Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 111 (2017) 48687-48692

ISSN: 2229-712X

Iterative Software Process Based Collaboration Model for Software Stakeholders

Muhammad Ahmad Shehu, Haruna Abdu and Ovye John Abari
Assistant Lecturer, Computer Science Department, Federal University Lokoja, Nigeria.

ARTICLE INFO

Article history:
Received: 4 August 2017;
Received in revised form:
30 September 2017;
Accepted: 10 October 2017;

Keywords
Software engineering,
Software Development Complexities,
Software Project Failure,
Software Process Models,
Software Stakeholders' Collaboration,
Iterative Software Process Model.

ABSTRACT
Software engineering is well known for its significance on software development complexities minimization. Due to software engineering significance researches were carried out to improve software engineering practices. However, there are some identified problems that lead to software development complexities like the lack of understandable collaboration or communication between software stakeholders during software development. And to this identified problem a research was carried out to solve this problem by proposing a collaboration model for software stakeholders' collaboration during software development. However, the model proposed was restricted to a waterfall software process model. This study used segment of the framework used for developing the waterfall software process based collaboration model to develop an iterative software process based collaboration model for software stakeholders during software development. And this study proposed model will help minimize the problem of lack of understandable collaboration between software stakeholders during software development

Introduction

The demands for software to replace manual system of solving problems in numerous domains are increasing due to the effective (desired results productions) and efficient (speedy or timely result production) significance of software. Using computerized tools during automation of problem (like mathematical problem, record management problem and so on) resolution processes defines software development. More significance is given to automated method when Comparing automated method of (software development) solving problems and traditional method (manual process) of solving problems due to the efficient and effective significance of automated method of solving method [1].

With the significance of software, there are still complexities experienced by software. The measure of difficulty experienced by software during its development state or during its execution state is referred to as software complexities [2]. Software complexity during the software development state is experienced by the software developers or software development stakeholders while the software complexities during the software execution state (state of solving the problems) is experienced by the software. However, the software complexity during software development state is considered in this paper. The effect of software development complexities are directed towards the failure of the software purpose (software project failure) [3]. The higher the complexities the higher the failure rate of software.

Due to the effects of software development complexities, software engineering was introduced. Software engineering involves software development process guided by acceptable and well-defined principles. The principles of software engineering are well described in Larman [4]. Among various practices of software engineering, the use of software development process models during software development is the common practice and this practice is the consideration in this paper.

Research contributions in software engineering have been carried out to improve the practices of software engineering due to the complexities reduction significance of adopting the practices of software engineering during software development. To improve software development models Balsamo et al [5] carried out a research on how model-based performance can be embedded in the early stages of software development processes and their purpose for the research was that the commonly used software development process models (reviewed in their study) for software development have less consideration on software performance in the early stages and causing major software performance setbacks on the software developed.

To identify the most suitable software development process model for software development of various systems, Kour [6], Munassar, and Govardhan [7] and Maheshwari and Ch. Jain [8] did a comparative analysis of different types of software development process models. Kour [6] carried out an analysis on various software development process models to identify a generalized and suitable software process (which was called Software Development Life Cycle (SDLC) in the study) models for companies software development. Munassar, and Govardhan [7] and Maheshwari and Ch. Jain [8] did analysis of 5 and 4 types of software development process models respectively. A conclusion was met in Kour [6], Munassar, and Govardhan [7] and Maheshwari and Ch. Jain [8] analysis that each software process model has its

Tele: +2348108984117
E-mail addresses: sdfax2014@gmail.com

usage merits and demerits and each model usage suitability in domains depends on the condition (size, requirement complexities and so on) of the domain.

Software project fails due to various reasons (software development complexities) [3], to identify these reasons Kaur and Sengupta [3] did a review on software development process models and the rate of software project failure. Among other various important reasons why software project fails identified by Kaur and Sengupta [3], the lack of understandable collaboration or communication between experts within the team involved in the software development and the lack of user requirement was a base for Muhammad, Haruna and Rufai [1] research. However, it was observed in Kaur and Sengupta [3] research that the strength of understandable collaboration within the team of software development experts depends on the size of the team, that is the smaller the team size the stronger the understandable collaboration.

