Defect Prevention and Process Improvement Methodology for Outsourced Software Projects

Muhammad Faizan, Sami Ulhaq and M.N.A. Khan

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan

Abstract: Software quality has been an important issue since long. There are lots of disciplines related to the quality of software but we have focused towards the defect prevention for outsourcing projects. Nowadays, quality is considered to be the most important aspect which has direct impact on success of the software product. Significant research pertaining to quality of the software and its methodologies has been done earlier. Nevertheless, defect prevention is the key aspect to achieve quality of a product. Many organizations outsource their projects and as a result the quality issues arose. A number of quality models, metrics and standards have been development for the software to run on standalone and distributed environment. However, the issue of quality for the outsourced projects particularly for the projects which are partially outsourced still needs further research. In this paper, we propose a framework for effective monitoring the software quality for both fully and partially outsourced project. The proposed framework consists of two phases: one for the complete outsourced project and second for the partial outsourced project, which are described at length in this research paper. This research fundamentally focuses towards the implementation of defect prevention activities among the distributed organizations working groups.

Key words: Software Quality Engineering • Project Management • Process Improvement • Outsourcing • Defect Prevention

INTRODUCTION

From customer’s perspective, the quality of a product is of foremost importance. Defects are the main cause for degrading the quality of product. The software quality assurance activities aim at producing better quality software products. According to IEEE Standard Glossary of Software Engineering Terminology (1990) [1], “the quality assurance is a sequence of specific actions to give a boost so that a product meets the customer requirements”. Quality assurance phase maintains a balance between customer requirements and product quality. Quality assurance is not simply meant for the software testing, rather it is linked to the overall process improvement of every development phase [2]. There are two types of quality assurance technique: first one is reactive which includes testing to find defects in the product and the second one is proactive which aims at preventing occurrence of defects. The proactive quality assurance technique is also known as defect prevention technique. The main goal of defect prevention is to strike a balance between schedule and cost of the software product by averting the chances of introducing errors into the code. On the other hand, defect prevention can be definite as a progression of quality whose rationale is to classify defects by categorizing their derivation causes and put together efficient counteractive and defensive procedures to avert defects from returning in future [3, 4]. Quality of the product is considered to be achieved when defects are completely removed from a product. According to Suma and Nair (2008) [5], “A defect is a desirable behavior of an application in every consumable aspect.” A number of methodologies regarding implementation of defect prevention activities are described in [6]. Applying the defect prevention methodology also improves the project and organization process due to the implementation of defect prevention activities in all phases of project lifecycle. Defects are reported to a centralized defect database so that by applying causal analysis and defect classification techniques the possibility of occurrence of the same defects does not occur again.
Since technology is advancing at a fast pace, therefore, organization are facing several challenges and difficulties in developing projects by themselves as their IT team members particularly the novice developers find it hard to grab the new technologies. As a result, to maintain balance among resources, schedule and cost, organizations occasionally outsource completely or partially their software development projects. However, when a complete or partial project is outsourced to a low level organization, several issues pertaining to the quality arise. A low level organization is one which is at the initial stage of adopting international standards and improving its processes. The main reason identified for happening of such issues is linked to the fact that only defect detection method is used in most of the low level organizations [7].

To overcome the quality issues in the outsourced projects, we have proposed a framework for improving the project processes through implementation of defect prevention activities for both the developing organization and client organization which is outsourcing the complete or partial project. As a result, the action plans help improve the product quality and organization process.

This paper is organized into six sections. In section II, we define problem statement and discuss key challenges faced in order to produce quality products. The literature review regarding defect prevention methodologies is presented in the third section. Proposed framework is elaborated in the section IV, followed by the implementation of the proposed methodology in a software development company is defined in section V. Comparison analysis is conducted for the validation of proposed framework is described in section VI. Conclusion along with future dimension to this research is provided in the last section.

**Problem Statement and Key Challenges:** Some of the key challenges identified for process improvement are:

- Inadequate research regarding integration of defect prevention activities for the sake of process improvement of the client and software development organizations.
- Applying distributed development approach virtually creates distances between the parties [8], which brings about problems for the management to deal with quality issues.
- Integrating the developing projects/modules in the distributed environment is very much problematic and rework is required most of the time [8].
- Integration of complete or partial outsourced project processes without implementation of required defect prevention activities and process improvement makes it difficult to maintain balance among quality, cost and time.

