Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 8 No. 1, March, 2017 - www.cisdijournal.net

Resolving the User-Developer Requirements Elicitation Conflict Using a Psychological Lens

Aziken, G.O. & Egbokhare, F.A.
Department of Computer Science,
Faculty of Physical Sciences
University of Benin
Benin City, Nigeria

***Corresponding Author*:** graceomikels@yahoo.com

ABSTRACT

Requirements elicitation though very critical activity in software development, has been plagued with challenges arising from poor user involvement in software development projects and difficulties in communication between software developers and users. Developed Software systems are unusable because users' actual needs are not captured. These needs are cognitive and formulated in the mind of users as mental models which are difficult to express in technical terms understood by developers leading to incomplete requirement definition. Using qualitative methods, this research explored certain psychological aspects of users' behavior to stimulate users to enable them formulate mental models along the line of the solution to their problem. Using Think Aloud Protocol Analysis techniques, users verbalizations were captured and analyzed to obtain users inner needs which ordinarily are difficult to express. These needs when incorporated into other functional and non-functional requirements will result in more complete software requirements and the development of usable software systems.

Keywords: User-Developer conflict, Think Aloud Protocol, Verbalizations, Mental Model, Cognitive psychology

CISDI Journal Reference Format
Aziken, G.O. and Egbokhare, F.A. (2017): Resolving the User-Developer Requirements Elicitation Conflict Using a Psychological Lens Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 8 No 1. Pp 15-20.
Available online at www.cisdijournal.net

1. BACKGROUND OF THE STUDY

The first phase in any software developmental effort is requirements engineering where the exact requirements of a software system are determined (Jiang et al., 2002; Verner et al., 2005; Sommerville, 2007; Pandey and Pandey, 2012). The user's needs which are sometimes imprecise and incomplete are translated into complete, precise and formal specifications in this phase (Nuseibeh and Easterbrook, 2000; Chakraborty et al., 2012). It is viewed as the most difficult phase in the software development process (Chakraborty et al., 2012; Swarnalatha et al., 2014) because users needs are established in this phase and errors relating to systems requirements undiscovered in this phase are expensive and difficult to fix in other phases of the software life cycle. Brooks (1987) described requirements engineering as the " hardest" single part of building a software system because it can cripple the resulting system if something goes wrong. Pressman (2001) opined that a sound requirements engineering process remains the best solution for the development of software systems that meet user needs and expectations. Without a well defined requirement, it may be difficult to make proper estimations of cost, time and scope of software development (Aggarwal and Singh, 2008). However, lack of user involvement and incomplete requirements occupy and retain a top position in the ranking of reasons for software failures (Kujala, 2008; Viskovic *et al*, 2008; Rasmussen *et al*, 2011, Standish CHAOS Report 2012). Requirements elicitation is the process of searching, revealing, acquiring and detailing of requirements for computer based systems (Coulin et al, 2005).

It is an important activity in the software development process that involves discovering, capturing, learning, determining, elaborating and gathering the needs of users and other stakeholders (Mulla and Girase, 2012; Sharmila and Umarani, 2011; Zarinah and Siti-Salwa, 2009). Requirements elicitation is a major aspect of requirements engineering where user involvement is critical. However, a persistent problem of software engineering is that users know what they want in a software system but they have difficulties expressing these needs. This has resulted in poor communication and eventual isolation of users from the development process, leading to developers having a poor knowledge of users and their actual needs (Walid and Happel, 2009; Nicolas et al., 2008). The requirements elicitation process is a collaborative social practice involving multiple stakeholders that must interact, communicate and eventually agree on the requirements since people perceive information differently and every individual conceives their own representation of the real world based on their experience and working environment.

Fuentes- Fernandez et al, (2010) opined that understanding the human context within which a system will operate is fundamental for the requirements of that system. Developers may not have the requisite skills to successfully elicit the needs of users based on only the technical knowledge they possess. This study therefore proposed certain psychological factors necessary to deal with various human aspects to gather accurate requirements in addition to technical skills.

