Proposed Software Development Model for Small Organization and Its Explanation

Vinish Kumar[1] and Sachin Gupta[2]

[1] Department of Computer Science and Information Technology,
Sunder deep College of Engineering & Technology, Ghaziabad,
Affiliated to U.P. Technical University, Lucknow UP, India
vinishkumar_2001@rediffmail.com
[2] Department of Computer Science and Information Technology,
Raj Kumar Goel Institute of Technology, Ghaziabad,
Affiliated to U.P. Technical University, Lucknow, UP, India
imsachingupta@rediffmail.com

Abstract. Software Development is the process to illustrate the overall mechanism involved in the progress of software during different stages of development. To moderate the computational efficiency of earlier and later phases of development often occurred in small scale software developing organization we have proposed a new software development model for small organization. Through this model we can elicit the software requirement and we can also compute the functionality and risk in each and every phase of the software development.

Keywords: Software Model, Elicitation Techniques, Function Point, Software Risk.

1 Introduction

A software development life cycle (SDLC) model is a set of activities, if performed in a manner will produce desired product. A SDLC model specifies about the way these activities are framed to produce quality software.

The various life cycle models are-

1. The Waterfall model
2. Prototype model
3. Iterative model
4. Spiral model
5. Fish model
6. V-model
7. Object oriented modeling.

1.1 The Waterfall Model

This is the most common and classic of life cycle models, which states that the phase are organized in a linear order and phase must be completed before the next phase begins and the output documented in each phase are reviewed to determine that the

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 167–175, 2012.

project is in the right direction as per the requirement or need. The advantage of this model is that it is a simple model, stage and attributes are well defined and errors can be detected due to verification at each stage. The disadvantage of this model is that it can't incorporate new changes during the project development.

1.2 Prototype Model

In this model a throwaway prototype of the actual product is prepared. The prototype gives the client on actual feel of the system, after interacting with the prototype the client provide feedback to the developer regarding the changed to be needed in the system. Based on the feedback, the prototype is modified to incorporate some of the changes required, and then the client is again given the prototype and the cycle repeat until product completed.

The advantage of this model is that errors can be detected much easier as the system is made side by side, leads to a better system to uses and reduce risk [21].

The disadvantage of this model is that it may increase the complexity of the system as scope of the system may expand beyond original plans, Prototype has to be build on company's cost, The user may strict to the prototype and limit his requirements [25].

1.3 Iterative Model

In this model, the main idea is that software should be developed in increments. In 1st step a simple subset model of the overall problem is designed. The subset designed contains some of the key aspects of the problem. Which are easy to design and forms a useful and usable system? Each increment adds some new functional capability to the existing system and the process continues until the full system is implemented.

The advantage of this model is to generate working software quickly easily during the software life cycle. It accommodates changes easily, easy to test and debug due to smaller iteration and requirements are prioritized.

The disadvantages of this model are that each iteration can have the planning overhead. System structure may suffer as frequent changes are made and cost may increase due to undo work of one stage in other one.

1.4 Spiral Model

Spiral model is competitive of incremental model in which risk analysis has major work. It is most widely used to perform risk analysis at every level of development. It has base line which starts from planning phase, requirements and then terminates with risk analysis every time each subsequent spiral builds on base line spiral. The advantage of this model is High amount of risk analysis, good for large and mission critical projects.

The disadvantage of this model is that it is complex and can be costly to use, Risk analysis requires highly specifies expertise and does not work well for smaller projects [24].

1.5 Fish Model

This model is totally based on the validation and verification development strategies. It is widely used to avoid the any loss of content mismatch or cause of failure of

project/product. It is performed on the basis of verification and validation of product. It is started with initial phase and goes till the completion of project. The disadvantage of this model is that it is time consuming and extra cost for verification & validation [20].

1.6 V-Model

It is similar to waterfall model having sequential path of execution of process. Each phase must be computed before the next phase begins. Testing has move importance in this model as compared to others waterfall model. It has major vision on the testing of deliverables of each phase. We just prepare the test plan for each phase & test the plan objectives at the end of every phase [26].

