

# Collaboration Model for Software Stakeholders' Collaboration during Software Development of Small Scaled System

Muhammad Ahmad Shehu, Haruna Abdu, Malik Rufai Adeiza

**Abstract**— The rate of demands of software development in various domains system is increasing due to software effective and efficient execution of processes. However, even with software development importance, it is also faced with complexities. And to minimize these complexities software engineering was introduced. To improve the practices of software engineering for further minimization of software development complexities various researches were carried out. However, among the complexities of software development is the fact that major software developed fails (software project failure) and among the reasons why software project fails is the lack of understandable collaboration between the software stakeholders. To minimize this issue of lack of understandable collaboration between the software stakeholders, this study developed a collaboration model that can be used for collaboration between the software stakeholders during software development of a small scaled system using the most suitable software development process model and replacing it's processes (software process model stages) with their respective experts and the logical flow of interaction (using flow chart representation) between the experts.

**Index Terms**— Flowchart, Organization System, Software development, Software engineering, software complexities, software project failure, software stakeholders, software process models, Software Experts' Interaction, Software Stakeholders' Collaboration, Waterfall Process Model.

## 1 INTRODUCTION

THE automation of manual problems resolution processes (like mathematical problem resolutions, records management and so on) using computerized tools describes the process of software development. The significance of software development is directed towards the ease (less effort, time and resources) in carrying out a process for a desired output compared to manually carrying out the same process with the same output. And due to this significance, software development applicability to other domains has increased overtime.

However, there are complexities experienced by software. Software complexity is the measure of difficulty faced by software during its development state or during its execution state [7]. 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 eras of software development are eras before and after software engineering introduction [5]. The era of software development before the introduction of software engineering

was referred to as the preindustrial era and software complexities as software crisis [5]. And during this period the complexities of software development was very high due to the lack of well organized principles guiding software development process. But the introduction of software engineering (the use of acceptable guide and principles during software development) reduced the complexities of software development [5].

Larman [4] describes the principles of software engineering as follows: Abstraction; the software design should be decomposed in a way that the users or other software engineers should view and understand the system without knowing the technicality involved. Modularity; decomposed the design of the software into segments (modules) such that each segment can functionally or logically dependent. Localization; logically related computational unit should be collected as a module. Uniformity; there should consistency in syntax, comments and so on during coding. Completeness; every required components in modules and overall system should be visible. Software Confirmation; confirm the software readiness by testing module by module then overall software.

There are various practices involved in software engineering, however the use of software development process models for software development is the practice considered in this paper.

Due to the complexities minimization significance of adopting the practice of software engineering (the use of software development process models) during software development, various research contributions were carried out to improve this practice of software engineering.

Balsamo *et al* [1] focused on how model - based performance can be embedded in software development process so as to improve the maturity of software development. They

- Muhammad Ahmad Shehu is currently a lecturer of Federal University Lokoja, Nigeria, Mobile number: +2348108984117. E-mail:sdfax2014@gmail.com
- Haruna Abdu and Malik Rufai Adeiza are currently Lecturers of Federal University, Lokoja, Nigeria, Mobile Numbers: +2348038433143 (Haruna), +2348033585619 (Malik), E-mail:haruna.abdu@fulokoja.edu.ng, rufai.malik@fulokoja.edu.ng

discovered that the commonly used software development process models do not consider software performance in the early stages therefore causing some major performance setbacks of software developed using the common models. A review of those research contributions in software development process that considered software performance at the early stages of their proposed model was carried out.

Munassar, and Govardhan [10] and Maheshwari and Ch. Jain [9] both carried out a comparative study of some software development process model. Munassar, and Govardhan [10] did a comparative analysis on five models while Maheshwari and Ch. Jain [9] made a comparisons between four models. After both analyses it was concluded that each studied model has its respective advantages and disadvantages, any of the model can be used for development of software depending on the size of the software projects. It was concluded by Maheshwari and Ch. Jain [9] that for smaller software projects the waterfall model is commonly used due to the easy understanding of the user requirements and for large projects the spiral process model is commonly used due to the complexity involved in understanding the user requirements and the handling of risk involved in large software projects.