2. METHODOLOGY

To determine some psychological factors that influence users behavior, formal interviews were conducted. The aim of the interview was to understand the factors that influence users to think along the line of the solution to their problem or formulate mental models which represent their exact needs. Since most times this mental models are formulated in the minds of the users (cognitive), the study also sought cognitive psychology tools used to extract users mental models, as this will help the developer capture those needs that users often find difficult to express.

3. KEY INFORMANT INTERVIEWS

The key informant interviews were used to elicit information related to what mental models are, how they are formulated and how mental models can be extracted. The key informants consisting of five (5) Educational Psychologists from the Faculty of Education, University of Benin and two (2) Clinical Psychologists from the University of Benin Teaching Hospital, both in Edo State, Nigeria were interviewed. A typical interview session lasted for about 30 minutes. All the psychologists were asked the same questions so as to ensure consistency in feedback/response. Each session of interview commenced with a brief introduction of the purpose of the study. Permission was thereafter sought to get their responses on the topic of discussion and to record the interview. Two of the Interview sessions were recorded (one with an educational psychologist and one with a clinical Psychologist) with the permission of the interviewees and the responses were later transcribed.

4. FINDINGS /DISCUSSION

The interview reports were transcribed and keywords related to the subject of interest were underlined and extracted. It was deduced from the interview that mental models are internal representations of an individual's perception of a particular situation. Though they are difficult to represent especially by people not in the domain of reference (e.g. a user in a software development environment), they cannot be ignored. Mental models are constructed and simulated within a conscious mind. Mental models can also be described as psychological representations of real, hypothetical or imaginary situations with a form that corresponds to the structure of what they represent (i.e. a person's understanding of the surrounding world). They are based on incomplete facts, past experiences, and even intuitive perceptions hence they can be seen as a set of assumptions which help shape actions and behaviour. They expose what people pay attention to in complicated situations, and define how people approach and solve problems.

4.1 Factors that Stimulate the Formation Of Mental Models

An individual's thought towards a phenomenon is stimulated and directed by experience, training and instruction. (Figure 1).

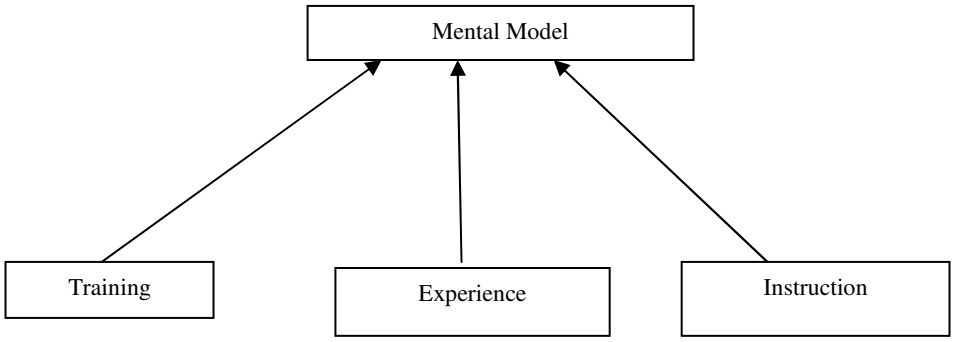

Figure 1: Formation of Mental Models

When users of software are exposed to certain stimulus (e.g a prototype), it helps them formulate mental models about the goal to be achieved, the actions required and how these actions are translated into the desired software system which represents what the perceived outcome should be.

4.2 Perception

Perception is a process by which individuals organize and interpret their sensory impressions in order to give meaning to their environment. People's behaviour are based on their perception and the world that is perceived is the one that is behaviourally important (selective attention). Selective distortion is perception in a way that is along what the user already knows. It affects belief system. Other interrelated factors that influence a user's perception and the formation of mental models are:

i. Belief system
ii. Environment
iii. Memory.
iv. Sensory impressions
v. Motives
vi. Experiences
vii. Expectation
viii. Interest
ix. Learning
x. Motivation
xi. Ability to think logically and coherently.
xii.