Muhammad, Haruna and Rufai [1] contributed to software development complexities minimization by analyzing the reason why software project fails identified by Kaur and Sengupta [3] (that is the lack of understandable collaboration or communication between software development experts and lack of user involvement during software development). To help minimize software project failure complexity, Muhammad, Haruna and Rufai [1] developed a software stakeholders (software development experts and users) collaboration model for software stakeholders' collaboration during software development. However, the collaboration model usage was restricted to software development of small scaled systems.

However, based on the limitation specified by Muhammad, Haruna and Rufai [1], that their proposed model usage restriction was within the boundary of waterfall software process model (used for software development of small scaled system), this study used the study framework of Muhammad, Haruna and Rufai [1] to develop a collaboration model for software stakeholders' collaboration during software development using iterative software process model.

Study Concepts and Methods

(i) Stakeholders and their collaboration

The term 'Stakeholders' has various descriptions by various research contributors in software engineering. Individuals having a defined stake or interest in software project is a description of Stakeholders described by Cotterell and Hughes [9], Sharp, Finkelstein & Galal [10]. This description was used in Muhammad, Haruna and Rufai [1] which serve as base of this study.

The stakeholders identified in the study of Muhammad, Haruna and Rufai [1] were categorized as Users and Developers, where users are described as the stakeholders that are existing system (system that need to be automated) experts and users, and the direct users of the proposed system (software). And developers are the experts (called system analyst) in analyzing the existing system based on the ideas of the users, experts (called software analyst) in analyzing the proposed system based on the results of existing system analysis, and experts (programmer) in implementing the result of the proposed system analysis. Software stakeholders' collaboration is a structural process of how stakeholders communicate or interacts during software development.

(ii) Software Development Process Models

Based on software engineering practices it is important to use software development process model for software development due to its guidance and development principles it provides to minimize software development complexities. Software development process model is an abstract or graphical definition of the architectural and design process of software development [11]. There are various software development process models used for software development, and each with its problem domain applicability, advantages, disadvantages, and contingencies. However, in this study the iterative software development process model was used for developing of the proposed model.

a) Iterative Software Development Process model

Iterative software development process model is an improvement of the traditional waterfall software process model due to the setbacks experienced by the waterfall software process model [7][8]. In iterative model the whole software project is divided into segments with the current project segment depending on the initial project segment. Each project segment is designed using waterfall process model. The software project division into segments depends on the size of the software project and/or the project segments conclusion of the software project stakeholders (software developers). The advantage of iterative model over waterfall model is the fact that results (software) are developed earlier for user feedback during the software project development. Each current iteration result is submitted as incremental software of the initial iteration result [8].

Muhammad, Haruna, and Rufai [1], research work was the base for this study and Waterfall Software process model was the consideration of their work. However, the proposed collaboration model limitation of Muhammad, Haruna and Rufai [1] falls within the boundry of waterfall software process model limitation [1].

Since Iterative software development process model improves the limitation of waterfall software development process model, this study considered Iterative software process model during development of the proposed collaboration model to improve the limitation of Muhammad, Haruna and Rufai [1] proposed collaboration model.

(iii) Flowchart Representation

This is a symbolic and unified representation of problem resolution process [12]. Flowchart is broadly acceptable in various problem domains due to its descriptive and understandable nature.

Since software stakeholders' collaboration deals with interactive processes to solve problem then in this study, flowchart representation was used for describing the interaction between stakeholders.

Study Framework

Based on the research framework of Muhammad, Haruna and Rufai [1] in Fig. 1, a process to identify the software development process model for small scaled system software development was carried out and of which waterfall software process model was chosen. However in this study, a software process model had been chosen to minimize the setback of the waterfall software process model chosen by Muhammad, Haruna and Rufai [1] during the development of their proposed model, therefore the process of small scaled system based software process model identification was skipped in this study's framework. This study framework is shown in Fig. 2.