Defects introduced during software development not only affects quality of the product, but also requires more work to be done to remove them [9]. The delay in detecting the defects can cause serious problems related to quality, cost and schedule, so instead of applying reactive actions the proactive actions like defect prevention should be applied for the process improvement and product quality. Due to the innovation of new technologies, software product development has technically become more complex [7]. Due to lack of technological skills, resources, budget and time constraints, organizations face a dilemma to develop their software projects; and to manage such complexities, the organizations have to outsource some parts of the project or the entire project. To achieve better quality of a product, defect prevention is the key process activities for an organization. When organizations outsource their projects, certain issues like integration of software process improvement arise. On top of it, organizations face difficulties to apply their own standard defect prevention activities for achieving better product quality. As a result, the quality of a product is compromised due to non-implementation of the required defect prevention activities and software processes for the entire or parts of the project.

In view of the above, we propose a framework which consists of two phases. The first phase describes the structure implementation (comprising of framework components and their interaction) of complete outsourcing project by applying defect prevention activities and process improvement. The second phase describes the structure implementation of partial outsourced project by applying defect prevention activities and process improvement.

**Literature Review:** Defect prevention and process improvement are the main factors for producing quality products and considerable research has been done in these areas. The focus of research has been on the improvement of defect prevention activities at organization level, but not at cross-organization level. Therefore, we have proposed a new problem area which is focused towards the implementation of defect prevention work flow for process improvement in different organizations working for outsourced project.
Raninen et al., (2012) analyzed defect database of three different companies which had different tools to support the defect database. Defects are categorized and analyzed through applying different approaches. Work done by Van Moll et al., (2002) [7] focuses on problems arise in low level organization like developing of complex projects, delay in delivery of product, budget issues and rework efforts in the software development lifecycle. Defect density is reduced by applying the defect prevention activities throughout the software development lifecycle. Meng et al., (2006) [10] presented a framework to advance the software possesses by applying defect prevention activities. Marcos et al., (2008) [11] stress on using defect causal analysis technique for defect prevention and proposed DBPI (defect prevention based process improvement) approach for process and product improvement. Chang and Chu (2007) presented an action-based defect prediction approach which analyzes the steps causing the defects in the system and maintaining the defect prediction model for blocking the defects in the software system. Data sampling and undersampling techniques are used in action-based defect prediction approach for sampling of dataset which is helpful for the defect prediction model. Kumaraes and Baskaran (2010) used orthogonal defect classification (ODC) methodology to classify defects and conducted root cause analysis on the classified defects which could help in implementing preventive actions. By virtue of such an approach, repeatedly occurring bugs are controlled and quality is improved. Trivedi and Pachori (2010) [12] also highlighted importance of applying the ODC approach by which the root causes of the software defects are carried out and preventive measures are taken and defect density is reduced. Tiejun et al., (2008) [13] defined the way of tracking bugs in the software development by applying the bug tracing system which is based on ODC. By applying this approach bugs are easily tracked and defect processing is increased. Mujtaba et al., (2011) [14] devised a methodology to measure defect fixation process through the defect fixation efficiency method. The higher value of defect fixation efficiency (DFE) represents more percentage of defects in the software. Suma and Nair (2008) [3] discussed software inspection techniques which required less time in project development lifecycle for ensuring quality of the product. Such an approach also reduces development cost and time. Jantti et al., (2006) [15] highlighted problems related to defect management process which can be very useful to prevent defects from the product and achieve better quality. Ghazarian (2009) [16] discussed a case study on defect introduction system and illustrated the worth of defect causal analysis. However, the above mentioned approaches are proportionately deficient in one way or the other and are related to a particular environment or system in the defect prevention and process improvement progression.

In view of the limitations cited in the literature review, we intend to propose framework for the improvement of software project processes by applying defect prevention activities for the partially or completely outsourced project. The proposed framework can help software organizations to enhance their processes and improve the overall quality of the product.

