ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2012

Defect Handling In Software Metrics

Arpita Mittal[1], Sanjay kumar Dubey[2]

Department of Computer Science and Engineering, Amity School of engineering & Technology, Amity University, Noida[1,2]

ABSTRACT: Defect Handling is one of the major and important activity involved in the software various software projects. Defect Handling basically includes identification and taking actions for defect prevention for improving the software quality. Earlier defect identification in software projects will make defect removal and prevention much easier. Software engineers generally take experiences from the past and prevent the defects from re-occurring. In this research paper our first section is the introduction of software defects, second section is the introduction of defect handling life cycle models, third section is use of Cost Constructive COQUALMO model for defect handling, fourth section is conclusions and last section is future scope.

Keywords: Defect, Defect Analysis, Defect Prevention, Defect Prediction, Root Cause Analysis

I. Introduction

Software Defect is termed as **"imperfections in the software development process that would cause that software to fail to meet the desired requirements".** A defect generally represents the undesirable aspects of the software quality. Generally we are concerned with three types of defects artifacts in this research paper and they are: Requirement defects, design defects and coding defects. [2].
defects occur during all the phases of the Software development life cycle. Hence defect prevention is very essential part of Software development Life cycle for improving the Software Quality.
Defect prevention firstly involves identification of defect, and then modification and changing the relevant processes, preventing the re-occurring of the defects in the development process. As early as defects are identified in the development process, the more smoothly the development process progresses.
For improving the quality of the Software process it is necessary to identify the defects from the given set of projects at the first step, then it involves classification and analysis of the pattern and after that it involves elimination for prevention of defects.[1].

II. Work flow Stages Defect Handling [2][1]

2.1 Defect Identification OR Defect Detection in Software Process

Defect Identification is the first activity involved for improving the quality of the Software Process. It is widely used in many of the Software projects, for discovery of the Software Defects, then documenting them for improving the quality of the Software product.

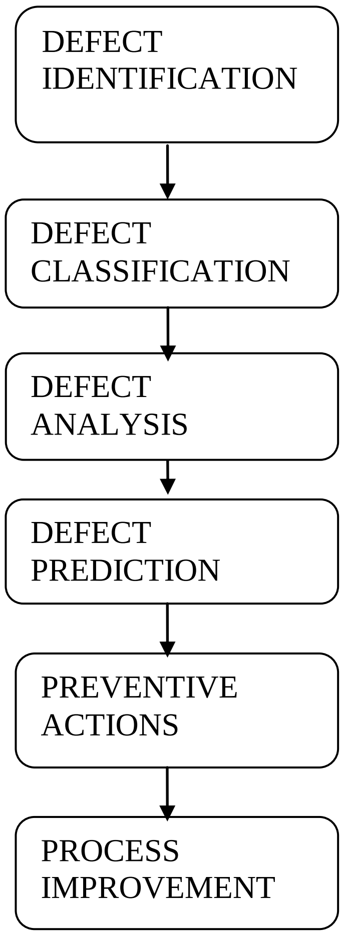


Fig.1

Variety of testing techniques used to identify the various types of the Software defects involved in the Software development process, will may involve either functional or non-functional domains. Hence in software development process firstly defects are discovered and then are documented. Output of Defect document information will be the input for defect analysis technique.

2.2 Defect Classification [2]
ODC classifies defect at two different points in time: One is Opener Section, where the defect were firstly investigated and second one is Closer Section, where the defects are fixed. For Small Sized and Medium Sized Projects defects are classified to first level of ODC to save efforts and time. For larger projects defects are deeply understood and analyzed. Firstly for all types of projects defects are firstly investigated and then they are fixed.[4]. Action planning and tracing helps in achieving the degree of defect reduction and cross learning.

2.3 Defect Analysis [1]
By the term analysis we meant the identification of the root cause of the defect and then further devising the solution to overcome the defect in further development process which will be further useful in improving the software quality and productivity of the software project. Some of the defect analysis techniques such as Fish Bone Analysis, Defect Classification and using defect taxonomies and the Root Cause Analysis (RCA). RCA goal is to first identify the root cause of the defects and then initiating actions for the defect elimination.

2.4 Defect Prediction Technique
By the term defect prediction technique we mean to identify and prevent the defect causing failures before occurring.
This is done by first gaining the experiences from the earlier Software Projects by Software Engineers and then identifying the root cause for the defects and then eliminating the causes.

Some Defect predictions models are COQUALMO model which we have taken and expanded
further in this research paper and the second one is mining defects using ODC. Identification of Defects at the early stages so that wastage in process and product development can be eliminated. And Cost efficiency which involves meeting the deadlines and leading to process improvement for many organizations.