The research framework used in the development of this study proposed model begins with describing each process carried out in an iterative software process model to enable

the identification of expert or experts (software stakeholders) of each process described.

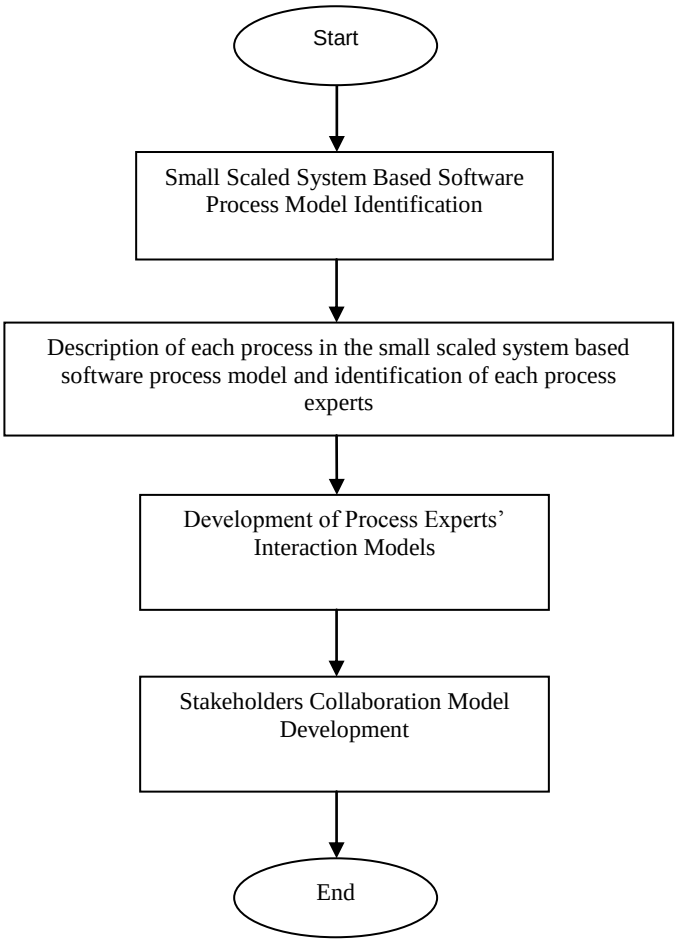

Figure 1. Muhammad, Haruna and Rufai. .Framework

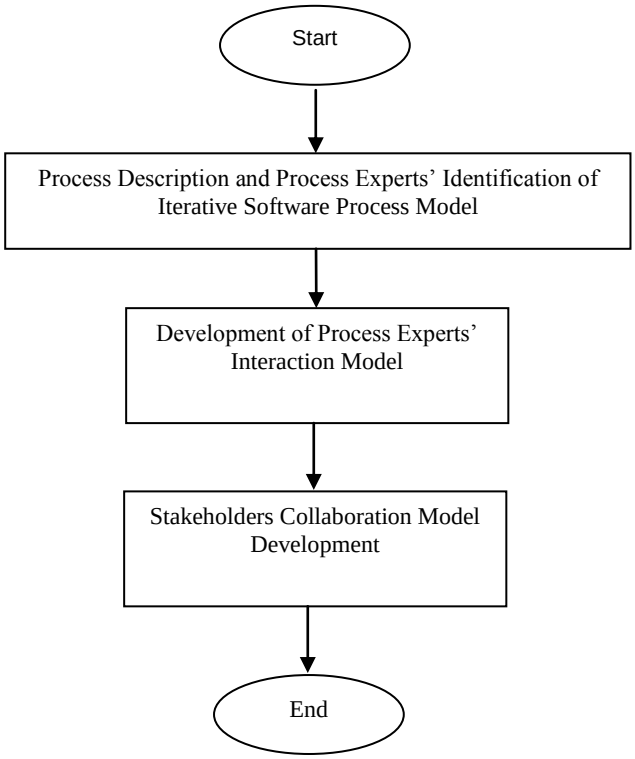

Figure 2. Study Framework.

