A Methodology to Manage the Changing Requirements of a Software Project

Muhammad Wasim Bhatti
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
mwasim_bhatti@yahoo.com

Farah Hayat
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
farahayat@yahoo.com

Nadeem Ehsan
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
m4nadeem@yahoo.com

Sohail Ahmed
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
sohail_phatak@yahoo.com

Azam Ishaque
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
azamishaque@yahoo.com

Ebtisam Mirza
Engineering Management Department
CASE, Center for Advanced Studies in Engineering
Islamabad, Pakistan
ibtisam.m@gmail.com

***Abstract*—Requirements evolve and managing requirements evolution is necessary to meet the business needs of a customer. The continuous changing requirements can affect the cost, schedule and quality of a software project. Inability to manage the changing requirements may lead to inconsistencies in system requirements and results in failure of the software project. An effective methodology to manage changing requirements is, therefore, necessary for successful project execution. This paper suggests a formal methodology to manage the changing requirements of a software project. In proposed methodology the requirements change management process is break-downed in multiple phases. The major stakeholders and typical work products of each phase are identified. The role of each stakeholder is explained with respect to relevant phase of the process. It has been identified that the Change Control Board (CCB) is the process owner and the main stakeholder of the change management process. The proposed methodology, which is based upon six phases, continuously manages the change requests throughout the whole life cycle of the project.**

***Keywords* - Change Management Process, Changing Requirements, Methodology, Software Project**

I. INTRODUCTION

Requirement management is the first link of the software development chain [1]. The purpose of requirement management is to develop an understanding of system requirements between customer and software team [2]-[4]. Requirements related problems are one of the major reasons for the failure of software projects [5]. Changing requirements affect the cost, schedule and quality of software. Changing requirements is one of the major requirements related problems, and it is considered as a major cause of the failure of software projects [6, 7]. Normally, it becomes impossible to define the problem, design the solution and implement the software without change. Especially, for complex software it becomes difficult to understand all requirements in beginning of the life cycle. The reason of changes can be external factors or internal factors. External factors can be customer's pressure, and the changing business needs of the customer's organization. Internal factors can be poor understanding, and technical unfeasibility of customer's requirements [8].

The process of requirement change management not only manages the changing requirements but also helps in successful project completion [9]. Without proper requirement change management process a single change in one requirement can have ripple effect on many other requirements. This eventually leads towards many inconsistencies in software requirements and ends with the customer dissatisfaction and failure of the whole project.

The software industry lacks from a well organized change management process. A formal methodology for managing the change requests can be helpful in addressing the requirements' inconsistency issues. The formal methodology can also be helpful in reducing the project failure ratio of the software industry.

In this paper, the main reasons of changes in requirements are discussed. A formal change management process is introduced. All the steps of the change management process are explained and main stakeholders of each step are identified. The role of each stakeholder is explained in each step of the process. Typical work products of each step of the process are identified.

II. REVIEW OF LITERATURE

According to the Capability Maturity Model Integrated (CMMI), the change requests of all configuration items should be tracked by storing the change requests in a database and by analyzing the impact of change requests on existing requirements [3], [4].

Wang and Lai [8] believe that for complex software it becomes difficult to understand all requirements in beginning

978-1-4244-7818-7/10/$26.00 ©2010 IEEE

of the life cycle. Normally, when the requirements are accepted and approved then they are designed and implemented. But, it becomes impossible to define the problem, design the solution and implement the software without change. Wang and Lai proposed a requirement management model for incremental software development. They proposed that quantitatively management of software requirements in the form of a matrix is a good approach. Therefore, the structure of data should let some metrics to be built, and it should let to measure the quality of associated processes. They built a data collective structure called requirement matrix. Each requirement is assigned some attributes. These attributes are id, priority, scheduled iteration, status, change cause, etc. Distribution of requirements is the first matrix in requirement management process area. This matrix measures the number of requirements, number of requirement changes and type of changes in an incremental cycle. Wang and Lai proposed that Target Requirements are equal to the sum of Original Requirements and Increased Requirements minus Deleted Requirements. Similarly, All Changed Requirements are equal to the sum of Increased Requirements, Deleted Requirements and Revised Requirements.