Kaur and Sengupta [6] carried out an analysis on software process models and failure of software development projects. Based on their research the measure of difficulty of software development (software complexity) analyzed was the failure of software project. And it was observed that software project failure was due to various reasons of which the lack of understandable collaboration or communication between experts within the team involved in the software development is one of the reasons. However, the size of the team involved during the software development affects the collaboration or communication strength, that is the smaller the team size the better the collaboration or communication strength [6]. And the lack of user involvement was another reason discovered for software project failure. User involvement during software project development is very essential due to their understanding of the system that needs to be automated therefore the software user needs to be involved during the entire software project [6]. However, with their analysis on software project failure, explicit resolution on how the reasons for software development failure could be handled was not analyzed during their research.

Based on the lack of understandable collaboration or communication between experts within the team involved in the software development and the essential feature of the user involvement during software development discovered by Kaur and Sengupta [6], a collaboration model for software stakeholders' collaboration during software development is proposed in this research. Software development process model was considered during the development of the proposed model since the proposed model is to be used during software development and the use of software development process model during software development is a necessary practice in software engineering.

Stakeholders in a project as defined by Cotterell and Hughes [3], Sharp, Finkelstein & Galal [12] are individuals who have stake or interest in the project. Based on this definition this research describes the stakeholders involved in soft-

ware development of a small scaled system as the individuals who affects or is affected by software development of a small scaled system.

The stakeholders identified in this study are categorized as Users and Developers. Each Category is elaborated as follows: Users; different studies have its interpretation of users. Sharp, Finkelstein & Galal [12] interprets users as those individuals or companies interacts and control the system directly and use the results of the system. However, this study identifies users as the stakeholders that are users and experts of the existing system, and the direct users of the proposed system (software). Developers; are the experts (called system analyst) in analyzing the existing system based on the ideas of the users, experts (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.

## 2 STUDY METHODS

### 2.1 Software Development Process Model

Software development process model is a graphical or abstract representation of the architectural or design definition of the development process of software [11]. The use of software development process model for software development is a very essential practice in software engineering due to its software complexities minimization significance [11]. There are various software development process models but each of the process models have it constrains and contingencies [10], [9], [8]. However, all the process models are practically used in software development. And one of the software process models is used as a base for developing this study proposed model.

### 2.2 Flowchart Representation

Flowcharts are symbolic and unified representations connected by arrows [2]. They are used to represents sequential steps for solving problems. Due to its descriptive nature, flowchart representational approach applicability in problem domains is not restricted to software engineering or computer science problem domains. Collaboration deals with interactive process between two or more entities to resolve problems, therefore this study used flowchart representational approach to represent the interactions between software stakeholders during software development.

## 3 STUDY FRAMEWORK

There are various software development process models in software engineering and their applications during software development depends on the size of the system considered to be analyzed and then computerized into the proposed software [10], [9]. However, in this paper the software development process model considered during the development of the proposed model is applicable to small scaled system. And the research framework used for the development of the proposed model is described in Fig. 1.