The identification of each process expert or experts is an essential step due to the fact that the proposed collaboration model is to be used by software stakeholders. Apart from identification of experts in each process of iterative software process model, the process description step of this study framework also helps to understand the links (interaction) between the identified experts (software stakeholders) within each process. Based on the understanding of the links (interactions) between the experts (software stakeholders), the next step (Development of Process Experts' Interaction Model) of this study framework is enabled.

Using each of the process experts links (interactions) understood from this study framework step of process description of an iterative software process model, an Experts' Interaction Model was developed for each process of an iterative software process model. A flowchart representation was used to represent the experts' interaction.

To develop the Software Stakeholders Collaboration Model, each process representation in an iterative software process model was replaced with their respective experts' interaction model developed in the initial step (Process description and process experts' identification of Iterative software process model) of this study framework.

Study Results and Discussions

(i) Process Description and Process Experts' Identification of Iterative Software Process Model

Each software project segment development process (iteration) in an Iterative Software Process Model is carried out using waterfall process model [7][8] and therefore the processes involved in an Iterative Software Process Model are the processes in a waterfall software process model, that is Requirement analysis process, Design Process, Implementation Process, Verification Process and Mentainanace Process. Fig 3 and fig 4 are abstractions of a Waterfall and Iterative Software Process Models respectively.

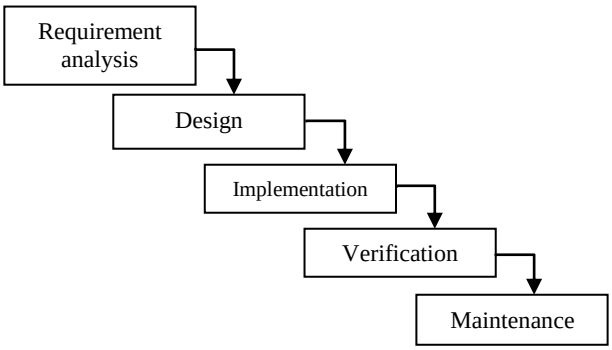

Figure 3. Waterfall Software Process

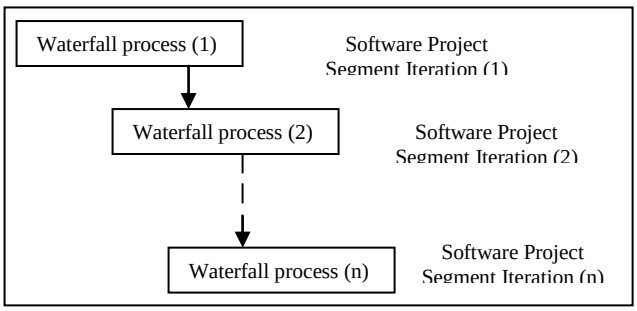

Figure 4. Iterative Software Process Model.

a) Requirement Analysis Process and its Experts

In simple term, requirement analysis involves the collection of needed requirements to be met by the proposed software. Requirement analysis process or stage is called Inception phase in some literatures like [4]. In iterative software process model, feedbacks (requirements) from users during each software project segment iteration are always achieved in the requirement analysis phase and it determines how the subsequent processes are carried out or achieved. Requirement analysis are categorized into two namely; system requirement analysis and software requirement analysis.

System requirement analysis involves observation of the existing system and interaction with existing system experts (users) for collection of the necessary components that makes up the existing system and how the components interact to achieve results. Collection of necessary requirements is carried out through software engineering acceptable textual documentation (like Event list, Business Use Case Scenario and so on) and graphical or pictorial modeling (Business Use Case, Context Diagram and so on). It is based on the collected requirements of the existing system that the software requirement is analyzed. However before software requirement analysis is carried out, the analysis of the existing system must be confirmed or acknowledged by the existing system experts (user).