2.5 Defect Prevention
By the term Defect prevention we mean that gaining the experience from the past projects by the software engineers and identifying the cause of Defects and further taking the corrective measure to prevent the defect re-occurrence [2].
Defect Prevention is one very important activity in the Software project thereby further improving the quality of the software project.
2.6 Process Improvement
By the term Process improvement we mean the continuously working for improvement for the quality of the software process. Process Improvement meant that following preventive actions for software improvement and then further taking actions for further improvement of quality.

III. Case Studies

COQUALMO-Defect Prediction Model [3]
COQUALMO is a Constructive quality model for the defect prediction density of the Software development. COQUALMO which basically predicts the defect density where defects conceptually flow from one phase to another thereby making larger defect and leading to project chaos [2]. Defect flow from one tank to another tank, thus making larger defect introduction pipes and then defects are removed with various defect removal pipes.
COQUALMO which is a defect prediction model is basically a two-step process which involves in defect identification (DI) at the first step and the Defect Removal (DR) model in the second step thereby leading to the defect improvement of the software quality. Hence two steps of COQUALMO are as follows:

3.1 Defect Identification OR Defect Introduction Model (DI) [3]
This model typically results in the development of various phases of software development life cycle models. There are basically three types of defects which mainly occurs during the development of various phase of software development life cycle which are as: Requirement defects, Design Defects, and coding Defects. For Defect identification we can use Defect. Introduction Range which is particularly the ratio between the highest and the lowest defect identification range. If the defects are identified in the earlier phases of the software development, the better will be the Software Development.

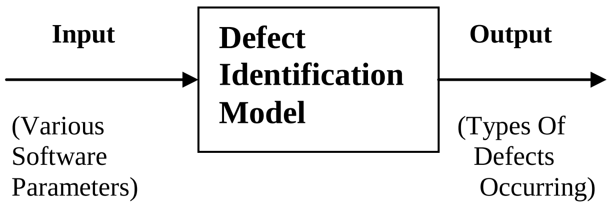

Input
(Various Software Parameters)
Defect Identification Model
Output
(Types Of Defects Occurring)

Fig. 2

3.2 Defect Removal (DR)
By the term defect removal we mean removing the defects such as requirement defects, design defects and coding defects which would typically results in the software failures. Defect removal basically results in the estimation of defect removal by the defect removal activity.

There are basically six orthogonal profiles for the defect removal which are very low, low, nominal, high, very high and very very high.
Very low, low profiles basically requires no peer reviews, no testing and simple compilers, nominal profile requires well-defined sequences, unit and integration and system testing and some more extensive compilers,High,very high and very- very high typically requires formal reviews and procedures, highly advanced testing tools and more formalized compilers.
For Very low profiles:

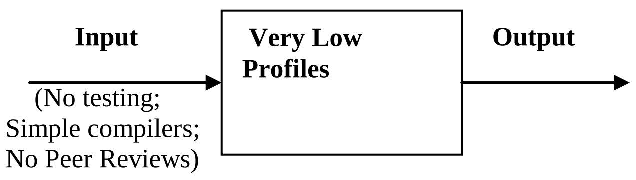


Fig. 3A

For Nominal Profiles:

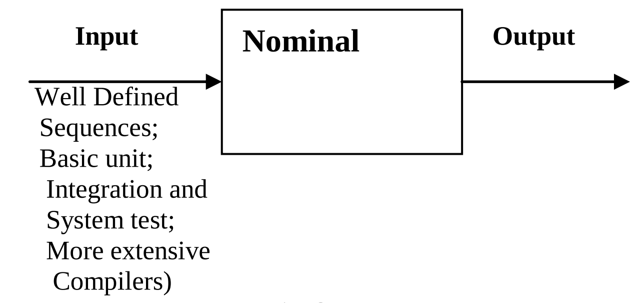
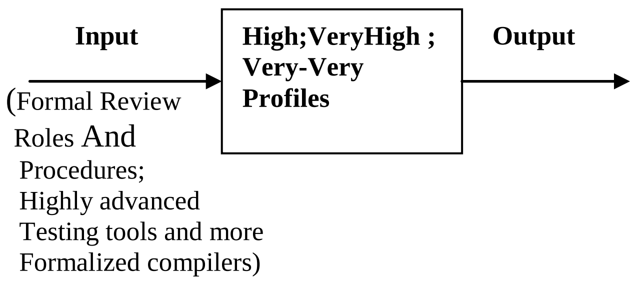