Findings from the interview helped to understand the gap between users and software developers. While the user thinks along the line of the application domain, the developer perceives the problem in technical terms, thus creating a gap and placing both at two extremes of a continuum as illustrated in Figure 2

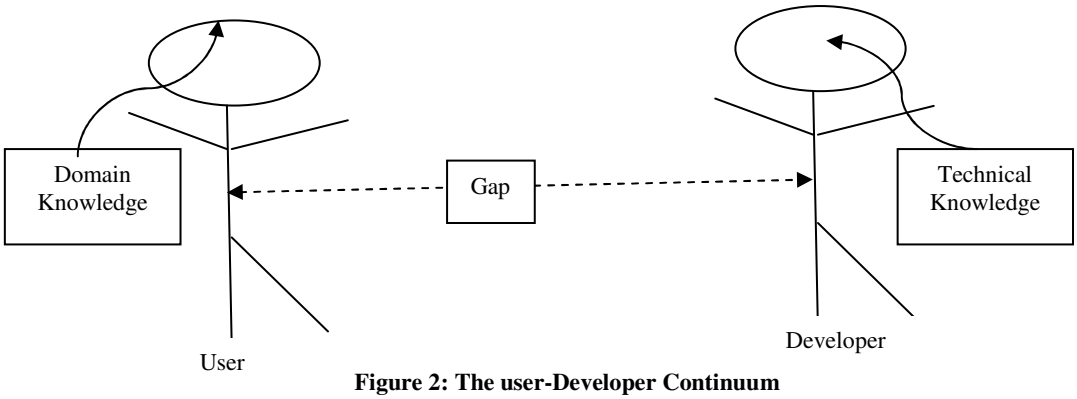

Figure 2: The user-Developer Continuum

The developer therefore needs to empathize with the users through close contact to appreciate the users difficulty in expressing these needs. To bridge the gap between software developers and users and ensure that they both think along the same continuum to find a solution to the users' problem, the developer needs to stimulate the user through experiments that prompt them to think and formulate mental models in line with the developers conceptual model as presented in figure 3.

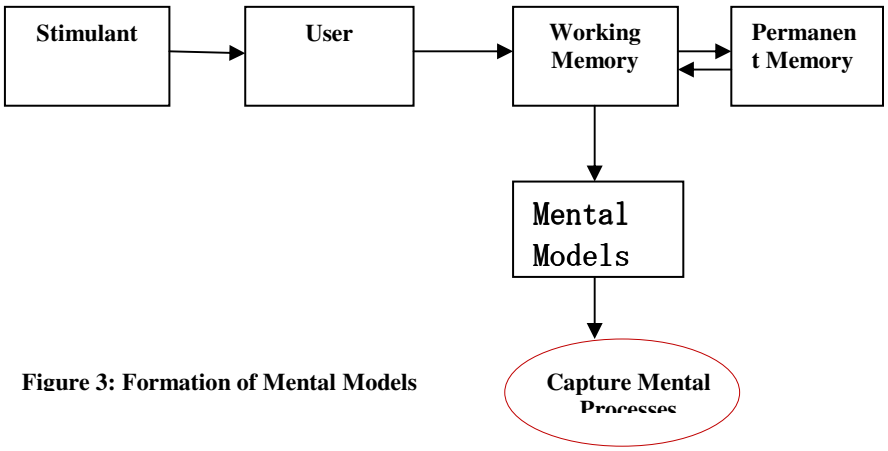

Figure 3: Formation of Mental Models

When a user is stimulated along the line of the solution to the problem, domain knowledge (stored in permanent memory) is combined with the new experience to formulate visualizations (mental models) in the working memory. These visualizations represent the images and expressions of the users needs. If they are captured while still in working memory, verbalizations representing the users implicit needs are elicited. This becomes more difficult if these mental visualizations get into the permanent memory. The next critical question is how to capture the users mental models?