Delgadillo and Gotel [10] proposed the concept of lightweight requirement management. They proposed the concept of Story-Wall. Story-Wall consists of story cards. Lightweight traceability of changes of the story can be achieved by adding simple history to each story card.

Pressman [11] believes that in an incremental development model, a software is developed in multiple iterations. Each iteration consists of a complete set of processes, i.e. requirement management, system design, implementation, testing and so on. Each iteration acts as a foundation for successive iterations. Therefore, requirements in one iteration should be stable. Customer's change requests can be handled in successive iterations.

Luo, Kar, Sahu, Pradhan, and Shaikh [12] proposed a model for requirement change management process. The main steps of the proposed model consist of "Change Requirement Determination", "Change Approval", and "Perform Change". They proposed this model for Enterprise IT services.

Kobayashi and Maekawa [13] proposed a Need Based Requirement Change Management Model (NRM). NRM model effectively manages the change requests of a project. NRM model defines the system requirements in the form of 4W (where, who, why, what). Because of 4Ws, it becomes possible to understand the requirements more accurately. NRM model consists of verification and validation activities, and it continuously watches the change requests of the system.

III. Main Objective

The main objective of this research is to propose a methodology to manage the changing requirements of a software project. This, therefore, may reduce the inconsistencies of requirements and failures of the software project.

IV. Reasons of changing requirements

Initially, software requirements are provided by a customer. Then the change in requirements is requested by the customer and the project team. There can be many reasons for the change in initial requirements. Customer may request to change the initial requirements of software, if some changes occur in the business process of client's organization, if analysts couldn't initially understand the exact requirements of customer, or if initially provided requirements remained incomplete. There can also be some other reasons for the change requests of the customer, depending upon the nature of the project and the type of the deficiencies in initial requirements. Project team can also request changes in requirements. They may request a change if some requirements are technically impossible to implement, or if some requirements fall outside the scope boundaries of the software project.

V. The Process of "Requirements Change Management"

The process of requirement change management begins when a customer or a member of the project team requests for a change in existing definition of requirements. This step is called the initiation of the change management process. This change request is received, evaluated, approved or disapproved, if approved then implemented and then finally stored in a configuration repository. Different stakeholders are involved in different steps of the process. The main stakeholder and owner of the process are the change control board (CCB) team. CBB consists of members from all teams of a software project. The main role of CCB is to keep track of all changes and to take decisions in entire change request cycles of a project. Fig. 1 shows the flow of the requirements change management process and the role of stakeholders in each phase of the process.

The detailed explanation of all steps of the requirement change management process is as follows;

A. Initiation of Change Management Process

The initiation of change requests is the first step of the change management process. This step is simply putting a request for changes in requirements of a software project. The process of change management is initiated when a customer or a member of the project team requests for a change. The major stakeholders of initiation step are "customer" and "team members of the software project". The typical work product of this step is "change request" of requirements.