1.7 Object Oriented Modeling

In order to avoid the different problems associated with quality data; we use the concept based on system dynamics. Now, on this basis an objects oriented approach to the development of software was proposed in late 1960's.This modeling requires the object oriented technique for development of software started with initial analysis phase to the implementation phase.

The advantage of this model is that it is based on problem domain. Hence it is easier to develop and debug the design; it is quick to accept the requirement changes, i.e. backtracking a reporting is much easier in OOM. Its design has higher re-use factor and allow changes easily. Due to above facts, it has reduced development cost and cycle time. The disadvantage of this model is that it has higher cost and cycle time. This paper is organized in various sections as following: The background and related work is explained section two. Section three briefs about block diagram of the proposed model. We have explained the proposed model in section four and at last the paper is concluded in section 5.

2 Background and Related Work

Various models have been developed over the years to manage the structural set of activities involved in developing and maintaining software system [20]. Since the advent of the first software development process model (waterfall model) , a lot of software development process models have been put forward [21], for example prototyping model, iterative model, spiral model[22],agile model[23] ,and so on .

The various models developed so far have come under attack, due to its rigid design and inflexible procedure [24], for example if the problem is well defined and well understood, waterfall model is sufficient .However, when we come up against a poorly defined problem, a popular variation of the prototyping model called Rapid Application Development (RAD) introduce firm time limits and relies heavily on rapid application tool which allow for quick development [25]. Have to go for spiral model [24].Prototyping model helps to understand uncertain requirements but leads to false expectations and poorly designed system.

Software project fails when they do not meet the criteria for success. most of project run over budget or are terminated prematurely and those that reach

complication often fall short of meeting user expectations and business performance goals [26].Using various reports on failure of software product, projected by Dan Galorath [27], author in [26] conclude the major factors involved in failure of a project are Requirement Elicitation, Lack of user involvement, Team size, Time Dimensions, Testing Etc.

The successful implementation or failure of a software system is totally dependent on the quality of the requirements. It is influenced by the various techniques employed during the very first phase i.e. requirements elicitation. Requirements elicitation, the most critical part of the software development, prompts errors at this beginning stage propagate through the development process and the hardest to repair later [28].

In the continuation of the above work we have proposed a software development model that will overcome those problems that we have in the previous models.

3 Proposed Software Model for Small Organization

In this paper we have proposed a Software model for small organization, the block diagram of the model is given in the figure (1).There are no description in the

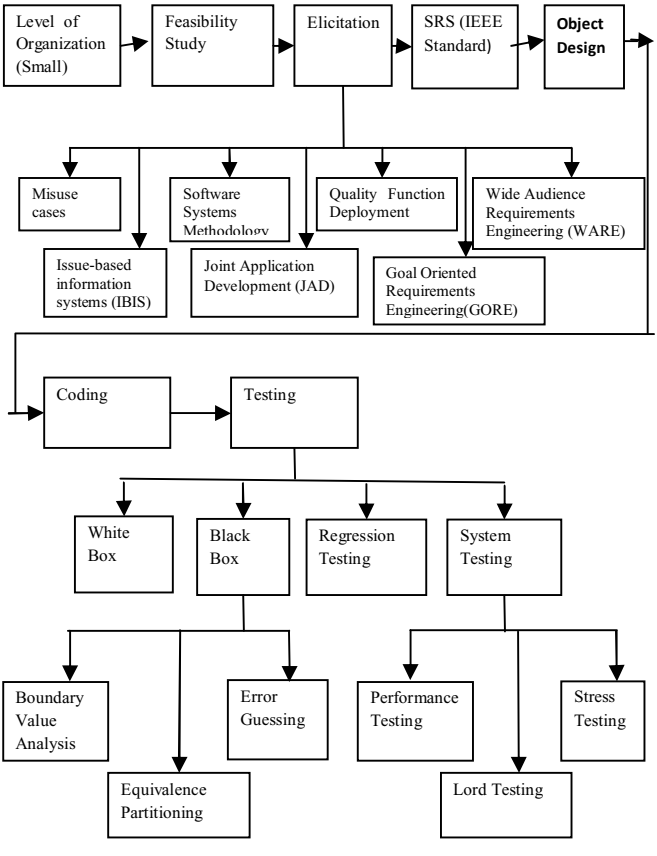

Fig. 1. Proposed Software Model for Small Organization

[21,22,23,24,25,26] about the type of organization where their proposed model works while we have considered the requirement of different types of organizations in our proposed model. The brief description about the organizations will be discussed during the phases of the model.