4.3 Capturing Users Mental Models

Several techniques exist for capturing metal models such as Card sorting, concept maps, repertory grids, cognitive maps and Think aloud protocol. A literature search showed that think aloud protocol is one of the widely used methods by psychologists to understand people's mental processes hence this study proposed it to capture mental models.

4.4 Think Aloud Protocol (TAP)

Think Aloud protocol is a verbalization method designed by Psychologist Karl Duncker in 1945 as a way of understanding people's mental processes (Johnstone et al., 2006). Ericsson and Simon (1993) posited that Think Aloud Protocol is a valid method for researching cognitive processes and it uncovers thought processes and reveals the content of working memory (Sahebkheir and Davatgari, 2013). By prompting the participants to 'speak out' what goes through their mind, it is believed that users mental models can be captured and analyzed. Wilhelm (2006) opined that think alouds gives a glance into hidden activities, thereby making it possible to understand what is taking happening below the surface of consciousness. Whitehead *et al* (2015) agreed that Think Aloud protocol be applied to record cognitive processes while the task is being performed as was suggested in previous studies by Ericsson and Simon, (1980; 1993). It has been used to develop an understanding of the cognition of individuals and differences in decision making processes of various individuals (Calmeiro and Tenenbaum, 2011; Arsal, 2013). According to Kobrin and Young (2003), it is considered by some as the most direct way of uncovering the psychological processes that an individual uses to carry out a task and they provide a plethora of information than would otherwise be accessed about cognitive processes (Hayes and Flower, 1980). Think Aloud Protocols have also been applied in Usability testing and Human Computer Information (Blair and Brick, 2010).

5. EXPERIMENTS TO CAPTURE USER'S MENTAL MODELS

An experiment was conducted using Think Aloud protocol to capture users requirements. The users were undergraduate students who usually throng the Students Complain Section (SCS) of the Information Technology Department to lodge complaints and seek solutions to problems arising from the use of the University portal. A large number of students daily throng the SCS, leading to chaos and difficulty in attending to students. To avoid this situation and minimize physical contact, the unit proposed a help desk system to handle students complaints online. The researcher worked with the Requirements Definition team for the Help Desk System to capture users mental models using TAPs. An initial prototype (stimulus) was developed based on the documented complains. The aim was to use the prototype to stimulate the users along the line of the solution to their problem so that their mental models can be formed along the line of the solution that will meet their needs. Four (4) participants were randomly selected for five (5) days to participate in the experiment as suggested by Young (2005) that a maximum sample size of six (6) participants is most effective for a think aloud experiment.

The following steps were followed to capture the users mental models:

i. Train participants. The prototype was presented to the students and they were asked to use it. There is a standard think –aloud instruction extracted from Ericsson and Simon (1993) in Austin and Delaney (1998) which was modified .

6. MODIFIED STANDARD THINK-ALOUD INSTRUCTION

"In this experiment, we are interested in what you think about this system. In order to do this I am going to ask you to THINK ALOUD as you use the system. What I mean by think aloud is that I want you to tell me EVERYTHING you are thinking as you use the system. I would like you to talk aloud CONSTANTLY. I don't want you to try to plan out what you say or try to explain to me what you are saying. Just act as if you are alone in the room speaking to yourself. It is most important that you keep talking.

If you are silent for any long period of time I will ask you to talk. Do you understand what I want you to do?"

i Conduct experiment and elicit verbalizations. A recorder was placed by each participants and the experiment was conducted on one participant at a time. Though Ericsson and Simon (1993) argued that the researcher and the device should be 'out of sight' during the recording session to enhance reliability of the data, the researcher discovered that the participants suddenly became quiet when stuck and this was the core of the experiment; getting those mental models hence the participants were instructed to verbalize the thoughts that came to their minds as they performed the experiment (Ruso *et al.*, 1989). They were prompted from time to time to speak out their mental models when they became quiet so as to maximize their verbalizations. Each experiment lasted for an average of 12 minutes.

ii. Transcribe the Recording. Each recording was transcribed immediately after the session. It is also a good idea to do the transcribing shortly after the recording session. This way some of the parts otherwise missing due to some background noise, unintelligible speech or too low voice will not get lost.
iii. Extract keywords and phrases relating to system requirements from TAP output. It's been observed that analyzing think aloud data can be overtly subjective. A common practice therefore is to review verbatim interview transcripts to identify word /strings that indicate a particular type of requirement (Blair and Brick, 2010). Also, the granularity of a phrase in relation to the requirements is considered before extraction.

