Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 2, Issue 3, pp: (32-37), Month: September-December 2015, Available at: www.noveltyjournals.com

Role of Defect Prevention Techniques Vs Defect Detection to Improve Software Quality: Critical Analysis Summary of Defect Preventive Approaches

[1]Nasir Jamal, [2]M Zulqarnain, [3]M Waqas Boota, [4]Saima Khan, [5]S M Abrar Akber

[1, 2, 3, 4, 5] Department of Computer Science, Virtual University of Pakistan, Lahore

Abstract: **In the world of software development the statement applies "Even a little prevention is better than spending a large in cure". Software is developed by the programmers. As human being a programmer be major cause of defects in software along with many other type of defects caused by lack of domain knowledge, poor requirements analysis, design flaws etc. which can lead to the failure of a software project. Due to lack of meeting the standards and quality most software projects fail to achieve desired level due to which many defects in requirement gathering, design and coding phase introduced. One of the important reasons of this trouble is be deficient in following appropriate defect prevention arrangements during the development stages of software design process. Having direct influence on software quality defect prevention is a systematic critical segment which if ignored may lead to the poor quality product becoming the cause for project failure. Defects in software can involve the both financial cost in detection then removal and time. So defect prevention is preferred then defect detection after which fixing is needed. The cost spent on activity of discovering and fixing the software bugs is the most expensive among the activities of software development. Moreover, the software project is at high jeopardy in cost and time domains if software defects transferred in final approval testing step of project development life cycle. A reasonable cost can be saved for discovering and eliminating defects by putting a little effort on quality assurance activities to get defect free high quality software product. It is necessary to study, observe and scrutinize the details of detected errors and defects to get in depth understanding of success of software process. Due to newer methodologies and process improvements how defects can be eliminated is studied. Due to lack of meeting the standards and quality most software projects fail to achieve desired level due to which many defects in requirement gathering, design and coding phase introduced. These defects unavoidable delay the development operations of software in smooth fashion. One of the important reasons of this trouble is be deficient in following appropriate defect prevention arrangements during the development stages of software design process. Having direct influence on software quality defect prevention is a systematic critical segment which if ignored may lead to the poor quality product becoming the cause for project failure. The critical analysis of existing defect prevention strategies and trends used by them for defect identification is the scope of this paper.**

Keywords: **DBPI- Defect prevention based process improvement, ODC- Orthogonal Defect Classification.**

I. INTRODUCTION

To improve the software quality, defect prevention is another quality assurance activity focusing on the production of higher quality products. Software quality has multiple meanings one of which measuring how number of defects in software product. Improved quality of software is kept by minimal defects. The software projects do not utilize sufficient

defect prevention methods in effective way to fail to fulfill the required level of quality standard and user acceptance. In different stages of software development life cycle defect prevention can be performed. Key activities included in defect prevention are defect prevention plan, defect casual analysis, defect inspection, defect classification, defect detection etc. Defects can relate to any flaw or inaccuracy in software process [1]. Usually the word fault, flaw, bug, error or failure referred as defect [2]. The phrase failure and fault can additional be described as 'act that push to false result', incorrect decision due to erroneous interpretation of existing information and deficient in capability to meet up the projected performance. The purpose of defect prevention techniques is to keeping out the defects to occur in advance [3]. The term defect prevention alternatively also described as process of quality improvement by discovering defects during root cause analysis and construct useful corrective and preventive procedures to reduce them from persisting in future [4]. Likewise Root Cause Analysis is performed using casual analysis [5] which is major technique used for risk identification linked with individual type of defects. IBM design the Orthogonal Defect Classification technique usually used for finding and classifying the defects in software [6]. Performance and Continuous Re commissioning Analysis tool used for formation, reviewing and recording defects found in defect prevention process is recommended in [9]. To prevent the defects there is no ideal and perfect solution due to heterogeneous nature of software products.

Software defects or bugs or errors as defined by the software experts are "imperfections in software development life cycle that would be the origin to lead the software towards failure in term of not meeting the desired expectations [1]. A massive number of defects during the software product development process would appear in. The defects can be removed in remaining development process that injected in initial stage is a misleading believe [8]. During the whole development process till final testing and implementing step defects may occur. So for the improvement of software process quality, defect prevention is become necessary. The process of quality improvement aims to recognize the ordinary causes of defects is to be termed as defect prevention and to prevent similar typed defects to appear again by changing related process [2]. The overall resources, costs and schedule reduced by increasing the software product quality in defect prevention process and quality-cost-schedule balance of a project is to be ensured.

