr. Ahmad Kamil B. Mahmood	Assoc. Professor	7418	kamilh@petronas.com.my	Staff	Active
111278		Assoc. Prof. Dr. Baharum B. Baharudin	Assoc. Professor	7417	baharh@petronas.com.my	Staff	Active
112230		Dr. Halabi B Hasbullah	Senior Lecturer	7418	halabi@petronas.com.my	Staff	Active

Fig. 8. Administrator Page

Home My Profile Users Management Room Management Schedule Management Logout

Select Staff Member to view booking: Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli

	Date	8 am	9 am	10 am	11 am	12 pm	1 pm	2 pm
Delete	Sunday 15 Apr 2012	sdssds Room ID: 02-03-010	sdssds Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room
Delete	Wednesday 25 Apr 2012	ghghgh Room ID: 01-03-025	ghghgh Room ID: 01-03-025	ghghgh Room ID: 02-03-010	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room
Delete	Saturday 19 May 2012	ghghgh Room ID: 01-03-025	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room	Room ID: No Room

staff

Fig. 9. Classroom Booking Page

VII. RESULT AND FUTURE WORK

The paper has briefly illustrated an initial attempt to use the UEM for software development. The goal was to provide a complete model that covers SE, HCI and Usability evaluation factors in one life cycle. The result is based on the analysis of Cost, time and Resources (CTR) and found that the UEM is cost effective, take less time for development and minimum use of resources.

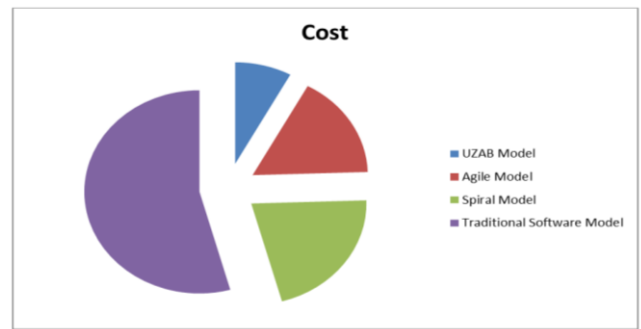


Fig. 10. Result on the basis of Cost

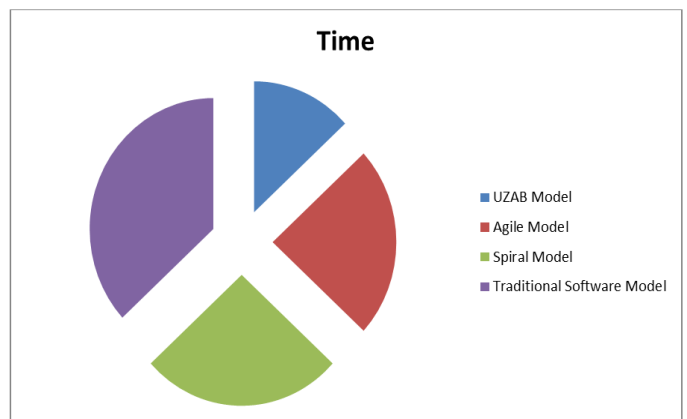


Fig. 11. Result on the basis of Time

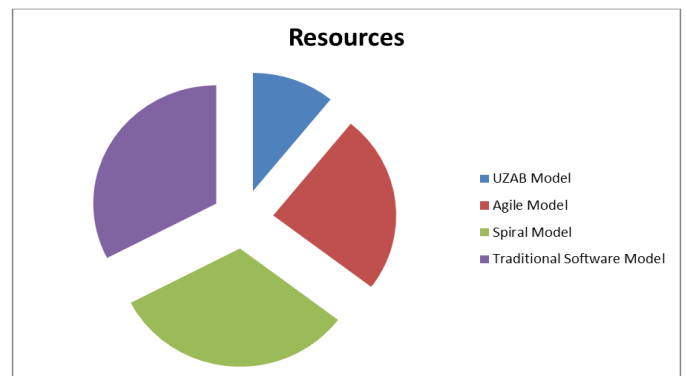


Fig. 12. Result on the basis of Resources

For the future, the UEM will be an effective technique for increasing usability and evaluating the usability of the software during the development of software. Unlike other software models as mentioned in Section 3 that do not focus on user involvement, the interactive user interface handles rapid changes in the requirements and the software interface evaluation, the UEM proves to be an effective technique for software development. Further new FACTS and RULES can be defined to evaluate the software.

VIII. CONCLUSIONS

Overall, this paper has provided a complete description of a model for the SE and HCI Experts to make their software-development process easier and evaluate their software during the development phase using the expert system CASI. This model not only bridges the gap between the SE and HCI experts but also links to the AI experts to make the development more intelligent.