Software requirement analysis involves using the collected system requirements (components) to structure the components of the proposed software. During software requirement analysis the following are carried out: (a) Structure the components of the proposed software using the software engineering defined textual documentation (like System Use Case Scenario) and graphical documentation (like Design model). (b) Determine if the proposed software development can be achieved or not. Also the cost (funds, efforts, etc) of carrying out the project is determined. Based on the cost of project determined the user and the analysts also determines if the project is worth building. This implies that software requirement analysis result has to be viewed and acknowledged by the system analyst and user.

Based on the description of the requirement analysis phase, it is observed that there is system requirement analysis process which is handled by the system analyst with the help of the existing system expert (user) and partial process of software requirement analysis which is handled by the software analyst. This implies that in the phase or stage of requirement analysis, three software stakeholders are involved (user, system analyst and software analyst).

b) Design Process and its Experts

The design process is also referred to as the software architectural design or elaboration phase [4]. This stage involves the further elaboration of the software requirement analysis. The proposed software components identified in the requirement analysis phase are detailed out by analyzing the interactions between the components. Since the proposed software component are structured with respect to the existing system components analyzed in the requirement analysis phase, then the interaction between the proposed software components are also structured with respect to the interaction of the existing system components.

In this stage, the general framework of the proposed software is structured to meet the existing system requirements analyzed in the requirement analysis phase. Each of the proposed software components are detailed by decomposition into sub components with specification of interaction between the sub components, this process is referred to as detailed design, and this will enable the specification of how each of the proposed software components are implemented to be determined. Software engineering modeling tools (like Design class diagram, Collaboration diagram, system sequence diagram etc) are usually used for the documentation of the design process stage. However before the implementation of the design stage process result, the design stage process results must be confirmed or acknowledged by the system analyst based on the existing system requirements.

Based on the description of the design stage process, it is observed that the concentration of analysis is on the architectural design of the software. This implies that the software stakeholders involved in this stage are software analyst for the architectural design and the system analyst for the confirmation of the design.

c) Implementation Process and its Experts

The design process result is submitted for implementation by the software analysts. The implementation stage involves the process of implementing the result of the design stage using specific programming package (like Java, VB.Net, PHP etc). The software stakeholders involved in this stage are the programmers (for implementation) and software analyst (for acknowledgement of the implementation results).

d) Verification Process and its Experts

This process involves determining whether the software developed meets the system requirements and resolving any errors generated during implementation. Since this stage involves testing the software developed by the programmer based on the existing system requirements analyzed by the system analyst with the directions or idea of the existing system expert (user), then the software stakeholders involved in this stage are programmer (developed software), system analyst (analyze existing system to guide the proposed software development) and user (guide the system analyst based on what user need during existing system analysis).

The verification process marks the end of developing the software if the software meets the system requirements. The next stage involves future requests from the user.

e) Maintenance Process and its Experts

The maintenance process involves addressing future problems and user's enhancement requests after the initial software is released. Since this process involves enhancing the initial software then the same set of software stakeholders involved during developing the initial software is involved in the maintenance process.

However, in iterative software process model the maintenance process represents the iterations due to the fact that the current software project segment execution depends on the feedbacks (enhancement/requirement) achieved in the initial software project segment execution. Therefore, maintenance process is modeled using iteration representation during the development of this study proposed model.**(ii) Development of Process Experts' Interaction Models**

Every process in an iterative software process model has its expert or experts (stakeholders) and in order to achieve the desirable result of each process the experts must interact. In this stage of this study framework, an experts' interaction model was developed (as in fig5, fig6, fig7 and fig8) to describe the interactions between experts of each process of an iterative software process model and flowchart representations were used to model the experts' interactions.