The intention of defect prevention strategy is to categorize by recognizing the defects in the initial development life cycle and not allow them to reappear. Software defects are recognized for analyzing them to improve the software product process quality. To prevent re occurring same defects in new projects root causes are identified in preventive mechanism to eliminate these defects which may result in improvement in quality. Software development life cycle quality being continues to improve in this cycle. Broad view of defect prevention approaches followed in project development make available in the scope of this paper.

The paper is arranged as Introduction in Section I, literature Review and related work in Section II, critical analysis of defect prevention techniques discussed in literature review are summarized in Section III, Proposed Future work in Section IV and Conclusion in Section V followed by References.

II. LITERATURE REVIEW AND RELATED WORK

In [1] express the significant approaches of defect prevention by underlining that 15% of overall time to complete the project is required by software inspections and is productive quality fact to reduce the time and cost of development. Orthogonal defect classification and defect removal efficiency ratio are also important techniques to prevent defects. The software company can increase improvement in quality standards and providing best environment through following the famous standards and training. The importance of this research study is that different defect preventive approaches are compared to provide careful analysis of organizations.

At the initial phases of project do necessary preventive procedures to control and reduce the defects [3]. To determine the defect causes and taking suitable corrective measures Casual Analysis is a general technique. Conversely, once defects appear in access, it become hard to take preventive measures and handle them. So for the improvement of software process and defect prevention, defect detection is important. Using association mining and rules the study of defect prediction technique is carried. The actions that can become the cause of enormous defect rate be predicted by applying this approach along with classification decision tree performing association rule mining on defects to software development life cycle. The data mining approach used to determine the common patterns, causal structure and association in big dataset is termed as association rule mining.

The previous research in defect prevention paying attention on predicting defects and eliminates by utilizing the lot of resources in debugging process to in time complete the project decided upon the required resources and team size. Majority of software development organizations formulate their approaches for defect prevention with the arrival of software development life cycle. For defect prediction and prevention many research studies were carried out. Bug tracing system to trace the defects were introduced in [6] becomes famous for accurately identified defect tracking improvements and low cost. HP and IBM projected the defect classification strategies proposed in [9]. Based on modes, defect-origin and types dimensions based approach is by HP and orthogonal defect classification by IBM. How defect prevention feedback in later iterations provided by defect analysis originated in first iteration focused in [7] which leading to improvement in productivity and quality. The ideology that defects identification in the initial phase of development life cycle become the factor of defect prevention in afterward steps of life cycle and focusing on suggesting the appropriate action to prevent functional defects worked in [1].

Defect data is classified using defect analysis procedure and ODC [4]. Preventive measures are applied by conducting the root cause analysis based on defect analysis. By this technique improvements are observed in enhanced productivity with reduced defect density. While the defects concerned to requirements and coding are determined using ODC approach. The outcome shows the role of defect prevention is fundamental in development lifecycle of project to decrease the density of defect recurring which leads to improvement in software product quality by reducing cost, rework and effort. Defect density found in the literature measured as: Defect Density = No. of defects / size in kilo line of code.

In any organization the improvements of software process are accomplished by defect casual analysis [5] but still defect rate not reduced more than 50% by DCA. To attain higher quality it is recommended to implement DCA by following steps.

i. Defect samples
ii. Defect classification
iii. Summary of repetition of similar defect
iv. Identification of causes
v. Action planning
vi. Documentation

Defect prevention based process improvement (DBPI) methodology introduced which is capable to improve the quality in the study.

In software organizations implement ODC approach for defect prevention which provides better means to detect and classify defects [6]. To determine root causes for preventive action defect classification ODC is used by the IBM and HP. Defect qualifier, defect type and defect trigger are the attributes for applying ODC approach. The analysis of implementation of ODC results in software development life cycle predicts there is noticeable reduction in defect detection and cost of detecting defect. The study contributes that how the quality of software development life cycle is improved by implementing Orthogonal Defect Classification processes.

Signify the importance using of ODC to prevent and classify defects [7]. The development time and cost of the project can be reduced while quality of the product is improved by applying the defect prevention technique. Root cause analysis is performed to prevent defects from reoccurring after the classification of defects. Every defect is classified into managerial and technical type attributes by the ODC technique. Defect detection, classification, cause analysis, learning, result verification and measurement are steps included in defect prevention workflow. Some causes seems to be incorrect requirements and some other as missing requirements after implementing root cause analysis.

