

PROPOSED FRAMEWORKS FOR USER – DEVELOPER INTERACTIONS IN SOFTWARE DEVELOPMENT PROJECTS



ISSN: 2141 – 3290
www.wojast.com

EGBOKHARE, F. A. AND ONIBERE, E. A.

*Department of Computer Science,
University of Benin, Benin City. Nigeria.
E-mail: febokhare@yahoo.com*

ABSTRACT

Software development is highly human-centric and recent literature on software project management reveals high rates of failure in software projects. The causes of most failures are linked with improper management of the human-level interactions in software project. This paper developed frameworks for user-developer interactions in all phases of the software life cycle.

INTRODUCTION

Literature on software project management has reported numerous failures of information system projects (Keil *et al.*, 2002). IT report indicates that only 28 percent of software development projects are successful (Tesch *et al.*, 2007) while the others either failed or are challenged. Lack of user involvement in software projects has been identified as one of the prominent failure factors (Onibere and Egbokhare, 2007). In this paper, the term “developers” describe the various individuals directly involved with the investigation, design, coding and testing of software systems while “user” describes the end user, organization that requested for the software system or any stakeholders interested in the resulting software system. Leonard-Barton and Sinha (1993) observed that software project success depends not only upon the cost, quality, and compatibility of the technology, but largely on the processes of interaction between developers and users. Bob (2003) advised that developers who understand requirements in technical terms should interact directly with users throughout the software development project.

User-developer interactions afford the user not only the opportunity of being an involved partner in the software development process but also the opportunity to ask questions that relate to implementable features in the expected software system, the length of time to develop the software system, milestones and deliverables and the level of effort required to obtain the expected product. Barki and Hartwick (1994) observed in a certain study that clear statement of requirements, user involvement and participation contributed about 13.83% to software project success. Hillelsohn (1996) described a well-managed project as one in which users and developers are in constant communication. User involvement is a widely accepted principle especially in information system development projects that aim to produce usable systems (Kujala, 2003; Kujala *et al.*, 2005; Dittrich and Lindeberg, 2004) since users act as informants who can supply facts about their work (Olsson, 2004). Users may not be able to express what they want using the right vocabulary but to achieve project success, developers need to listen to user stories and transform the user’s task and domain knowledge into requirements. Liu *et al.*, (2011) identified two primary stakeholders (users and IS staff) that possess crucial, complementary skills and knowledge that can be utilized during the different phases of an Information system development project since successfully constructing systems require effective control of relationships among these stakeholders to elicit their contributions and cooperation. Dittrich and Lindeberg (2004) suggested that controlled cooperation between users and software developers be encouraged to improve usability and usefulness of the resulting

software system. Livari (2004) however observed that despite the wide acceptance of the need for user involvement in software development projects, user participation is often difficult and quite rare in software development organizations especially in the product development context. In a study of selected Software Development Organizations in Nigeria, (Onibere and Egbokhare, 2007) observed isolation of users from developers as one of the major causes of software project failures. Users and software developers are important contributors to software project success. The extent and type of communication between these two can greatly influence software project success and reduce the risks associated with IT/Software development projects. Zhiwei (2013) described user involvement in all the phases of the software lifecycle in software projects as a critical contributor to software project success. He further opined that lack of effective user involvement can cause incomplete definition of requirements. Users in most cases are the custodians of the knowledge of the actual problem that the software system seeks to solve. This paper therefore designs frameworks for user-developer interactions in software development projects in Nigeria. Petri Net structures (Peterson, 1977; Murata, 1989) were used as the building blocks to define the frameworks.

METHODOLOGY

This paper is based on findings in a study of one hundred and fifty software developers drawn from twenty software development organizations in Nigeria using (Onibere and Egbokhare, 2007). The study revealed that most software development contracts are initiated by the software project manager who in most cases is also the owner of the business. All contacts relating to the software work project are between the manager and the user organization to prevent the developers from taking undue advantage. This is one of the major causes of failures because the role of the developers is usurped from the project inception stage. Other failure factors obtained from the study include:

- (i). Lack of user involvement in the systems development process
- (ii). Lack of adoption of structured methodologies in software development projects
- (iii). Isolation of developers from users
- (iv). Faulty requirement engineering process leading to complex and ill-defined requirements
- (v). Lack of documentation of project information.

This paper used Petri-net structures to model points of user-developer interactions in a typical software development project. A Petri-net (Peterson 1977 and Murata, 1989,) is an abstract formal model of information flow. It consists of places, transitions and directed arcs. Arcs run between places and transitions, transitions and places but not between places and places or transitions and transitions. Desel and Juhás (2001), formally define a Petri Net as a triple (P, T, F) , where

- P is a set of places
- T is a finite set of transitions ($P \cap T = \emptyset$)
- F is a set of arcs known as a flow relation. It is subject to the constraint that no arc connects two places or two transitions, or more formally:

$$F \subseteq (P \times T) \cup (T \times P) \text{ is a set of arcs (flow relation)}$$

Since Petri Nets are an established tool for modeling and analyzing processes, user interactions, etc., Murata (1989) in this paper, marked Petri Nets are used to model the tasks involved in a typical software development process (marking denotes that a Petri Net is active. Resources are denoted with circles and processes/documents (transitions, which are active components that model activities that can occur) are represented with rectangles. Arcs connect resources with processes.