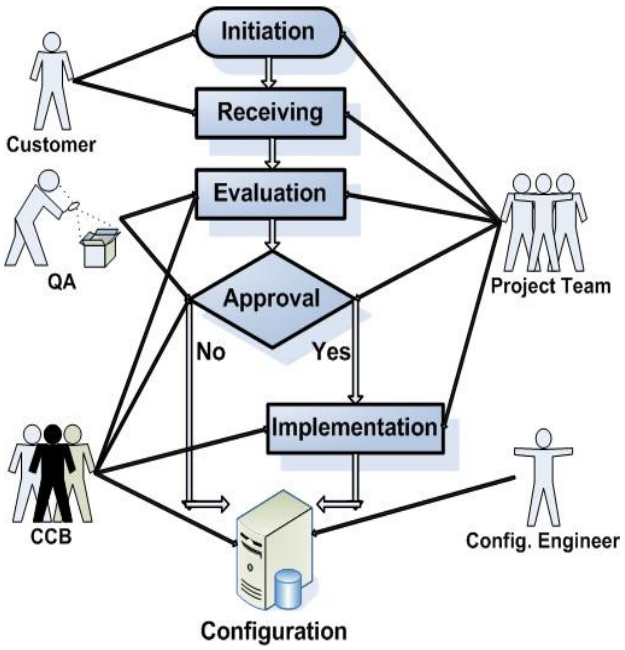

Fig.1, Requirement Change Management Process

B. Receiving of Change Request

There should be a standard change request form. When a customer or a member of project team requests for change in requirements, this change request should be received on a standard change request form. The typical components of a change request form can be "Requested by", "Date of Request", "Title", "Project", "Module", "Work Product", "Request Type", "Severity", "Priority", "Request Received by", and "Description". Change requests can also be received through any workflow application. Once a change request is received, it should be processed through a complete change management process. The major stakeholders of this step are the "initiator of the request (customer or project team member)", and the "receiver of the request".". The typical work product of this step of the process is complete filled "change request form".

C. Evaluations of Change Request

There should be a formal change control board (CCB) to process the received change requests. CCB should be evaluating a change request through a number of checks, depending upon the nature of software project and nature of the change requests. The impact of change should be analyzed on existing requirements. The change request should also be evaluated against technical constraints of the project. The cost benefit analysis of the change requests should be performed. Quality assurance team should also be part of evaluation step of the process. The major stakeholders of evaluation step are "CCB", "quality assurance team", and "relevant team members" of the project. The typical work product of evaluation step is "evaluation sheet". The evaluation sheet explains the evaluation details of a change request.

D. Approval or Disapproval of Change Request

After complete evaluation, the change request should be approved or disapproved by CCB. The base condition of a change request for being approved or disapproved is that whether the change is doable or not doable. The decision of a change request for being doable or not doable is done in evaluation phase of the change management process. In approval or disapproval step of the process, the formal decision is issued by CCB. If the change request is approved then notification is issued to relevant project team members for implementation of the change requests. However, if the change request is disapproved then the reasons of the disapproval are forwarded to the initiator of the change request. The major stakeholders of this step are "CCB", "quality assurance team", "relevant team members" of the project and "initiator of the change requests". In case of approval of change requests, the typical work product could be the "approval notification (may be in the form of an email)" of the change requests. In case of disapproval the "reasons of disapproval" of the change requests are the typical work products of this step.

E. Implementation of Approved Change Request

When a change request is approved by CCB, it goes for implementation. The nature of implementation depends upon the nature and time of approval of the change requests. If a change request is initiated and approved in requirement analysis phase then the implementation will require only the changes in requirements related documents e.g. SRS, use-case documents, requirement management plan etc. But if the change request is initiated in later phases of the project then the implementation will include the changes in running phase along with changes in all previous phases of the project. For example, if a project is in testing phase and customer requests for a change, and CCB approves it. Then the implementation of this change will involve the changes in testing related documents (e.g. test cases, test plan etc.), changes in code, changes in design document (SDS, ERD, DFD, and UML diagrams etc.), and changes in requirements related documents. So the change requests in early phases of project costs less than the changes in later phases of the project.

During implementation of changes, if any change occurs in the requirement management plan then project plan should be updated accordingly. The major stakeholders of implementation step are "CCB" and "change request implementers". The typical work product of this step is the document related to the "details of implementations" of the change requests.

F. Configuration of Change Request

History of all change requests of a project should be maintained and stored in a configuration management repository. Separate lists of approved and disapproved change requests should be maintained and stored in the configuration repository. The list of approved change requests should be maintained along with implementation detail. The list of disapproved change requests should be maintained along with the reasons of disapproval of change requests. All configurations related activities are performed by the configuration engineer. Therefore, the main stakeholder of this

step is configuration engineer and CCB. The typical work product of this step is the "list of configuration items".

VI. Findings

The formal methodology of the change management process proposed in this paper consists of six phases; initiation, receiving, evaluation, approval or disapproval, implementation and configuration of change requests.