**Framework Description:** Proposed framework is divided into two phases. Phase I describes the structure for the completely outsourced project process improvement and defect prevention activities flow whose conceptual view is shown in figure 1. Phase II describes the structure for partial outsourced project process improvement and defect prevention activities flow as shown in Figure 2.

**Phase I: Fully Outsourced Project:** Two organizations are presented in this phase, one is client organization which outsources the project and the second is software development organization which is tasked to develop the project for the client organization. First of all, the project which has been outsourced is consulted by the designers of both organizations for the appropriate system design. Once the requirements and the design are finalized, the development of the project is started within the software development organization. Then, a build is created for the software testing, which results in reporting and classification of defects according to the defined classification technique into the central defect inventory. Both of organizations can use this central defect database as per their interest and usage requirements. An analysis meeting will be held by both the organizations to inspect the defect database. On the basis of observations made during the analysis meeting, action teams of both the organizations will suggest proposed process improvement plan. Client organization will adopt the process improvement plan for further project development and for organization process as well. Since improvement in the project processes is an on-going activity due to evolving of new requirement and changes in the business activities, therefore, software development organization will need to adopt a vibrant process improvement plan for the project processes. Nevertheless, such process improvement plans ordinarily remain in continuous.
improvement state. The software development organization can also update the organization process according to their defined rules and procedure. This cycle will run simultaneously to improve the quality of end product and action plans.

Figure 1 illustrates workflow of the proposed framework for the complete outsourced project in which the defect prevention activities suggested by the client organization are applied to the software developing company as discussed earlier. Framework components
and their interaction with each other is provided in the following subsections.

**Complete Outsourced Project:** In case a complete project is required to be outsourced, the top management will be responsible for the solution of problems which surfaces during the process of outsourcing.

**System Design:** Designers from both the organization are responsible for documentation and managing processes related to the complete system design. Generally, three to four persons are tasked for collecting the requirements and design document information.

**Development and Software Testing:** Development of the project starts according to the desired processes, procedures, technologies and resources. Build is created for the complete software testing. Testing approaches and methods are applied during testing and as a result, defects are detected which are reported to the local bug tracking software.

**Central Defect Database:** The detected defects are reported to the central defect database which is in use of both the organizations. The defects are also recorded into the internal defect database of client organization. Defects are classified according to the defined attributes by both the organization. Defects are maintained and classified according to well known ODC methodology which brings ease for the stakeholders to identify causes of defects.

**Causal Analysis Meeting:** As the defects are categorized, so an analysis team consisting of five to six persons from both the organizations conducts the causal analysis meeting in which causes of the occurred defects are identified and proposed actions are presented for the continues quality improvement. Defects are analyzed and examined to prevent them from occurring again in the future. As a result of causal analysis meeting, a report describing causes of defects along with their corresponding action plans is generated which is passed onto the action team.

**Process Improvement:** The action teams of both the organizations are responsible for implementation of the proposed action plans for process improvement. Project processes are enhanced after applying the action plans which are suggested by the analysis team.

**Phase II: Partially Outsourced Project:** Similar to phase I, the phase II also describes two organizations: one is client organization and the second is software development organization tasked with partial development of the project. Some project modules are developed in the client organization and rest of the modules are developed in the software development organization. The modules which have to be developed within the client organization and outsourced modules which have to be developed in software development organization are firstly consulted by the designers of the client organization for the system design. Once the requirements and design is finalized, meeting with the software development organization is convened and development of respective modules is started in both the organizations. Then a build is created within both the organizations for the unit testing which results in reporting and classification of defects according to the defined classification technique into the central defect database. Afterwards, the modules/units are integrated for the integration testing. Problems and defects found during the integration phase are also reported into the central defect database. Once the integration testing is successful, the build is created for conducting complete software testing. Defects found at this stage are also reported into the defect database. Both of organizations can use the central defect database for their interest and usage. An analysis meeting will be held by analyst teams of both the organizations to discuss the errors stored in the defect database. By the help of causal analysis of the defects, action teams of both the organizations will suggest process improvement plan. Client organization will adopt the process improvement plan for further project development and for updating the existing organization processes as well. On the other hand, software development organization will adopt the process improvement plan for the module development processes which is a continuous improvement state. Now this cycle will be run simultaneously to improve the quality of end product and action plans.