The output of the TAPs revealed some cognitive requirements that improved the quality of the requirements for the help desk system. For example, the need to profile users and extract requirements from each user group became clearer. Also other functional requirements were extracted from the TAPs.

7. CONCLUSION

Requirements elicitation is a complex process since it is difficult for users to express their exact needs and developers view the users problem from a technological perspective. This had resulted in incomplete requirements and the development of software systems that are unusable. This research studied the role of Cognitive psychology in thinking, feeling and behavior of users and developers in resolving the user-developer conflict during requirements elicitation. Adopting certain psychological processes to stimulate users to think along the line of the solution to their problem and that when they encounter difficulties, they begin to visualize in working memory, thereby producing mental models. Users thoughts were captured using verbalization techniques and keywords relating to their inner needs were extracted and formulated into more complete software requirements. This technique can be adopted and combined with any other existing techniques to reduce software failure rates.

REFERENCES

1. Aggarwal, K. K. and Singh, Y. (2008). Software Engineering. New Age International Publishers, 528P
2. Arsal, G. (2013). Investigating skilled and less skilled Golfers psychological preparation strategies: The use of a think – aloud cognitive Process – tracing measure, Ph. D thesis, University of Florida
3. Blair, J. and Brick, P.D. (2010). Methods for the Analysis of Cognitive Interviews. Section on Survey Research Methods – JSM , 3739 - 3748
4. Brooks F.P. (1987). No silver Bullet: Essence and Accidents of Software Engineering. IEEE Computer 20(4), 10 – 19
5. Calmeiro and Tenenbaum G. (2011). Concurrent verbal protocol analysis in sport: illustration of thought process during a golf putting task. Journal of clinical sport psychology 5, 223-236
6. 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.
7. Coulin C., Sahraoui, Kader, A and Zowghi D. (2005). Towards a collaborative and combination approach to requirements elicitation within a system engineering framework. 18th int conference on system engineering, 456-461
8. Ericsson, K.A. and Simon, H.A. (1980). Verbal reports as data. Psychological Review, 87(3), 215-251.
9. Ericsson, K. A., and Simon, H. A. (1993). Protocol analysis: Verbal reports as data(Revised edition). Cambridge, MA: MIT Press.
10. Fuentes- Fenandez, R., Gomez- Sanz, J., and Pavon J. (2010). Understanding the human context in requirements elicitation. Requirements Engineering 15(3). 267- 283
11. Hayes J. and Flower L. (1980). Identifying the organization of writing processes. In Gregg L. W., Steinberg E. R. (Eds.), Cognitive processes in writing. Hillsdale, NJ: Erlbaum. 3 - 29
12. Jiang, J.J., Klein, G. Chen, H. and Lin, L. (2002). Reducing user-related risks during and prior to system development. International Journal of Project Management 20, 507 - 515
13. Johnstone, C. J., Bottsford-Miller, N. A., and Thompson, S. J. (2006). Using the think aloud method (cognitive labs) to evaluate test design for students with disabilities and English language learners. Technical Report 44. Minneapolis, MN: University of Minnesota, National Center on Educational Outcomes.
14. *Duncker, K. (1945). On Problem Solving. Psychological Monographs, 58, 1-113.*
15. *Kobrin, J.L. and Young, J.W. (2003). The Cognitive Equivalence of Reading Comprehension Test Items Via Computerized and Paper-and-Pencil Administration. Applied Measurement in Education.16(2), 115 - 140*
16. Kujala, S. (2008). Effective user involvement in product development by improving the analysis of user needs. Behaviour and Information Technology, 27(6), 457- 473.
17. Mulla, N. and Girase, S. (2012) A New Approach To Requirement Elicitation
18. Based On Stakeholder Recommendation And Collaborative Filtering. International Journal of Software Engineering & Applications.3(3), 51 - 60.