3.1 Level of Organization

According to Henery Fayal, organization is " A group of people work together i.e. the act of organizing a business or an activity related to business". Every organization has certain goals which they achieve in a specific period of time. There are various factors that play an important role in an organization are Size of organization, Manpower, Cost, Standards followed by the organization. On the basis of these factors the organization can be classified as:-

i). Small organization
ii). Large organization

In this paper we have proposed a model for small organization where the size of the organization is small. The man power is also less and the capital investment is also low, so there main focus is profit maximization along with maintenance of the standards and the product quality. But the organization has the capability to fulfill the functional and structural requirement of the project.

3.2 Feasibility Study

It is performed to decide the worth ability of product .It is based on information assessment, information collection and report writing. It is allowed to perform the feasibility study of any product with in two of three weeks with involvement of project manager, software engineers (who are about to develop the system), technical experts and customers (who will be using the system). Then prepare the FSR (Feasibility study report).It has different fields that incorporate technical, operational, economy, management, legal and time.

3.3 Requirement Elicitation

Elicitation is all about determining the needs of stakeholders. Requirement elicitation is recognized as one of the most critical knowledge intensive activities of the development of software because errors at this beginning stage propagate through the development process and the hardest to repair later. Studies indicate that 70% of the system errors are due to the inadequate system specification [31, 32].

3.4 SRS (IEEE Standard)

The origin of most software system is developed by developer and finally the software is used by the end user. so client does not understand the language of developing process which is used by develops and developers does not understand the need of customer, so to fulfill the communication gap client and developer SRS is

made. If the requirement is not understood properly the final product will be of no use, thus a SRS provides a reference for validation of the final product. IEEE (Institute of Electrical and Electronics Engineers) specification provides a standard approach to design SRS [21].

3.5 Object Design

The object design phase refers to the determination of the full definitions of the classes and associations used in the implementation, as well as the interface and algorithms of the methods used to implement operations.

3.6 Coding

Coding implement the design in the best possible manner. Basic criteria to judge a code of program are readability, time complexity, size of code (LOC) and space complexity. These are the objectives should be clear to the programmer. He must know weather to reduce the execution time or to reduce the required memory to save the program etc.It enables different techniques as code inspection, verification, code refraction etc. to avoid bugs.

3.7 Testing

Software testing is used to find out as many errors as possible in the given software. There are two types of software i.e. black box and white box testing [21].

4 Explanation of the Proposed Model

Before the development of any software project, the first step is to understand the requirements of the software. In the literature of software engineering we have standard frameworks like NFR [28] (non- functional requirements), i*[30], KAOS [27], and TROPOS [29]. From these frameworks we can easily find out the functional and non-functional requirements of the software. After capturing the requirements, we have the next step which is the elicitation of the software requirements. There are several techniques to elicit the software requirements like [32], but in this paper we have used the elicitation technique which is the slightly modified version of the algorithm proposed by Mohd. Sadiq et al. [31] and its algorithm is given in algorithm (01).

1. With the help of the following steps elicitate the software requirements
1.1 Draw information about user requirements.
1.2) Test and Train the Users, Clients and Managers.
1.3 Write the description of the user need for the proposed system.
1.4 Now you can apply Wide Audience Requirement Engineering (WARE) because, it helps the requirements engineering process in the international research projects. The different organizations that take part in these kinds of international projects can be divided at least into the following categories: Governmental funding organization, other nonprofit organization, and industrial research centers.

2. In this algorithm we have used AHP technique for prioritization.
For using AHP algorithm
The overall performance matrix is created
Then calculate the Importance Weight (Eigen vector)
3. Find out the risk associated with each requirement.
4. Compare the values of the importance weight of software requirements with step 3 and then rank or Prioritize the requirements.