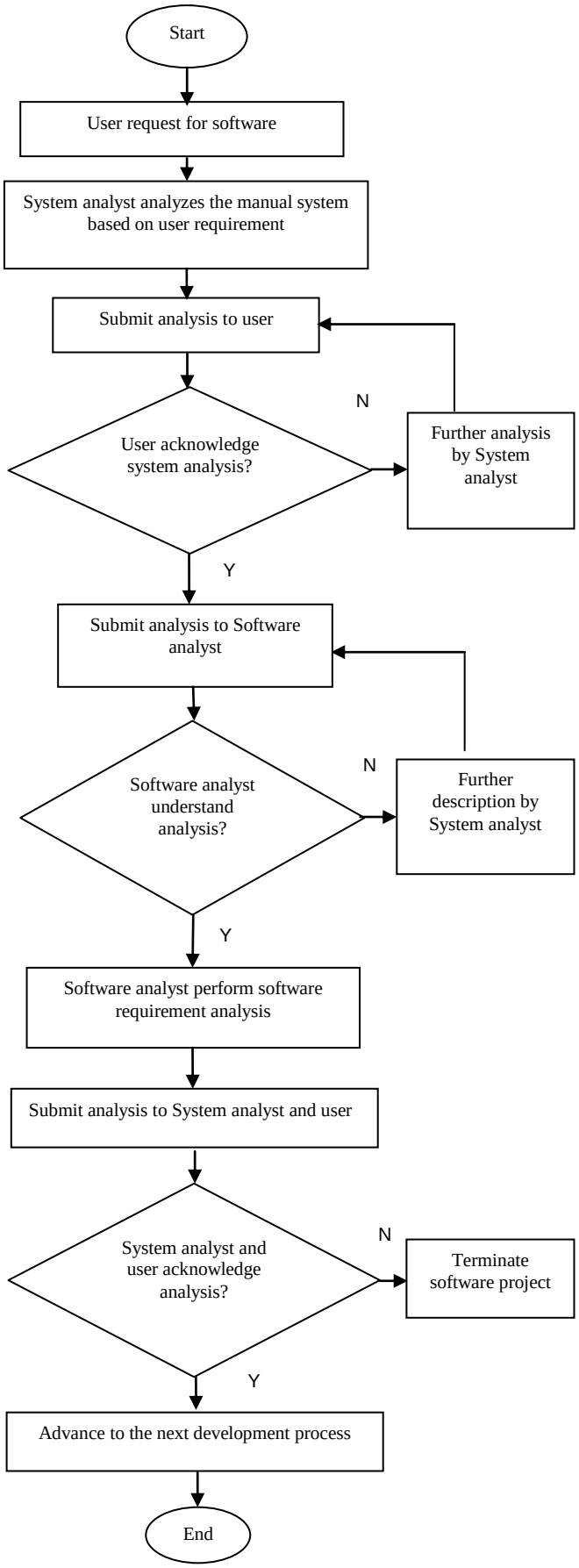

Figure 5. Requirement analysis process experts' interaction model.

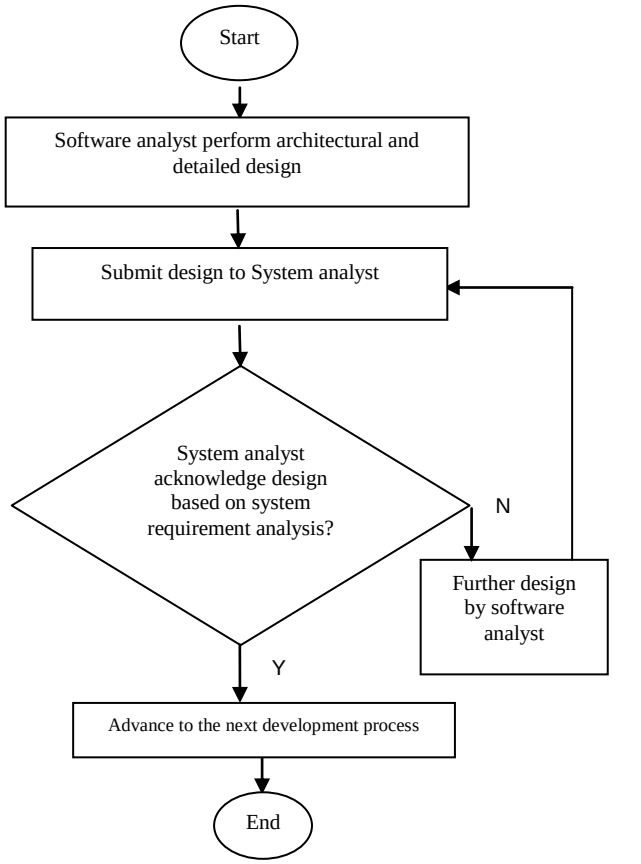

Figure 6. Design process experts' interaction model.