Fig. 3B

For High, Very High, and Very Very High Profiles



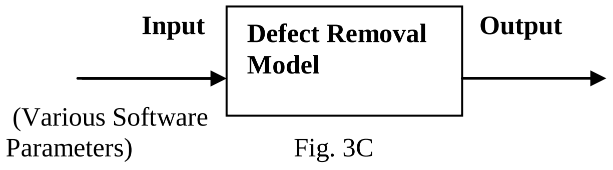


Fig. 3C

Basically three types of profiles occurs in the defect prediction in software modules [4]
Prop Minor = minor defects/total defects delivered;
Prop Major = major defects/total defects delivered;
Prop Extreme = extreme defects/total defects delivered.
Minor: It usually means small, a minor defect occurring in software projects does not affect the software much and doesn't make software unusable.
Major: It means large or more major defect occurring in software projects and making the application become unusable.
Extreme: It causes software totally unstable. A failure in any part of the application totally results in unstable software.

4. Conclusion

In this research paper we have first studied about the various types of defect techniques and then we have undergone through the survey of COQUALMO cost constructive model which is a two-step software defect prediction model for improving the software quality. Earlier we identify the defect introduced in the software, the better software will results. Hence, identifying defects at the earliest software development is very useful and leading to the prevention of software defects which will lead to failure.
The life cycle of defect identification consists of defect identification, defect classification, defect analysis, defect prediction, preventive actions and process improvement.
We have studied three techniques of Defect handling i.e. Defect Detection Technique, Second Defect Analysis Technique, and Defect Prediction Technique.[1]
We have work flow stages which consists of Defect Identification, Second action is Defect Classification, and Third action is Defect Analysis and Defect Prevention.[2]
But in this research paper we have used a Process Improvement model which includes Defect Identification, Second Action is Defect Classification, Third Action is Defect Analysis, Fourth Action is Defect Prediction and fifth action is Defect Prevention, and final Action is Process Improvement. Using these six work flow stages of Defect Handling provided us the added advantage for improving the quality of software projects, and follows a systematic approach, as in first phase we are able to identify the defect involved in the software, second action classifies Defects into opener section and closer section, third action that is Defect Analysis which identifies the root cause of the defect and then further devise the solution to overcome the defect in further development process, fourth action is Defect Prediction technique we mean to identify and prevent the defect causing failures before occurring, and is Defect Prevention Technique which means that gaining the experience from the past projects by the software engineers and identifying the cause of Defects and further taking the corrective measure to prevent the defect re-occurrence and finally action is for process improvement.
After that we have used COQUALMO i.e. cost constructive model as the case studies for demonstrating the fact of defect prediction technique.

But our Work Flow stages that we used in this paper are more complex, further increasing the number of stages of Defect Handling.

5. Future Scope

Since defect can cause malfunctioning in the software projects leading to software failures, so defect prediction is mainly necessary ,which is one of the important phase in the development of software development life cycle(SDLC).In this research paper we have used COQUALMO which is a two-step software prediction model, thereby improving the software quality. In future scope we can add various aspects for defect handling in various software projects. We can compare defects with various other techniques such as Genetic algorithm, neural network, fuzzy logic, decision tree for adding feature for handling defects in software projects.

