

Human Factors and HCI Software Applications

Samer Barakat

Department of MIS, Applied Science Private University, Amman, Jordan

Abstract: Human factors influence the way people interact with computers and applications. It is related to the activities of computer users physically and mentally. There is a lack of research on the human computer interaction as a decisive and important factor in designing software applications. Systems analysts usually don't focus on HCI while designing the proposed system when in fact they are designing software applications based on user requirements. The SDLC method neglects to take human factors in consideration in the user interface design stage. Although, there is plenty of research on usability but the topic of human factor consideration is still a version domain. This research aims at investigating how to take the human factors in consideration when designing software application.

Key words: Human factors, systems analysis, software design, human computer interaction, SDLC methods

INTRODUCTION

It is a well known fact for software analysts that for an information system to succeed and be used by the end user it must be designed based on requirements solicited through the system user (Beck, 2000). According to the (chaos report 2015) almost 73% of software application developed ends up being scraped or trashed. The reason varies from software to software but most importantly they are discarded since they don't provide the users with the required functionalities expected from the system (Cockburn and Highsmith, 2001).

Moreover, users find difficulties in working with software systems that were poorly designed and does not take into consideration their human needs, although some of these systems are performing the required functionalities (Booch, 2006). Human factors are the specific details that a user needs to guarantee the ease of use and flexibility when using or working on the system (Laxmi and Sanavullah, 2007).

Convince and ease of use is a major factor that users look for in a software system (Dinh and Bieman, 2005). However, the needs of each user differs from other users which make it necessary for software analysts and designers to conduct a full study of the Human Factors involved in a software systems (Laxmi and Sanavullah, 2007). Some poorly designed systems that does not take the human factors into consideration may cause physical injuries to the users or make their jobs difficult to perform.

MATERIALS AND METHODS

Systems analysis methodologies: Software systems development needs to pass through a series of steps

during the System Development Life Cycle (SDLC). The main steps in a software development approach that are a common denominator among systems analysis methodologies are problem definition, planning, analysis, design and implementation and maintenance. SDLS is used to generate information systems that leads to business solutions.

No matter what methodology the systems analyst or developer use the methodology is an important tool to control resources, product design and quality assurance in order to produce "effective software systems" (Laxmi and Sanavullah, 2007).

There are several systems analysis methodologies ranging from the waterfall model to prototyping to extreme programming. Each has its own benefits and shortfalls. In this research we talk about all methodologies in general and how we need to incorporate the human factor into each and every methodology (Godfrey and Tu, 2000).

Case systems development life cycle: Among the different software development methodologies, the CASE system development life cycle is the closest to taking into account the needs of the user when designing the system. Use case design is a logical model of the proposed system, it provides the system requirements through use case diagrams that has an Actor at the center stage (Jacobson, 2007). The actor in the use case diagram represent the user of the system.

Actors are divided into two categories, main actors and supporting actors. The main actors provide data and or receive information from the system in order to provide details on what the system should be like. The Supporting actors provides assistance to keep the system running (Laxmi and Sanavullah, 2007).

Use case also provides the behavior of the user within the system. System scope defines the boundaries of the system.

Human factor: However, the SDLC does not clearly show that Human Factor or what is known as human computer interaction is part of the user interface design in a software system. It is left to the user interface designer to design the interface based on his judgment or previous experience. For software systems to work and be used it must be planned for and a series of steps must be followed to guarantee the success of the software project (Darroch, 2003).

A systems analyst plays a vital role in the design of software applications, he must be capable of working with all type of people and must understand their needs. Sometimes it is difficult for user to explicitly inform users with their needs therefore the systems analyst must observe the users behavior and drafts an outline of the user's needs to take into consideration when building the system (Barakat, 2012).

RESULTS AND DISCUSSION

When building a software system the analyst must understand the human information requirements and usability needs in terms of activities performed by users when working on the information system. Analysts use a number of requirements gathering tools including interviews, questionnaires, sampling data, observations and prototyping the proposed system (Cody *et al.*, 2002).

It is imperative for the analyst to learn how the users perform their tasks and how they interact with the information system. This shall help analysis build usable and useful information systems that would accomplish the needed business processes as well (Laxmi and Sanavullah, 2007). Flow charts and diagramming tools enable analyst to design the system but there must be new tools that can be incorporated in the design phase that take into consideration the human factor needs with regards to the proposed system.

Usability and ease of use may require the physical interaction between the user and the information system in the design phase. There must be a smooth transition from old systems to new systems with a better and enhanced usability and ease of use (Cody *et al.*, 2002). Although, training is important in any software project, users must be served in terms of providing software systems that are well designed in terms of functionality and ease of use. Therefore, we propose that analysts learn more about the incorporation of human computer interaction considerations when designing the software

system and by adopting the human centered approach right from the start leading to improving the quality of life for system users.

CONCLUSION

The purpose of this study was to investigating how to take the human factors in consideration when designing software application. Several software development methodologies were investigated. Our research focused on use case as the best development methodology that would take into consideration the human factors when designing an information system. It is our recommendation that all software development methodologies, systems analysts and software engineers must take into consideration human factors when analyzing and designing a new system. The researcher open the door for future research in this domain where by the HCI and the human factor domain is incorporated in the tools and methodologies used for the design of effective HCI software applications.

ACKNOWLEDGEMENTS

The research is grateful to Applied Science Private University, Amman, Jordan, for the full financial support granted to this research project (Grant No. DRGS-2015- 2016-62).

REFERENCES

- Barakat, S.M., 2012. Implementing an academic research social network: An exploratory study at applied science university. *Am. Acad. Scholarly Res. J.*, 4: 1-4.
- Beck, K., 2000. *Extreme Programming Explained: Embrace Change*. 2nd Edn., Addison Wesley, USA., ISBN-13: 9780201616415, pp: 190.
- Booch, G., 2006. *Object Oriented Analysis and Design with Application*. Pearson Education India, New Dehli, India, ISBN: 978-81-317-0232-1, Pages: 537.
- Cockburn, A. and J. Highsmith, 2001. Agile software development, The people factor. *Computer*, 34: 131-133.
- Cody, W.F., J.T. Kreulen, V. Krishna and W.S. Spangler, 2002. The integration of business intelligence and knowledge management. *IBM Syst. J.*, 41: 697-713.
- Darroch, J., 2003. Developing a measure of knowledge management behaviors and practices. *J. Knowl. Manage.*, 7: 41-54.
- Dinh, T.T.T. and J.M. Bieman, 2005. The FreeBSD project: A replication case study of open source development. *IEEE. Trans. Software Eng.*, 31: 481-494.

- Godfrey, M.W. and Q. Tu, 2000. Evolution in open source software: A case study. Proceedings of the International Conference on Software Maintenance, October 11-14, 2000, IEEE, Ontario, Canada, ISBN: 0-7695-0753-0, pp: 131-142.
- Jacobson, I., 2007. Object-Oriented Software Engineering. Addison-Wesley, USA.,.
- Laxmi, N. and M.Y. Sanavullah, 2007. Collaborative software architecture. Asian J. Inf. Technol., 6: 681-684.