Proposed Algorithm Using WARE)
(Adopted from [31]
Algorithm (01)

After reviewing several research papers, we conclude that there are no descriptions about the goal oriented requirement engineering (GORE) and Wide Audience Requirement Engineering (WARE) in the previous software development model and there is also no description about the type of the elicitation methods that would be used to elicit the software requirements. Proposed model overcome these problems and gives the detailed information and knowledge about the type of GORE, WARE and also the type of elicitation method to the developers or the software engineer.

Once we have elicit the software requirements, the next step is to prioritize the software requirements, for this purpose we have used the Analytic Hierarchy Process (AHP). The detailed information about the AHP and its applications in software engineering are available in [34].

After prioritizing the software requirements the next step is to document these requirements. For documentation we generally used the IEEE format of the SRS i.e. Software requirements Specification [21]. Once we have finalized the SRS documents the next step is to develop the object design. The last two methods are common in most of the software development models i.e. coding and testing. But in the existing models there are no descriptions about the type of testing. In this paper we have preferred the black box testing [ ], because in this testing we generally work on the functionality of the software, for black box testing, we have several technique, like boundary value analysis, robustness techniques, worst case etc.

Apart from different models, in this model we have also added two extra features. The first feature is the risk estimation and the second feature is to check the functionality of the software. To compute the risk we have applied the risk estimation model proposed by Daya Gupta and Mohd. Sadiq [16] and [9, 11] .To compute the functionality of the software we have used the FFP [13, 14] (Fuzzy Function Point) instead of FP [4, 5, 6, 7, 8, 12, 15] i.e. Function point.

5 Conclusion and Future Work

In this paper we have proposed the software development model for small organization. In this model we have added several new features which are the demand of the software industry. In this model we have included requirement elicitation method. GORE and WARE are two different key concepts that are used to find out the requirements from the stakeholder's point of view. In this model we have also provide the facility to compute the functionality of the software during each phase of the development of the software. Risk estimation is also the added feature of this model. In future we will apply to this model in real life project and will differentiate the functionality of the proposed model with the existing ones.

References

[1] Albrecht, A.J., Gaffney, J.E.: Software Function, Source Lines of Code and development Effort Prediction: A software Science Validation. IEEE Trans. Software Engineering, 639–648 (1983)
[2] Albrecht, A.J.: Measuring Application Development Productivity. In: Proc. IBM Applications Development Symp., Monterey, Calif., pp. 14–17 (1979)
[3] Boehm, B.W.: Software Engineering Economics. Prentice Hall (1981)
[4] Lokan, C.J.: An Emprical Study of the Correlations between Function Point Elements. In: Proc. of 6th International Symposium on Software Metrics (1999)
[5] Ho-Leung, TSOI: To Evaluate the Function Point Analysis: A Case Study. International Journal of the Computer, the Internet and Management 13(1), 31–40 (2005)
[6] International Function Point User Group (IFPUG), Function Point Counting Practices Manual, Release 4.0, IFPUG, Westerville, Ohio (April 1990)
[7] Kemerer, C.F.: An Emprical Validation of Software Cost Estimation Models. Comm. ACM 30(5) (1987)
[8] Low, G.C., Jeffery, D.R.: Function Point in the Estimation and Evaluation of the Software Process. IEEE Trans Software Engineering 16(1) (1990)
[9] Sadiq, M., Rahman, A., Ahmad, S., Asim, M., Ahmad, J.: esrcTool: A Tool to Estimate the Software Risk and Cost. In: IEEE Second International Conference on Computer Research and Development, pp. 886–890, doi:10.1109/ICCRD.2010.29
[10] Sadiq, M., Ahmed, S.: Computation of Function Point of Software on the basis of Average Complexity. In: Proceedings of 2nd International Conference on Advanced Computing and Computing Technologies, ICACCT 2007, Panipat, Haryana, pp. 591–594 (2007)
[11] Sadiq, M., Sunil, Zafar, S., Asim, M., Suman, R.: GUI of esrcTool: A Tool to Estimate the Software Risk and Cost. In: The 2nd IEEE International Conference on Computer and Automation Engineering (ICCAE 2010), Singapore, pp. 673–677 (2010)
[12] Sadiq, M., Ahmed, S.: Relationship between Lines of Code and Function Point and its Application in the Computation of Effort and Duration of a Software using Software Equation. In: International Conference on Emerging Technologies and Applications in Engineering, Technology and Sciences, ICATETS 2008, Rajkot, Gujarat, India (2008)
[13] de Souza Lima Jr., O., Parias, P.P.M., Belchior, A.D.: A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessment. CLEI Electronic Journal 5(2) (1999)
[14] Kumar, S., Kumar, V., Gupta, M., Gupta, S.: Function Point Analysis for Small Organization based Projects using Triangular Fuzzy Numbers. In: 2011 3rd International Conference on Conference on Electronics Computer Technology, ICECT 2011 (2011)
[15] Symons, C.R.: Function Point Analysis: Difficulties and Improvements. IEEE Trans. Software Engineering 14(1) (1988)
[16] Gupta, D., Sadiq, M.: Software Risk Assessment and Estimation Model. In: IEEE International Conference on Computer Science and Information Technology, Singapore, pp. 963–967 (2008)
[17] Walston, C.E., Felix, C.P.: A method of program measurement and estimation. IBM System Journal 16 (1977)
[18] Zheng, Y., Wang, B., Zheng, Y., Shi, L.: Estimation of Software Project effort based on functional point. In: Proceeding on 2009 4th International Conference on Computer Science & Education, pp. 941–943. IEEE (2009), 978-1-4244-3521