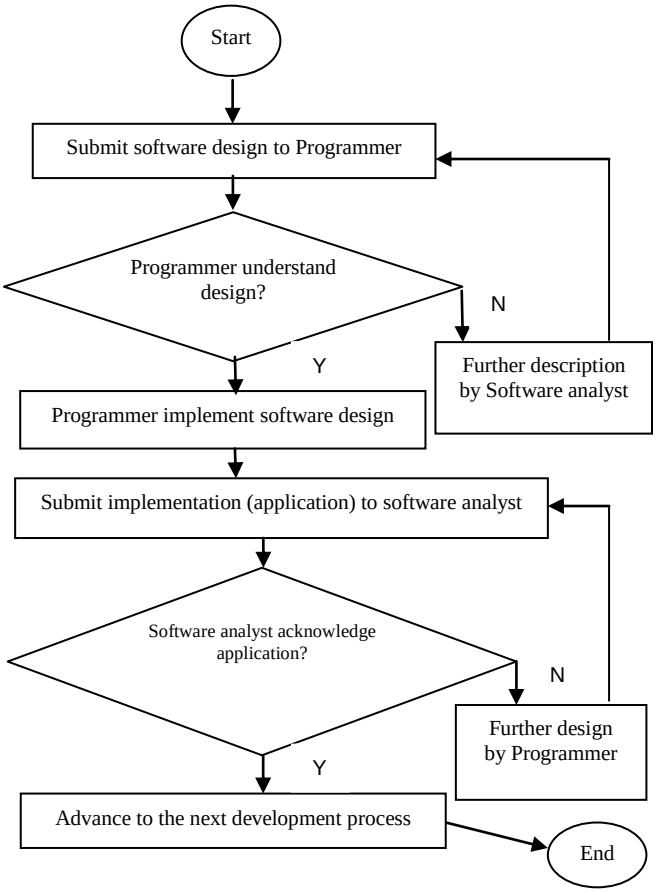

Figure 7. Implementation process experts' interaction model.

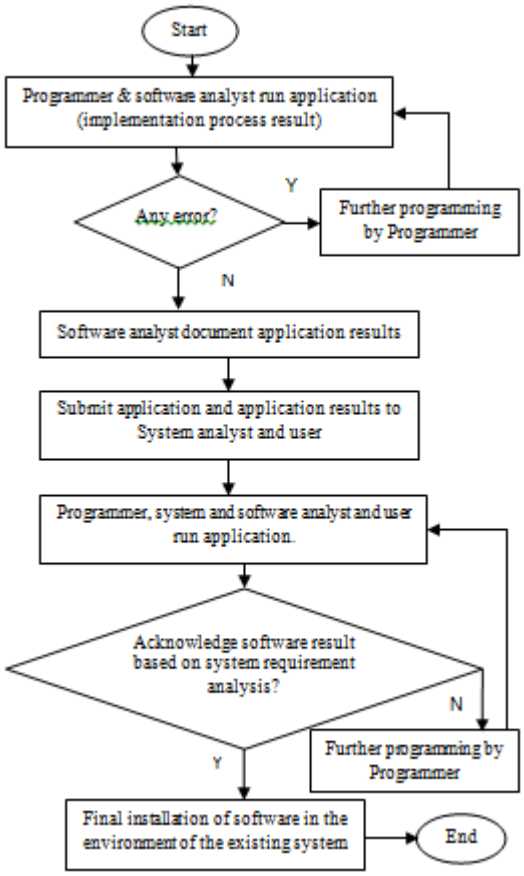

Figure 8. Verification process experts' interaction model.