Figure 2 demonstrates the work flow of the defined framework for the partial outsourced project in which the defect prevention activities suggested by the client and developing organization are applied for the improvement of working processes to enhance quality of the product as discussed earlier. Components and interaction between them is provided in the following subsections.

**Partial Outsourced Project:** At initial stage the partial project or sub modules due to inadequacy of resources or limitation of technology or any of reason is outsourced to the desired organization by the decisions of top
management. The top management will be responsible for the solution of problems which comes in between outsourcing the partial project.

**System Design:** As elaborated in section 5.2, the designers of both organizations are responsible for the documentation and managing the processes of system design related to the partial project or sub modules. A team consisting of 3 to 4 persons is generally tasked for collecting the user requirements and the design document information in their own organization.

**Development and Unit Testing:** Development of the partial project or sub modules is started within both organizations as per the desired processes, procedures, technologies and resources. Build is created for the unit testing within both organizations. Testing approaches and methods are applied while unit testing of sub modules as a result defects are detected which are reported to the local bug tracking software.

**Integration of Units/Modules:** After unit testing, the modules are integrated. Integration team consisting of 2-3 persons is responsible for the integration of units or modules.

**Integration and Complete System Testing:** After integration of units or modules, the client organization undertakes the cycle of integration testing. As a result of integration testing, more defects may be detected which are required to be reported to the local bug tracking software. The complete cycle of system testing is executed after the integration testing which off course brings out more defects and are reported to the local bug tracking software.

**Central Defect Database:** The detected defects in all phases of testing are reported to the central defect database which is in use of both the organizations. Defects are classified according to the defined attributes by both the organization. Defects are maintained and classified according to well know ODC methodology which brings ease for the stakeholders to identify the cause of defects.

**Causal Analysis Meeting:** Agenda for causal analysis meeting is same as that of the complete outsourced project. So the same steps as involved in this process as discussed in section 4.1.5.

**Process Improvement:** Here the process improvement steps are same mentioned in section 4.1.6.

**Framework Implementation in a Software Company:** We implemented our framework in both medium and low level organizations. The medium organization that outsourced the software development project consists of project managers, team leads, software engineers, database administrators, quality assurance team and support staff. Medium size organization also deals with the product development and provides support for the multimedia devices. The low level organization which was tasked for developing the software project consists of one project manager, two team leads for software development, one team lead for server side development, six software engineers and two quality assurance engineers.

The project was outsourced due to lack of resources and expertise in android technology at the client organization. The scope of outsourced project is related to the android development platform for the IPTV (Internet Protocol Television) set-top box technology through which television services are delivered using the Internet protocol. Since the project was outsourced due to shortage of time as well, the development organization opted for “code and fix” software development model in which developers immediately begin producing code and the testing phase starts immediately after development. The processes used by this organization were not well managed so the quality issues aroused during the software development cycle.

Some of the issues that occurred are defined below:

- Time and cost overrun
- Numerous bugs were introduced into the code
- There were hurdles in carrying out process activities due to delays in prerequisites of the process activities
- Several milestones were slipped

As a result, to address the upcoming issues, an analysis meeting was held in the client organization and team members from both the organizations participated in it. Root cause analysis was carried out in which the causes of these issues were identified. Some of the causes which were identified are listed below:

- Requirements uncertainty
- Miscommunication
- Inappropriate software development model
- No implementation of defect prevention activities
- No process improvement
- Less concentration towards project quality
Figure 3 shows the causes for deprivation of the products quality. After analyzing the causes of defects, it was decided by both the organizations to implement the appropriate processes, methods and technologies. According to the current scenario, our proposed framework fit for such a situation. The main causes included requirements uncertainty and miscommunication, which were removed by applying the system design process. The purpose of system design process is to provide a platform for interrelating requirements of software product as described in section 4.1.2. Keeping in view the nature of project, it was decided to implement waterfall model as a key software development model so that a number of issues could be resolved. Before implementation of the proposed framework, defect prevention activities were neglected. By employing the defect prevention activities, bugs reoccurring problem was reduced as argued in section 4. Central defect database feature allowed both organizations to analyze the defects mutually and helped preparing proper action plans. As a result of the execution of action plans, software development processes were improved. Through root cause analysis methodology which is also a part of our proposed framework, many causes for degradation of product quality were removed. Though the required time to build the software project exceeded because accomplishment of framework took more time and efforts to maintain stability of processes for both the organizations, but as a result of implication, the processes of software development improved. By implementing the proposed framework, most of the issues faced by both the organizations were overcome, which eventually resulted in overall improvement of project quality.