INTERACTIONS DURING REQUIREMENT DEFINITION

The requirement definition phase is one of the most sensitive and important stages of the software life cycle (Chakraborty *et al.*, 2012 and More *et al.*, 2011). Herlea (1997) described Software requirements definition as the process where the customers' needs in a software project are identified. This process is regarded as one of the most important and most difficult parts of building a software system because if the requirements are not right at this phase, it can cripple the resulting system (Keil *et al.*, 2002 and Brooks, 1987). To achieve success in the requirements definition phase, users and developers need to interact since users have the domain information while developers have the technical skills. Figure 1 depicts a framework for interactions during requirements definition. It involves a series of interactions between developers and users.

The user initially presents the requirements on the domain knowledge possessed. The developer takes the user's idea, applies technical skills and experience and refines this idea. This requires several meetings/levels of interactions between the users and developers. This requirement is then presented to the project manager by the developers as a basis for requirements negotiation.

INTERACTIONS DURING REQUIREMENT NEGOTIATION

Requirement negotiation is the process of resolving conflicts for the right balance of quality requirements. Boehm and Egyed (1998) described the requirement negotiation process as complex and difficult because of incompatibilities among stakeholders' interest and priorities, complex cost-quality requirements dependencies and an exponentially increasing resolution option space. The requirement negotiation process involves a round table meeting among the Project Manager, User Representatives and the Developers. This is shown in Figure 2.

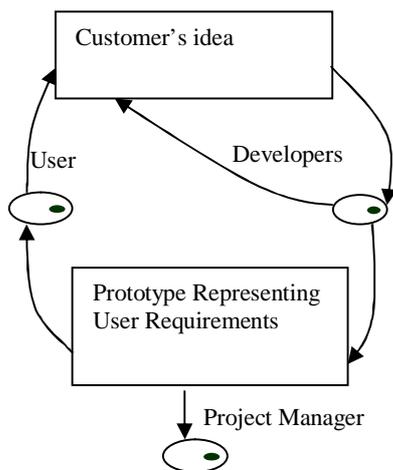


Figure 1: Users-Developers Interaction during Requirement Definition

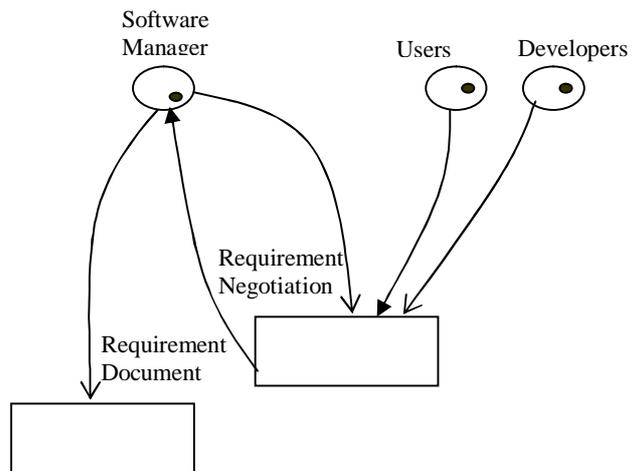


Figure 2: Requirement Negotiation

INTERACTIONS DURING THE DESIGN AND IMPLEMENTATION PHASE

Developers and users must collaborate in the design process since there is need for domain knowledge during the design process which the users can supply (Olsson, 2004). The users (or user representatives) are required at this stage to work directly with the developers and also to participate as the 'user interface' designers. Also, progress honesty is achieved since the developers are able to gain the user's trust through effective communication and regular interactions (Figure 3).

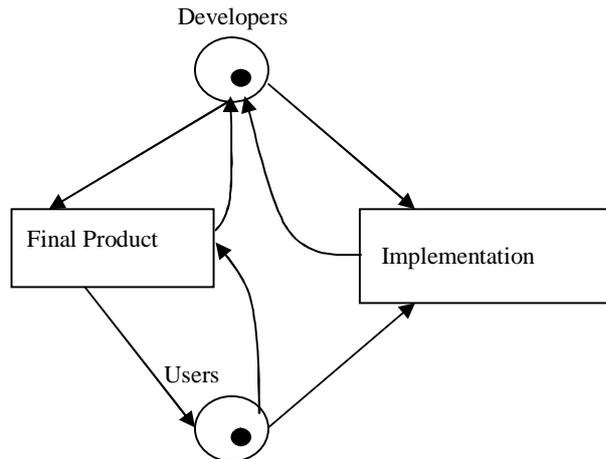


Figure 3: Interactions during the Design and Implementation Phase Development Process

DISCUSSION

The main direction of interactions between users and developers is important because user involvement and their willingness to understand and support the software development project positively contribute to success. There are specific roles for each actor involved in a software project. Problems arise when roles are swapped or neglected. In this paper, a roadmap to project success for Nigerian software managers has been defined using frameworks that depict human level interactions. To enable users understand the technicalities involved in the implementation of a software system, the software development organization need to create avenues for user participation.