1) In initiation phase, the customer or members of project team requests for a change in existing definition of a project.
2) In receiving phase, the change request is formally received on a standard change request form.
3) In evaluation phase, the change request is evaluated against a number for checks.
4) In 'approval or disapproval' phase, the approved request is forwarded to relevant team members for implementation. While the reasons of the disapproved requests are forwarded to change requestor.
5) In implementation phase, the relevant team members implement the change request and update the relevant work products.
6) In configuration phase, all the change requests are stored in a configuration repository for future reference.

VII. Conclusion

The changing requirements of a project can result into difficult and costly errors in the requirements' definition of a project. Changes are requested, if initially provided requirements remained incomplete, inconsistent or technically impossible to implement. The proposed formal change management process is useful in managing all the change requests of a software project. By adopting the formal change management process, the inconsistency in requirements and failures of the project can be avoided. The formal change management process is easy to understand and easy to use. Every step of the process is explained in detail. The role of stakeholders and typical work products of every step of the process is discussed in detail. Therefore, now any software company can adopt the proposed model to manage their varying requirements.

VIII. Future Work

There are two major dimensions of the future work of this research paper. These are as follows;

1) A workflow application can be developed for the proposed change management process. This workflow application can benefit the software industry in better decision making against change requests and in managing the change requests in more organized manners. The whole process will consume less of time in completing the change request cycle. The nature and history of change requests can be viewed and analyzed through the reporting module of workflow application.
2) The proposed methodology can be considered as a tool for the measurement of the change management process. The maturity of the change management process of software industry of any region can be analyzed and measured against the steps and practices processed in this research paper.

References

[1] D. Zylbermann, Y. Cohen, and L. Goldin, "The Road to Requirements Maturity", Proceedings of the IEEE International Conference on Software-Science, Technology & Engineering (SwSTE03), 2003.

[2] G. Cuevas, A. Serrano, and A. Serrano, "Assessment of the Requirement Management Process using a Two-Stage Questionnaire", Proceedings of the Fourth International Conference on Quality Software, IEEE Computer Scociety, 2004.

[3] Product and Development Team, "Capability Maturity Model Integration (CMMI), Version 1.1, Continuous Representation," tech. report CMU/SEI-2002-TR-011, Carnegie Mellon University, Pittsburgh, 2002.

[4] Product and Development Team, "Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation," tech. report CMU/SEI-2002-TR-012, Carnegie Mellon University, Pittsburgh, 2002.

[5] I. Sommerville and J. Ransom, "An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement", ACM Transactions on Software Engineering and Methodology, vol. 14, no. 1, 2005, pp. 85-117.

[6] S. Lock and G. Kotonya, "An integrated framework for requirement change impact analysis", Proceedings of the 4th Australian Conference on Requirements Engineering, Sydney, Australia, 1990, pp.29-42.

[7] E. Oz, "When Professional Standards are Lax, The CONFIRM failure and its Lessons", Communication of the ACM, vol.37 no.8, 1994, pp.29-43.

[8] Q. Wang, and X. Lai, "Requirement Management for the Incremental Development Model", Proceedings of the Second Asia-Pacific Conference on Quality Software, 2001.

[9] K. E. Emam, D. Holtje and N. H. Madhavji, "Causal Analysis of the Requirements Change Process for a Large System", Proceedings of the International Conference on Software Maintenance, Bari, Italy, 1997, pp.214-221.

[10] L. Delgadillo and O. Gotel, "Story-Wall: A Concept for Lightweight Requirements Management", 15th IEEE International Requirements Engineering Conference, 2007.

[11] R.S. Pressman, "Software Engineering: A Practitioner's Approach, 4th Ed.", McGraw Hill, 1996 (ISBN 0070521824).

[12] X. Luo, K. Kar, S. Sahu, P. Pradhan, and A. Shaikh, "On Improving Change Management Process for Enterprise IT Services", 2008 IEEE International Conference on Services Computing.

[13] A. Kobayashi, and M. Maekawa, "Need-Based Requirements Change Management",Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS '01), 2001.