The UEM will be challenging in the beginning when they are provided with the FACTS and RULES to evaluate every prototype of the system. Though, it is a good sign for producing usable systems that can fulfill user's requirements and work up to the user's perception. Effective examining of the UEM will give rise to assess software according to a user's intellect in a true way. However, it is not the last factor to assess software and improve functionality. Further, new ideas and techniques must be considered to enhance the features of the expert system CASI.

ACKNOWLEDGEMENT

The author of the paper would like to thank the Universiti Teknologi PETRONAS and other staff for their valuable feedback during the intermediate phase of the methodology presented in this paper.

REFERENCES

- [1] Yonglei Tao, "Work in progress - introducing usability concepts in early phases of software development", 35th ASEE/IEEE Frontiers in Education Conference, Publication Year: 2009, Page(s): 702 – 706.
- [2] Ritter,F.,E., Baxter,G., D., Jones, G., and Young, R., M., 2000. "User interface evaluation: How cognitive models can help".
- [3] Zhou,P., and Fang,X.,2008. "Analysis of cognitive behavior in software interactive Interface", Computer-Aided Industrial Design and Conceptual Design, CAID/ CD 9th International Conference, pp 13-116.
- [4] Ebba Thora Hvannberg, Effie Lai-Chong Law, Marta Kristín La'rusdóttir, "Heuristic evaluation: Comparing ways of finding and reporting usability problems", Interacting with Computers v19,2007,pp 225–240.
- [5] <http://en.wikipedia.org/wiki/Automation> last accessed 2-6-2012
- [6] Law, E.L.-C., Hvannberg, E.T., 2004a. Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation. In:NordiCHI 2004, Tampere, Finland, pp. 241–250.
- [7] Ashok Sivaji, Azween Abdullah, Alan G. Downe, "Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development", ISBN 978-0-7695-4412-4, Proceedings of 2011 Fifth Asia Modelling, AMS 2011 Conference,pp.68-72.
- [8] <http://www.usabilitybok.org/methods/p275?section=basic-description> last accessed 7-5-2011.
- [9] Kee Yong Lim,"Usability in Singapore",Human-Computer Interaction Series, 2011, Global Usability, Part 2, Pages 285-307
- [10] Integrating Human-Computer Interaction Development into SDLC: A Methodology, Proceedings of the Americas Conference on Information Systems,New York, August 2004.
- [11] J.brown, Sharing Human-Computer Interaction and Software Engineering Design Artifacts, Computer Human Interaction Conference, 1998. Proceedings. 1998 Australasian
- [12] Chua ed.al, "Understanding the use of Elicitation Approaches for Effective Requirements gathering", Fifth International Conference on Software Engineering Advances, IEEE Computer Society, 2010
- [13] Soundararajan, "A Soft-Structured Agile Framework for Larger Scale Systems Development", 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, IEEE Computer Society, 2009
- [14] A. Davis, O. Dieste, A. Hickey, N. Jurist, and A.M.Moreno. "Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review". In Proceedings of the 14th IEEE International. Requirements Engineering Conference, 2006, pp. 176 – 185.
- [15] Gerald Kotonya and Ian Sommerville: Requirements Engineering, John Wiley & Sons, 1997.
- [16] Xunwei Zhou, Hong Bao, "From the Inference System Suitable for Human to the Inference System Suitable for Computer," iitaw, pp.219-223, 2008 International Symposium on Intelligent Information Technology Application Workshops, 2008.
- [17] IBM Study on Software Models, "http://www.theserverside.com/tip/Waterfall-versus-Agile-methods-A-pros-and-cons-analysis"
- [18] Thomas Memmel, Fredrik Gundelsweiler, Harald Reiterer, Agile Human Centered Software Engineering, Published by the British Computer Society, Proceedings of HCI 2007
- [19] The Seven habits of effective iterative development, online [2002], <http://www.ibm.com/developerworks/rational/library/1742.html>, (Accessed: 11 July 2012).
- [20] Saad Masood But & Wan Fatimah. An overview of Software Models with Regard to the Users Involvement. IJCSI International Journal of Computer Science, Issues, Vol. 9, Issue 3, No 1, May 2012
- [21] V-shaped Lifecycle Model, online [2006], <http://www.softdevteam.com/V-shaped-lifecycle.asp>, (Accessed: 11 July 2012)
- [22] Rlewallen, "Software Development Life Cycle Models", 2005 ,<http://codebetter.com>.
- [23] Nabil M Ali & A. Govardhan.. A Comparison Between Five Models of Software Engineering. IJCSI International Journal of Computer Science, Vol. 7, Issue 5, September 2010



Authors

Engr. 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 Ph.D from Universiti Kebangsaan Malaysia. She is currently an Associate Professor at Universiti Teknologi PETRONAS, Malaysia. Her research interests include topics on Multi media, Human computer interaction, mathematics education, e-learning