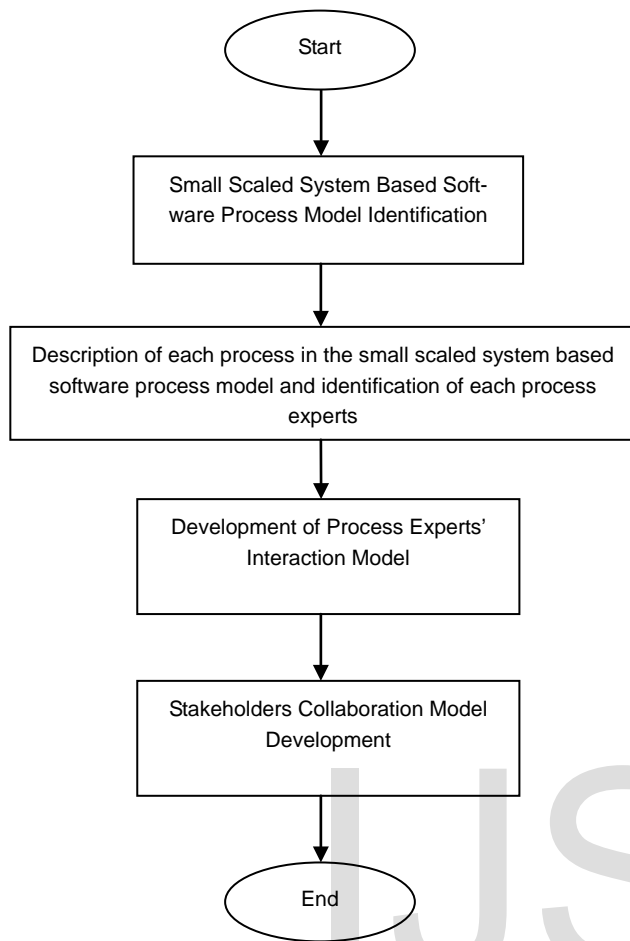


Fig 1. Study Framework

The framework used in the development of the proposed model begins with the identification of most suitable software development process model used to develop software for a small scaled system. In this stage, analysis on software process models was carried out, however the research analysis results of Munassar and Govardhan [10] and Maheshwari and Ch. Jain [9] where comparative analysis between several software process models were carried out was used to identify the software process model suitable to develop software for small scaled system.

In each software process model used during software development there are various processes and their respective experts (software stakeholders) carrying out the processes. However, to identify the expert or experts for each process of the chosen software process model for the development of software of a small scaled system, each process of the chosen software process model were analyzed. The identification of the experts (software stakeholders) involved in the processes of the chosen software process model was a necessary step due to the fact that the proposed model (collaboration model) was for the use of software stakeholders. And the description of each process of the chosen software process model helped in identifying the link between the software stakeholders.

Using the identified link between software stakeholders a flow chart representation was used to design experts' interac-

tion model for each process of the chosen software process model. Using the chosen software process model, the proposed model (collaboration model) was developed by replacing each of the process of the chosen software process model with the designed experts' interaction model.

## 4 STUDY RESULTS AND DISCUSSION

### 4.1 Small Scaled System Based Software Process Model Identification

Based on the analysis of Munassar, and Govardhan [10] where five software process models (Waterfall model, Iterative model, Spiral model, V- shaped and Extreme model) were compare, it was analyzed that each of the process models has its advantages and disadvantages and the suitability of the models depends on the size of the software to be developed (manual system to be computerized to software). Based on their discoveries, Waterfall and Spiral model are commonly used during software development.

According to Maheshwari and Ch. Jain [9] analysis, where he compared 4 software development process models, it was discovered that waterfall model was commonly used for software development of internal system of organizations due to its less expensive nature and feasibility of specifying all the requirements and Spiral model was used for software development of large and complicated system due to its ability of handling the highest risk involved in the project at the lowest cost.

Based on the comparative analysis of Munassar and Govardhan [10] and Maheshwari and Ch. Jain [9], this study concluded that for a small organization system to be computerized into software, waterfall software development process model is suitable to be used.

### 4.2 Description of each process in the small scaled system based software process model and identification of each process experts

The waterfall software development process model is described as a linear and sequential software development approach due to its steadily downward movement of development stages of execution. There are various descriptions of waterfall process model development processes or stages of execution by different literatures but Maheshwari and Ch. Jain [9] description was used in this paper. Waterfall process model stages are categorized into 5 stages namely; Requirement analysis, Design, Implementation, Testing or Verification and Maintenance (moving downward as described in Fig. 2).

It is said that waterfall process model serves as a baseline for the rest software process models. It emphasizes one time planning, time schedules, targeted dates budgeting and implementation of the entire system.

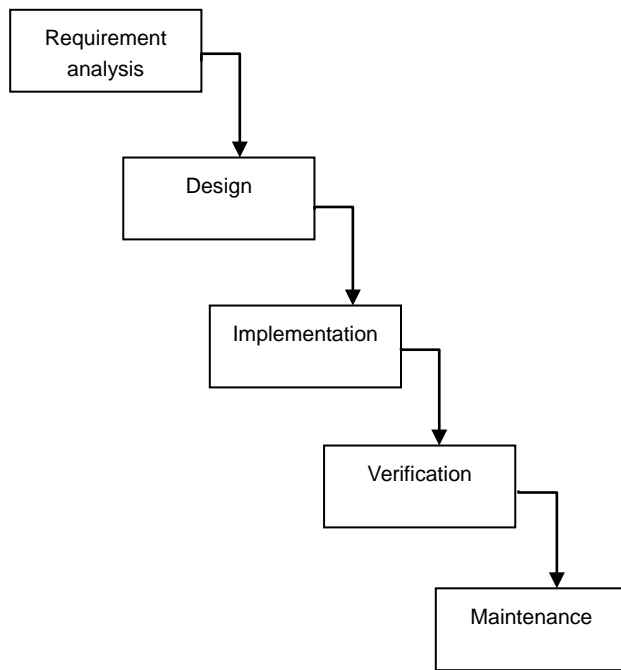


Fig 2. Waterfall Process Model

#### 4.2.1 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]. 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 require-

ment 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).