A defect investigative technique based on ODC to prevent defect is defined as defect tracing system in [8] and workflow presents improves the correctness of defect identification. Verification, consequences, development and reason are the primary attributes of ODC based Omni directional analysis which is proposed by using ODC concepts and activities of workflow. Bugs are easily traced using this reputed Bug tracing System technique. Submitted problem by initializing the process by the Bug Tracing System to record bug related attributes then verification is done by the BTS workflow activities and as the result, up to 40% improvement in defect processing may be achieved which enable the management to evaluate the software quality progress.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 2, Issue 3, pp: (32-37), Month: September-December 2015, Available at: www.noveltyjournals.com

An approach of defect prevention and detection recommended in application development tool know as **PACRAT** – performance and continuous re-commissioning analysis tool in [9]. Automated test tool is constructed by implementing the defect prevention actions such as defect tracking, root causes of defects identification and suggesting the solution to these identified causes earlier then the tool development. For whole defect detection activities a database is formed for syntax checking and validation and verification which makes the PACRAT an excellent automated tool for implementing the defect prevention and detection measures.

A framework for defect prevention which reduced the defect density significantly in KLOC (kilo line of code) is proposed in [10]. Data analysis method is used to describe the project planning and process tracking and to obtain higher level of quality defect prevention procedures are performed at various stages of development life cycle. The defect prevention procedures contain preparation, defect type definition, process of defect prevention and support to prevent defects. The improvement environment and better product quality in the organization is achieved by defect prevention activities beneficial for stakeholders, the client and organization.

Need for Defect Prevention:

In software project important activity is the defect prevention. Rework and defect detection is focused by project team in majority of software companies while ignore the defect prevention component. It is recommended to take necessary steps in the initial phase of project to prevent defects being injected in the product. Such preventive activities in quality assurance process has a little cost but have considerable advantage of saving overall cost comparing to defect repairing later. So time, effort, cost and required resources can be minimized by examination of defects using quality assurance activities in the start. Defect prevention better make possible by knowing the injection processes and methods in the product which helps to increase the overall productive capability of organization and quality of software product.

III. CRITICAL ANALYSIS SUMMARY OF DEFECT PREVENTIVE APPROACHES

The critical analysis of these different defect preventive approaches is summarized by describing the strengths and weakness as:

Approach / Strategy	Ref No.	Weakness	Strength
Inspection	[1]	Not describe any tool to apply defect prevention approach	It compares different defect prevention techniques to provide careful organizational level analysis
Association Rule Mining and Action Based Defect Prediction (ABDP)	[3]	Useful in case of large number of defect	Best results for software process improvement can be achieved by combining both ABDP and Association rule Mining
RCA and ODC	[4]	Not suggest root cause analysis for large size complex projects	Consider comprehensive analysis of organizational level data for whole development life cycle to prevent defects
DCA	[5]	Also not do comparison among different approaches and limited to cause effect techniques only	The implementation of DCA introduced DBPI approach based on defect cause analysis.
ODC	[6]	Consider only in one direction – Omni directional	The evaluation and improvements in project development life cycle done by ODC to increase the quality of product
RCA and ODC	[7]	Does not recommend more improvements in defect rate reduction	Introduced as work flow is the excellent practice for defect prevention
BTS- Bug Tracing System	[8]	Case study may not be considered using BTS technique	Improvement in defect processing is noted up to 40% by implementing BTS to enable the management to evaluate quality on each module of software

PACRAT	[9]	Project development life cycle and development automated software complex tool is not considered	About the implementation of defect detection measures and defect prevention PACRAT is better automated tool.
Defect prevention process improvement	[10]	Could not suggest the complexity level of project by applying the defect prevention measures	Software product quality is improved by removing the defect density due to defect prevention activities which cause to environment increase of the organizational level.

Process Improvement Work Flow:

Defect Identification:

The uncovered defect can be identified by activities like code inspection, reviewing graphical user interface, reviewing design, unit and function testing etc planned in advance performed at different development cycle steps. After the identification of defects it is easy to classify by orthogonal defect classification.

Defect Classification:

One of the most popular approaches used for defect identification is the orthogonal defect classification up to first level which instead considering the individual defects grouped them category wise into types. Defects are classified at two points in term of time by the ODC [1].

Time while defect detected first:

Time while defect fixed:

Administrative and technical attributes together provide necessary information pattern to analysis the root causes of defects are used as methodology by ODC to classify each defect [8]. Better tracking action plan together with ODC enables to attain high scale defect minimization. To get understanding and analyzing the defects to save effort and time, large scale projects requires identifying the defects in depth while the medium and small scale projects are classified by ODC at first level. Different defect types like requirement defect, design defect, coding defect including logical also and documentation defects are classified in first level of ODC after which defect analysis is conducted.