DISCUSSION

Main goal of this research is to propose a framework to improve the software development processes for the outsourced projects through applying defect prevention activities across the distributed working groups. We have carried out a meticulous comparison analysis to validate our proposed framework. For this purpose, best cited papers are analyzed and compared with our proposed framework as discussed below.

Kannabiran and Sankaran (2011) [17] categorize and appraise the determinants of software quality for the outsourced project development. The scope of research is focused on process improvement through identifying the key factors which affect the software quality, whereas our proposed framework supports the process improvement through applying defect prevention activities in the outsourced project development. The study of Kannabiran and Sankaran (2011) [17] falls short of the implementation of defect prevention which is main activity to achieve better software quality. Choi et al., (2007) [18] focus on outsourcing of software testing and inspection activities which have been studied in the context of defect detection activities through Analytical Hierarchy Process. Their study only describes criteria for outsourcing of testing and inspection activities but lacks the overall process improvement and applying defect prevention activities as defined in our proposed framework. To sustain quality, Jacobs et al., (2004) [19] presented a framework for decreasing the number of defects found in the products developed by the distributed teams. Defect prevention methodology (root cause analysis) is used in their research for finding the causes of defects and associating the defects to risk factors. As compared to our proposed framework, the methodology used in their work only finds out causes of defects in the distributed environment but do not suggest recommended actions to improve the processes for the overall development lifecycle. Whereas, our proposed framework provides a mechanism for the implementation of defect prevention activities for outsourced projects. Jayathilake et al., (2011) [20] proposed a framework which includes verification of quality attributes of a product developed through outsourcing. The focus of research is limited to the verification of quality attributes, whereas process improvement and overcoming quality issues.
through implementation of defect prevention and defect detection activities require due attention. Abdullah and Verner (2012) [21] discussed risks associated with the outsourcing of strategic software development through identifying the risk factors. But, a pressing requirement is to device mechanism to remove risks and improve quality as outlined in our proposed framework. In this regard, our framework provides complete structure for the removing of risks associated with the outsourced project through implementation of defect causal analysis technique. Chou and Chou (2010) [23] highlight the issues related to risk supervision in the fully outsourced projects, but issues related to organizational process improvement and partial outsourcing of development projects is not covered. Kumaresh and Baskaran (2010) [4] used orthogonal defect classification (ODC) methodology to classify defects and carried out root cause analysis on the classified defects which could help in taking preventive actions. Due to high capability of such an approach, repeatedly occurring bugs are controlled and quality is improved. Meng et al., (2006) [11] presented a framework to continuously improving the software possesses by implementing defect prevention activities within an organization. The framework is helpful to adopt the process improvement within the organization, but does not encompass mechanism to adopt processes improvement for the outsourced development.

CONCLUSION

In this research study, we have proposed a framework to address key challenges related to the quality perspective faced while outsourcing complete or partial project. The major causes that lead to these challenges are identified as requirements uncertainty, miscommunication, absence of implementing defect prevention and process improvement activities beforehand. We have devised a framework which basically focuses on removing the causes of challenges faced during undertaking of outsourced projects. Our proposed framework consists of two phases. The first phase describes implementation for the complete outsourced project and the second phase describes implementation of partial outsourced project. Promising results were achieved by implementing the proposed framework within the software development company. Currently our proposed framework is limited to oversee two organizations (i.e., client and software developing organization). As a future dimension to this work, we intend to add additional features in the framework to account for participation of multiple organizations and substantiate global software development for the outsourced projects.

REFERENCES