#### 4.2.2 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.

#### 4.2.3 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 programmer (for implementation) and software analyst (for acknowledgement of the implementation results).

#### 4.2.4 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.

#### 4.2.5 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 the involvement of each experts depends on the problem or user's enhancement request. The more the user requests the more the experts' involvement. However, it is noted that maintenance of the software can continue until the software life cycle is reached (when the software enhancement affects the logical results or the entire architectural or detailed design of the existing software).

#### 4.3 Development of Processes Experts' Interaction Models

Each Process of a waterfall model has its experts (as identified in the previous process) and the experts interact to achieve the result of the respective process. A model for each process is developed using flowchart representation for each process experts' interactions. The models are described in Fig. 3, Fig. 4, Fig. 5, Fig. 6 and Fig. 7.

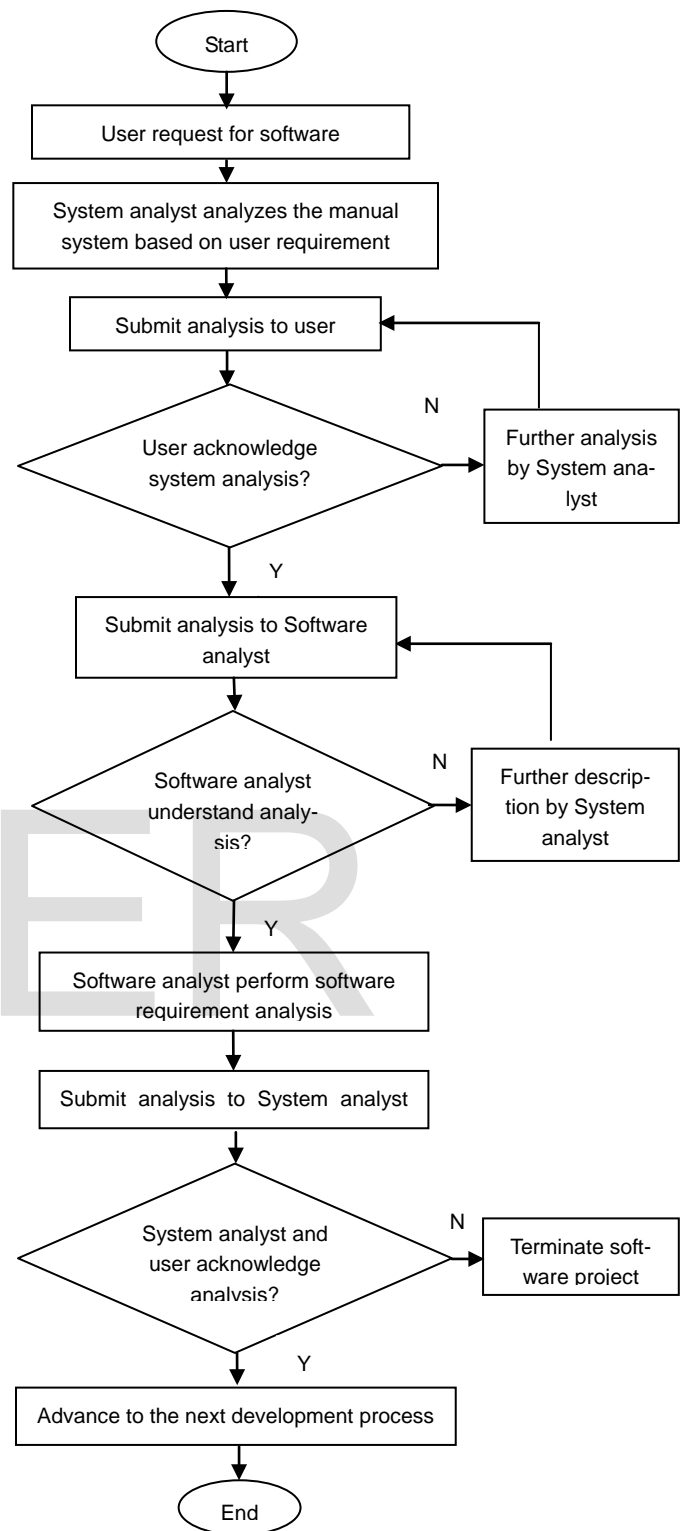


fig 3. Requirement analysis process experts' interaction model