REFERENCES

1. A Defect Prediction and Analysis Using ODC Approach in a Web Application: *Pranayanath Reddy Anantula and Raghuram Chamarthi(Tejoraghuram), ISSN:0975-9646.*
2. Defect Analysis and Prevention For Software Process Quality Improvement-*Sakti Kumaresh and R Baskaran, VLOUME 8-NO.7, OCTOBER 2010.*
3. Constructive Quality Modeling for Defect Density Prediction:COQUALMO-*Sunita Chulani, FAST ABSTRACT ISSRECOPYRIGHT 1999.*
4. A Prediction Model for Functional Defects in System Testing using Six Sigma: *Muhammad Dhiauddin Mohamed Suffian and Suhaimi Ibrahim,VOLUME 1 NO. 6 SEPTEMBER 2011.*
5. Predicting Defect Types in Software Projects:*dr Lukasz Radlinski,Institute of Information in Management,Faculty of Economics and Management,University of Szczecin.*
6. A Machine Learning Based Model For Software Defect Prediction:*Onur Kutlubay,Mehmet Balman and Dogu Gul,Ayse B.Bener,Bogazici University,Computer Engineering Department.*
7. A Critique of Software Defect Prediction Models,*IEEE TRANSACTIONS ON SOFTWARE ENGINEERING,VOL.25,NO. 3, MAY/JUNE 1999.*
8. Butcher (2002), "Improving software testing via ODC: Three Case Studies", *M.Butcher, H. Munro, T.Kratschmer, IBM Systems Journal, Vol.41, No.1*
9. Brad (2001)," How good is the software: A Review of Defect prediction Techniques", Brad Clark, Dave Zubrow, Carrnegie Mellon University.
10. Mullen (2002) "Orthogonal Defect Classification at CISCO", T.Mullen,D.Hsiao, Proceedings ASM conference.
11. Ram, "Orthogonal Defect Classification". www.Chillaregte.com/odc.
12. Ram (1992), "Adapting ODC to improve software quality: A case Study", Yang Gu, Software Engineer, IBM http://www.ibm.com.
13. Yang (1992), "Orthogonal Defect Classification A Concept for In-Process Measurements", Ram Chillarege, IEEE Transactions on software Engineering, Vol 18, No.11, November.
14. Paulk (1993), "Capability Maturity Model for Software", Version 1.1, Mark C.Paulk, Bill Curtis, Mary Beth Chrissis, Charles V.Weber,Software Engineering Institute.
15. Chillarege (2002) ,"Test and development process retrospective 0a case study using ODC triggers" , Chillarege, R.; Ram Prasad, K.'.
16. Ajit Ashok Shenvi, 2009,"Defect Prevention with Orthogonal Defect Classification", In Proc- ISEC '09, February 23-26, 2009.
17. Defect Prevention by SEI's CMM Model Version 1.1.,http://www.dfs.mil/technology/pal/cmm/vl/dp.
18. Linda Westfall, Defect Density http://www.westfallteam.com/Papers/defect_density.pdf.
19. Megan Graham, 2005, Software Defect Prevention using Orthogonal Defect Prevention. http://twinspin.cs.umn.edufiles/ODC_TwinSPIN.
20. Mukesh soni, 1997, Defect Prevention: Reducing cost and improving quality" published in IEEE Computer,(Volume 30, Issue 8), August 1997.
21. Pan Tiejun, Zheng Leina, Fang Chengbin, 2008, "Defect Tracing System Based on Orthogonal Defect Classification" published in Computer Engineering and Applications, vol 43, PP 9-10, May 2008.
22. Pankaj Jalote, Naresh Agarwal, 2007, "Using Defect Analysis Feedback for Improving Quality and Productivity in Iterative Software Development" In proc- ITI 3rd International Conference on Information and Communications Technology, pp. 703-713.
23. Ram Chillarege, Inderpal S Bhandari, Jarir K Chaar,Michael J Halliday, Diane S Moebus, Bonnie K Ray, Man-Yuen Wong, 1992, "Orthogonal Defect Classification - A Concept for In-Process Measurements", IEEE Transactions on Software Engineering, Vol. 18, No.11, Nov 1992.http://www.chillarege.com/odc/articles/odcconcept/odc.html
24. Stefan Wagner, 2008,"Defect Classification and Defect Type Revisited" Proceedings of the 2008 workshop on Defects in large software systems, (DEFECTS'08) pages 73-83, ACM Press, 2008.
25. Steve McConnell, "An ounce of prevention", IEEE software, May/June 2001.
26. Suma V and T R Gopalakrishnan Nair , 2008, " Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels" Proceedings of World Academy of Science, Engineering and Technology Volume 32 August 2008.
27. USC-CSE, 1999 - "COCOMO II Model Definition Manual", Computer Science Department, University of Southern California, Center for Software Engineering, Los Angeles, CA 90089-0781, 1999.
28. SEI, 1999 - "Process Maturity Profile of the Software Community: 1998 Year End Update," Carnegie Mellon University, Software Engineering Institute's Technical Report, Mar 1999.
29. Jones, 1998 - "The Impact of Poor Quality and Canceled Projects on the Software Labor Shortage," Capers Jones, Technical Report, Software Productivity Research, Inc.(an Artemis company),1998.

BIOGRAPHY

Ms. Arpita Mittal is working as Assistant Professor in Department of Computer Science at IIMT Merrut, UP, INDIA. Her Research activities are based on Software Engineering and Software Tesing. She is pursuing her M.Tech in Computer Science and Engineering from Amity University.

Mr. Sanjay Kumar Dubey is working as Assistant Professor in Department of Computer Science and Engineering in Amity University Noida, UP, INDIA. His Research area includes Software Engineering and Usability Engineering.He is pursuing his Ph.D in Computer Science and Engineering from Amity Unversity.