CONCLUSION

It is widely accepted that users should be involved in information systems development but this has been almost practically a myth. This is because user involvement often requires a willingness to understand and support the project management process and a desire to positively contribute to the success and quality of the final system. The frameworks proposed in this paper will improve software development outcomes and further motivate software developers to develop systems built on well formed requirements in collaboration with involved users.

REFERENCES

- Barki, H and Hartwick, J. (1994). Measuring User Participation, User Involvement and User Attitude. *MIS Quarterly*, 18(1): 59-82.
- Bob, J. (2003) Have developers "forgotten their place" Retrieved November 24, 2006 from: http://weblogs.java.net/blog/editors/archives/2003/12/have_developers.html
- Boehm, B. and Egyed, A.(1998). Software Requirement Negotiation: Some Lessons Learned. *Proceedings of the 20th International Conference on Software Engineering*, pp. 503 – 506.
- Brooks, F.P. (1987). Essence and Accidents of Software Engineering, *IEEE Computer*, 20(4): 10-19.
- Chakraborty, A. Baowaly, M.K, Arefin, A and Bahar, A.N. (2012). The Role of Requirement Engineering in Software Development Life Cycle. *Journal of Emerging Trends in Computing and Information Sciences*. 3(5): 723-729.
- Constantine, L.L., Lockwood, L.A.D. (2002). Usage-Centered Engineering for Web Applications. *IEEE Software*. pp. 42–50.

- Desel, J and Juhás, G. (2001). *What Is a Petri Net? -- Informal Answers for the Informed Reader*, Eds, LNCS 2128, pp. 1-25.
- Dittrich, Y and Lindeberg, O. (2004). How use-oriented development can take place. *Information and Software Technology*, 46: 603–617.
- Herlea, D.E. (1997) Users' Involvement In The Requirements Engineering Process. *Knowledge Science Institute*, University of Calgary. Calgary, Alberta, Canada T2N 1N4.
- Hillelsohn, M.J. (1996). Been There, Done That: Some Lessons Learned on Software Development Contracts. Retrieved November 24, 2006 from: <http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1996/05/BeenTher.asp>
- Keil, M. and Carmel, E. (1995). Customer-Developer Links in Software Development, *Communications of ACM*, 38(5): 33-44.
- Keil, M, Tiwana, A. and Bush, A.(2002). Reconciling user and project manager perceptions of IT project risk: a Delphi study, *Information Systems Journal*,12(2):103–119.
- Kujala, S. (2003). User Involvement: A Review of The Benefits and Challenges. *Behaviour & Information Technology*. 22(1):1-16.
- Kujala, S., Kauppinen, M. Lehtola, M. and Kojo, T. (2005). The Role of User Involvement in Requirements Quality and Project Success. *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*
- Leonard-Barton, D. and Sinha, D.K. (1993). Developer-User Interaction and User Satisfaction in Internal Technology Transfer. *Academy of Management Journal*. 36(5):1125-1139.
- Liu, J.Y.C., Chiang, J.C., Yang, M. and Klein, G. (2011). Partnering effects on user-developer conflict and role ambiguity in information system projects. *Information and Software Technology*, 53:722–729.
- Livari, N. (2004). Enculturation of User Involvement in Software Development Organizations- An Interpretive Case Study in The Product Development Context. *Proceedings of the Third Nodic Conference Of Human-Computer Interaction*. pp.287-296.
- More, N.T., Sapre, B.S. and Chawan, P.M. (2011). An Insight into the Importance of Requirements Engineering. *International Journal of Internet Computing*. 1(2):34 - 36.
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of IEEE*, 77(4):541-580.
- Olsson, E. (2004). What active users and designers contribute in the design process, *Interacting with Computers*, 16:377–401.
- Onibere, E.A. and Egbokhare, F.A.(2007). Software Development Process: A Case of Nigerian Software Development Organizations. *The Information Technologist*. 4(1):1-13.
- Peterson, J.L. (1977). Petri Nets. *ACM Computing Surveys*, 9(3):223 – 253.
- Tesch, D, Kloppenborg, T.J and Erolick, M.N. (2007). It Project Risk Factors: The Project Management Professionals Perspective. *Journal of Computer Information Systems*, pp.61-70.
- Zhiwei, S. (2013). User Involvement in System Development Process. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering*. pp410 - 413.