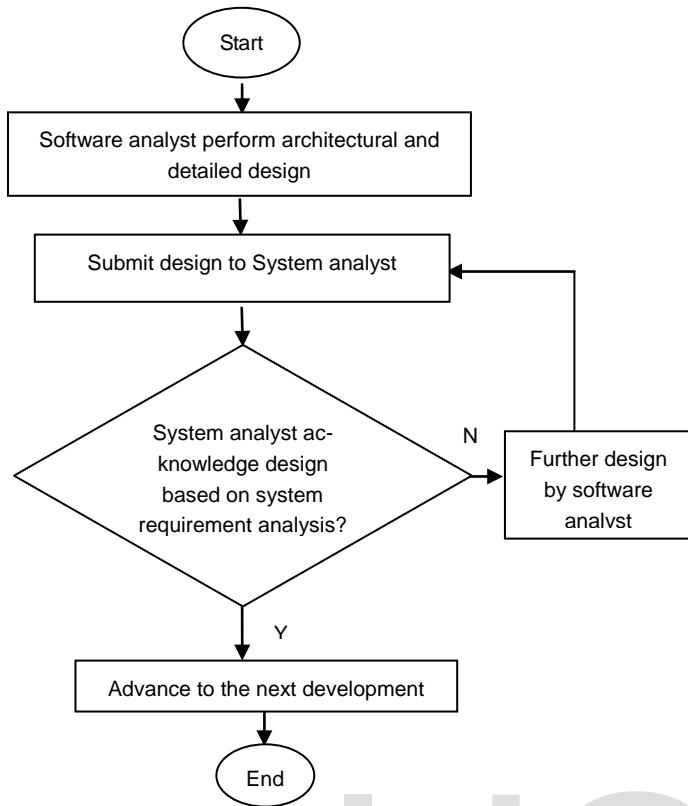


fig 4. Design process experts' interaction model

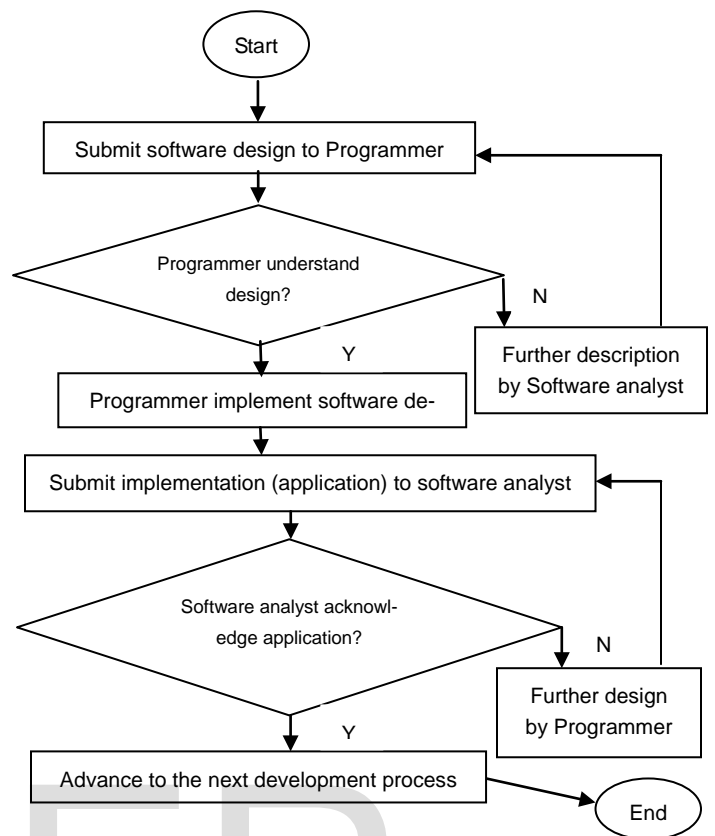


fig 5. Implementation process experts' interaction model

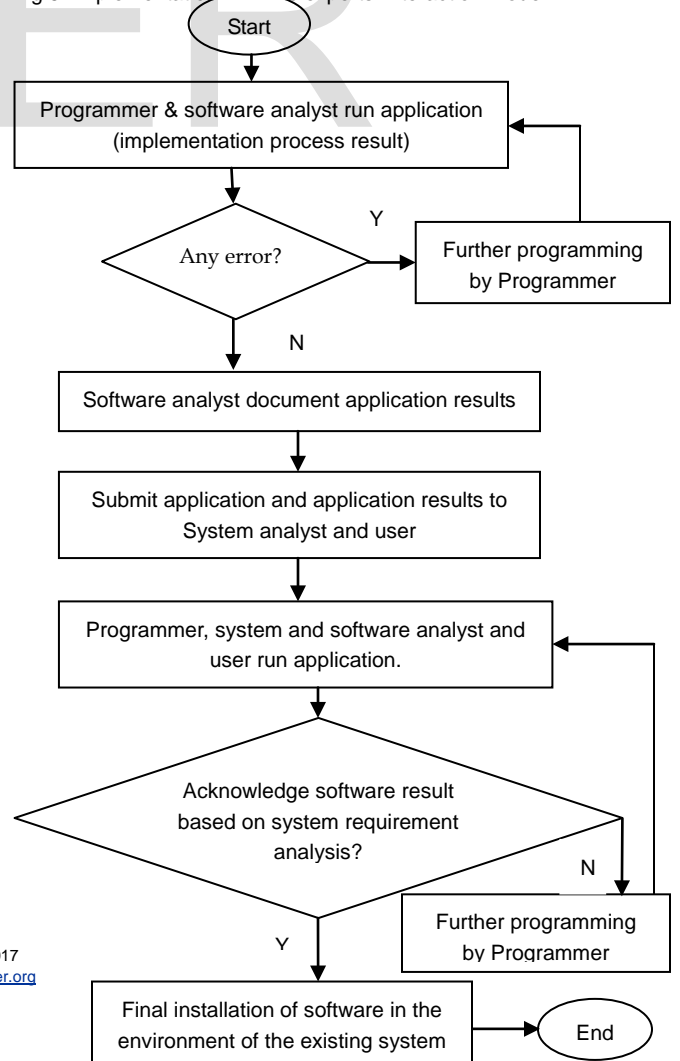


fig 6. Verification process experts' interaction model

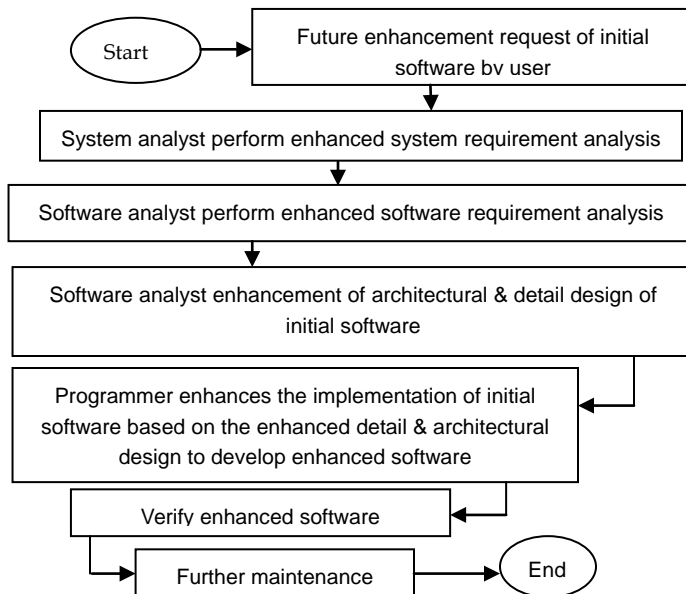


fig 7. Maintenance process experts' interaction model

#### 4.4 Stakeholders Collaboration Model Development

Using Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, the proposed collaboration model for software stakeholders' collaboration during software development was developed in Fig. 8:

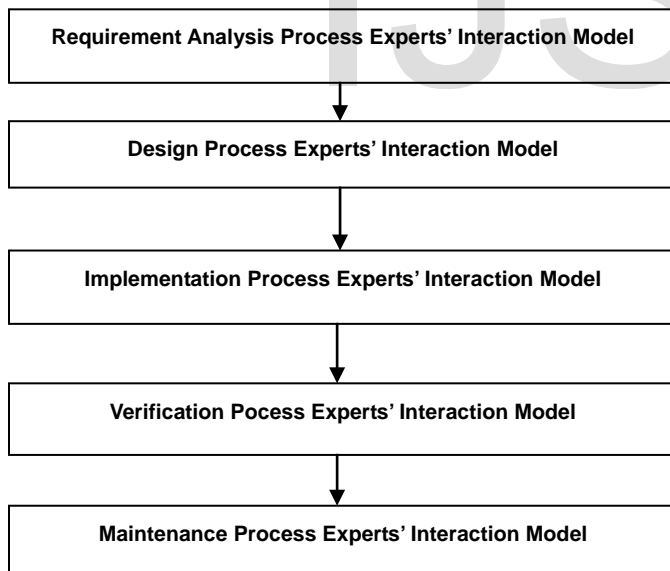


Figure 4 Software Stakeholders' Collaboration Model

#### 4.5 Potential Benefits of the Proposed Model

This study model helps to highlight and clarify the interaction and communication steps between software process experts' within a waterfall process model, and thus some well known problem that can be faced during the use of a waterfall process model for software development is clarified, like understand-

ing the activities carried out in each software process, identifying the experts carrying out the activities and every link between experts in each process, however for extensive clarification some literatures used in this paper in software development can be read. And the issue of user involvement during software development as a reason for software failure was extensively considered in this study model especially, where the knowledge of the users are essential, like in the requirement and verification processes.

#### 4.6 Study Limitation and Future Contributions

This study model (Figure 4) is a model for software stakeholders' collaboration during software development of a small scaled system. However, it is known that every proposed model should be validated to know how applicable the proposed model is in the real world, this way the proposed model setbacks and other contingencies can be observed. This study model is yet to be validated; however the validation of this study model is to be considered for future research contribution to the domain of software engineering.

This study considered waterfall process model during the development of the proposed model, therefore limitations affecting waterfall process model also affects the proposed model.

Also this study's proposed model is restricted to waterfall process model (i.e. collaboration of software stakeholders' during software development using waterfall process model), there are other software development process model that was not considered. Therefore further research can be conducted to develop a software stakeholders' collaboration model considering other software development process models.

#### 5 CONCLUSION

It is analyzed that software engineering minimizes the complexities of software development, therefore research are encouraged to be conducted to improve or encourage the practices of software engineering. Also there are various identified problems that can lead to software development complexities, therefore researches can be conducted to prevent or minimize or eliminate these problems. This study improved the use of waterfall software development process model and tries to resolve the problem of lack of understandable collaboration between software stakeholders by developing a collaboration model for software stakeholders' collaboration during software development of a small scaled system. With the future research directions identified in this paper and other literatures referred in this paper (specifically related literature to software development complexities), further research is advised to be conducted to improve the process of software development complexities in software engineering.

#### ACKNOWLEDMENT

#### REFERENCES

- [1] 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.
- [2] 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.
- [3] M. Cotterell, & B. Hughes. *Software project management*. International Thomson Computer Press. 1995
- [4] C. Larman. *Applying UML and Pattern: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3<sup>rd</sup> Edition. United State. 2005
- [5] W. GIBBS. SOFTWARE'S CHRONIC CRISIS. *SCIENTIFIC AMERICAN*, 271(3), 86-95. 1994.
- [6] R. Kaur, & J. Sengupta. Software process models and analysis on failure of software development projects. *arXiv preprint arXiv:1306.1068*. 2013.
- [7] 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.
- [8] M. J. Kour. A Comparative Analysis of Different types of Models in Software Development Life Cycle. *Global Journal of Multidisciplinary Studies*, 4(3). 2015.
- [9] 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.
- [10] N. M. A. Munassar & A. A. Govardhan. Comparison between five models of software engineering. *IJCSI*, 5, 95-101. 2010.
- [11] W. Scacchi. Process models in software engineering. *Encyclopedia of software engineering*. 2001.
- [12] 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.