Defect Analysis:

To improve the quality regularly defects as statistics is used by defect analysis to categorize defects and susceptible defect root causes are identified in direct process improvement activities. The role of root cause analysis in software defect analysis is important to begin action for the elimination of source of root causes of defects after the identification of defect root causes. The defect analysis is performed by human based on the strong analytical skills and give feedback to programmers in order to improve the productivity of company and the quality of the software together [8].

Defect Prevention:

The aim of defect prevention is the root cause identification of defects to avoid them reappearing in any later stage is the very important task in the software project. The actions included in defect prevention are analyzing the previous occurred defects to suggest appropriate action from stopping that type of defect to occur in future and can be implemented to improve the quality of software process on software life cycle phases [4].

Process Improvement:

The improved software development life cycle documents and processes based on the actions suggested in defect prevention phase which are applied through changing the software development life cycle and modifying the existing software quality manual. This revised preventive strategy is implemented carefully on next projects to improve the quality of software project by avoiding identified defects and reducing the overall cost, time and efforts needed to perform in defect identification and fixing at all phases.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 2, Issue 3, pp: (32-37), Month: September-December 2015, Available at: www.noveltyjournals.com

Root Cause Analysis:

The process of determining the activity responsible for the root causes of defects and discovering the corrective actions to minimize or eliminate the defects is performed in Root Cause Analysis which is based on concepts as

1. To improve quality by defect reducing: Root Cause Analysis should guide to implement modifications to stop errors in development stages and guarantee the detection of defects in the start if re appears.

2. Third party Proficiency using: As the defect root cause analysis is based on the critical analytical skills of individuals so for better results the expertise of third parties expert in finding the root causes of defects will be utilized along with the people of companies who understand what is wrong.

IV. PROPOSED FUTURE WORK

A framework is proposed for defect prevention process by combining the strengths and covering the weakness of critically analyzed techniques as a future work to reduce the effects of defects and bugs and to gain the higher quality defect free software product.

V. CONCLUSION

Defects and bugs in the software products are ordinary observable facts which needed more considerations to stop them from occurring products in advance because minimum cost and efforts are required to fix them at early stages. A critical review of various approaches used for defect prevention is suggested in this paper to propose a defect prevention framework tool as a future work to improve the quality of software products by reducing the defect density by fixing them at early phases of project as time, effort and cost required to fix them at later stages increase the overall cost and time schedule of the project. To get the high quality software product implementing defect preventive action is important priceless investment which increases the capability of software programmer to learn from the others as well as his own identified mistakes occurring during at different stages of software development life cycle. The advantages of defect prevention are huge some of which to reduce cost, rework efforts and development time and increases the reputation of development company which results in high esteem of development team, increases client satisfaction so improve quality by decreasing cost. The execution of first level orthogonal defect classification to identify and grouped defects categorically is proposed in this research study. To improve the software product quality and getting the thorough understanding of defects implement classified defects in ODC to next level which give better defect prevention ideas. It is considered that software inspection is a useful, general and simple approach to prevent defects. Casual analysis technique is normally used for defect identification. But innovative defect prevention approaches stresses on classifying defects like orthogonal defect classification (ODC) and ABDP are getting fame.

REFERENCES

[1] Nair and Suma, “Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels,” 2008.

[2] D. Zubrow and B. Clark, “How Good Is the Software: A review of Defect Prediction Techniques,” 2001.

[3] C.P. Chu and C. P. Chang,” Defect prevention in software processes: An action based approach”, 2007.

[4] R. Baskaran and S. Kumaresh, “Defect Analysis and Prevention for Software Process Quality Improvement”, 2010.

[5] D.N. Card and M. Kalinoski, “Towards a Defect Prevention Based Process Improvement Approach”, 2010.

[6] S. Pachori and P. Trivedi, “Modeling and Analysis of Software Defect Prevention Using ODC”, 2010.

[7] A. Shenvi, “Defect Prevention with Orthogonal Defect Classification”, 2009.

[8] F.C. Bin, Z. Leina and P. Tiejun, “Defect Tracing System Based on Orthogonal Defect Classification”, 2008.

[9] E. Bean, ” Defect Prevention and Detection in Software for Automated Test Equipment”, 2008.

[10] A. Sontakke, H. Xiaoyuan and M. Li, “Defect Prevention: A General Framework and Its Application”, 2006.

[11] Mukesh soni, ‘Defect Prevention: Reducing cost and improving quality”, 1997.

[12] A Shenvi , ”Defect Prevention with Orthogonal Defect Classification”, 2009