(iii) Stakeholders' Collaboration Model Development

As described in fig. 9, a Collaboration Model for software stakeholders' collaboration during software development using iterative software development process model was developed, and fig. 5, fig. 6, fig. 7, and fig. 8 was used during the development of the model. As it is done in an iterative software development process model or iterative software process model where every process are iteratively carried out based on the number of software project segment decided by the software developers, so will the experts' interactions be carried out iteratively based on the number of decided software project segment. In the proposed model, A, B, C and D are used to represent fig5, fig6, fig7 and fig8 respectively.

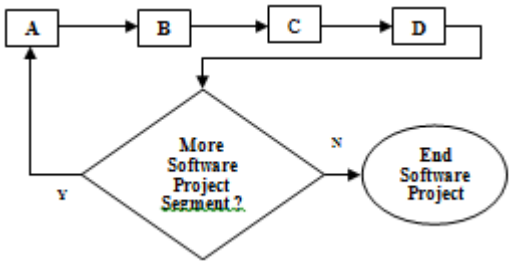

Figure 9. Iterative Software Process Based Collaboration Model for Software Stakeholders.

(iv) Benefits of this Study

In developing this study proposed model, certain common problems are clarified like understanding the activities carried out in each process of an iterative software process model, the experts involved in carrying out the activities in each process and how they interact to achieve results in each process. And this study proposed model can be used for software stakeholders' collaboration during software development using iterative software process model, thus helping to minimize the issue of lack of understandable collaboration or communication between software development stakeholders.

(v) Study Limitation and Future Contribution

Every model needs to be validated to know the contingencies involved and how applicable is the model. This study model is yet to be validated; however the validation can be a future research direction in software engineering. And this study proposed model is an iterative software process model based, therefore the limitations involved when using an iterative software process model for software development also affects this study proposed model.

Conclusions

Due to the fact that software engineering minimizes software engineering complexities researches are encouraged to be carried out to encourage and improve the practices of software engineering. The lack of understandable collaboration between software stakeholders during software development been part of the reason for software development complexities, research was carried out to minimize such problem. However, some limitations to the solution proposed in such research were taken into consideration and solution to some of the limitations was proposed in this paper. Based on the future research directions observed in this paper, further research can be carried out to improve this work or other areas of software development complexities minimization.

References

[1] A. S. Muhammad, A. Haruna , R. A. Malik. *Collaboration Model for Software Stakeholders' Collaboration during Software Development for Small Scaled System.* IJSER, 8(6), 2229-5518. 2017

[2] J. Kearney, P. R. L. Sedlmeyer, W. B. Thompson, M. A. Gray & M. A. Adler. Software complexity measurement *Communications of the ACM*, *29*(11), 1044-1050. 1986.

[3] R.Kaur, & J. Sengupta. Software process models and analysis on failure of software development projects. *arXiv preprint arXiv:1306.1068*. 2013.

[4] C. Larman. Applying UML and Pattern: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd Edition. United State. 2005

[5] S. Balsamo, A. Di Marco, P. Inverardi, & M. Simeoni, Model-based performance prediction in software development: A survey. *IEEE Transactions on Software Engineering*, *30*(5), 295-310. 2004.

[6] M. J. Kour. A Comparative Analysis of Different types of Models in Software Development Life Cycle. *Global Journal of Multidisciplinary Studies*, *4*(3). 2015.

[7] N. M. A. Munassar & A. A. Govardhan. Comparison between five models of software engineering. *IJCSI*, *5*, 95-101. 2010.

[8] S. Maheshwari, & D. C. Jain. A comparative analysis of different types of models in software development life cycle. *International Journal of Advanced Research in Computer Science and Software Engineering*, *2*(5), 285-290. 2012.

[9] M. Cotterell, & B. Hughes. *Software project management*. International Thomson Computer Press. 1995

[10] H. Sharp, A. Finkelstein & G. Galal. Stakeholder identification in the requirements engineering process. In *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on* (pp. 387-391). IEEE. 1999.

[11] W.Scacchi. Process models in software engineering. *Encyclopedia of software engineering*. 2001.

[12] M. Bresler, T. Van Phan, D. Prusa, M. Nakagawa & V. Hlavac. "Recognition system for on-line sketched diagrams." *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014.