[19] Zuse, H.: Software Metrics-Methods to Investigate and Evaluate Software Complexity Measures. In: Proc. Second Annual Oregon Workshop on Software Metrics, Portland (1991)
[20] Dieter Rombach, H., Verlage, M.: Directions in Software Process Research. In: Zelkowitz, M.V. (ed.) Advances in Computer, vol. 41, pp. 1–3. Academic Press (1995)
[21] Pressman, R.S.: Software Engineering: A Practitioner's Approach, 6th edn., pp. 77–99. McGraw-Hill, New York (2005)
[22] Boehm, B.W.: TRW Defense Systems Group. A Spiral Model of Software Development and Enhancement
[23] Martin, R.C.: Agile Software Development, Principles, Patterns and Practices. Prentice Hall, NJ (2002)
[24] Boehm, B.: A Sprial Model of Software Development and Enhancement. IEEE Computer 21(5), 61–72 (1988)
[25] Martin, J.: Rapid Application Development. Prentice-Hall Publication (1991)
[26] Kaur, R., Sengupta, J.: Software Process Models and Analysis on Failure of Software Development Projects. IJSER © (2011)
[27] Doerr, J., Kerkow, D., Koenig, T., Olsson, T., Suzuki, T.: Non-Functional Requirements in Industry – Three Case Studies Adopting an Experience-based NFR Method. In: Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering, RE 2005 (2005)
[28] Glinz, M.: On Non-Functional Requirements. In: 15th IEEE International Requirements Engineering Conference, pp. 21–26
[29] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Sytems 8, 203–236 (2004)
[30] Yu, E.S.K.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, RE 1997 (1997)
[31] Kumar, S., Kumar, V.: Proposed algorithm using WARE for the elicitation of the software requirements. International Journal of Engineering and Technology
[32] Sadiq, M., Ghafir, S., Shahid, M.: An Approach for Eliciting Software Requirements and its Prioritization using Analytic Hierarchy Process. In: IEEE International Conference on Advances in Recent Technologies in Communication and Computing, Kerala, India (2009)
[33] Sadiq, M., Shahid, M.: Elicitation and Prioritization of Software Requirements. International Journal of Recent Trends in Engineering, Finland
[34] Sadiq, M., Shahid, M., Ahmed, S.: Adding Threat during Software Requirements Elicitation and Prioritization. International Journal of Computer Application