19. Nicolas, B., Rahul, P., Thomas Z. and Sunghun, K. (2008). Duplicate Bug reports considered harmful ……. Really? In Proceedings of the 24th IEEE International conference on Software Maintenance (ICSM) Beijing, China
20. Nuseibeh B. and Easterbrook S. (2000). Requirements Engineering: a roadmap. Proceedings of the Conference on The Future of Software Engineering. 35-46
21. Pandey, D. and Pandey , V. (2012). Importance of Requirements Management. A Requirement Engineering Concern. International Journal of Research and Development -A Management Review (IJRDMR) 1(1), 66 – 70
22. Pressman R.S. (2001). Software Engineering: A Practitioner's Approach, 5th Edition. McGraw Hill Higher Education. New York. 855
23. Rasmussen, R., Christensen, A. S., Fjeldsted, T., & Hertzum, M. (2011). Selecting users for participation in IT projects: Trading a representative sample for advocates and champions? Interacting with Computers, 176-187.
24. Ruso, J., Johnson, E. and Stephens, D. (1989). The validity of verbal protocols. Memory and Cognition. 17(6), 759 - 769
25. Sahebkheir, F. and Davatgari , H. (2013). The Role Of Think-Aloud Protocols On Developing Iranian Efl Learners' Written Performance. Indian Journal of fundamental and Applied Life Sciences ISSN: 2231– 6345. . 4 (3), 1-7
26. Sharmila, P. and Umarani, R. (2011). A Walkthrough of Requirements Elicitation Techniques. International Journal of Engineering Research and Applications. 1(4), 1583 – 1586
27. Sommerville, I. (2007). Software Engineering. 8th Edition. Pearson Education Limited. Essex, England. 840P
28. Standish Group. (2012). CHAOS Manifesto. Available at https://cs.calvin.edu/courses/cs/262/kvlinden/resources/CHAOSManifesto2012.pdf 16/01/2017
29. Swarnalatha, K.S., Srinivasan, G.N., Meghana, D., Raunak, K. and Kopal S. (2014). A Survey on Software Requirement Engineering for Real Time Projects based on Customer Requirement. International Journal of Advanced Research in Computer and Communication Engineering. 3(1), 5045 – 5050.
30. Verner,, J., Cox, K., Bleistein, S., and Cerpa, N. (2005). Requirements Engineering And Software Project Success: An Industrial Survey In Australia And The U.S. AJIS. 13(1), 225 – 238
31. ***Viskovic, D, Varga, M, & Curko, K. (2008). Bad practices in complex IT projects. International Conference On Information Technology Interfaces, 301***
32. Walid, M. and Happel, H. (2009). From work to word: How do software developers describe their work? In 6th IEEE International Working Conference on Mining Software Repositories
33. Wilhelm, J. D. (2006). A study Guide for Improving Comprehension With Think-Aloud Strategies. Schlastocs Publications. Retrieved 16/12/2016 http://www.teachervision.fen.com/go/http/fen.com/resour ces/copyright.html
34. Whitehead, A. E., Taylor, J.A., Remco, C.J and Polman (2015). Examination of the Suitability of collecting in event cognitive processes using Think Aloud Protocol in golf. Frontiers in Psychology. 6 (1038). 1 - 12
35. Young, K. A. (2005). Direct from the source: the value of 'think-aloud' data in understanding learning. Journal of Educational Inquiry, 6(1), 19-33.
36. Zarinah M. K. and Siti-Salwah S. (2009). Supporting Collaborative Requirements Elicitation Using Focus Group Discussion Technique. International Journal of Software Engineering and Its Applications. 3(3), 59-70.