

Early Detection of Software Defects: Needs and Process

Anjaneya Awasthi

Abstract— Every Software Needs to be tested after it is developed. No software can be said to be free of defects or faults. Defects may be traced by employing a mechanism for testing of the Software and may be termed as Defect Sensing System. The defects or bugs reported by this mechanism can be handled in order to assure the quality of Software. This Paper presents the way that how the defects may be handled and suggests a method for doing so. This maybe termed as the Bug Management process. This may be used to get desired results such as: Iterative process handling and improvement on different levels or stages of Software development to avoid future failures and occurrences of these defects.

Index Terms— Defects, Defect Sensing System, Defect Analysis, Defect Handling

I. INTRODUCTION

A Bug or fault that may arise due to poor work done at any of the stages from starting through middle phases and final testing phase in Software development process may be termed as a 'defect'. The erroneous output may be present at any stage of software Development Life Cycle such as: Analysis, Requirement Specifications, Planning, Design or Coding. It is important to address these defects in timely manner so that they should not occur, and even if, they are missed or left at early stages, they should necessarily be detected in testing phase. Otherwise they may cause a disparity in the application of the software product and the product may not fulfill its desired outcome or result into total failure of the product. "To gain a deeper understanding of the effectiveness of the software process, it is essential to examine the details of defects detected in the past projects and to study how the same can be eliminated due to process improvements and newer methodologies."¹

This paper will focus on the study of related work to understand the software defects and importance of their early detection to avoid any hazardous consequences. The paper has highlighted the need for early detection of software defects. This is important to avoid the last moment issues in terms of operation of the software. To achieve the objectives of the paper, the various types of software defects have been identified in this paper. It is one of the most important contributions of the paper that the all possible types of defects have been clearly identified.

The early defect detection is very important and therefore, it is important to understand the process of early defect detection. This paper has been successful in developing a systematic process for early detection of the software defects.

II. OBJECTIVE OF THE RESEARCH PAPER

The study has the following objects to achieve:

- To identify the various types of defects in software development process
- To highlight the need for early detection of the software defects
- To develop a systematic process for early detection of the software defects

III. RELATED WORK

The previous studies in defect prevention were focused on defect prediction/forecasting and decide upon the team size of the testing resources required in order to complete the project on time and lot of effort were consumed in the debugging and get the defects eliminated. With the advent of SDLC processes many organizations formulated their own defect prevention mechanisms and many studies were conducted towards defect prediction/forecasting and prevention.

One study by Fang Chenbin² was introduction of a tool called Bug Tracing System (BTS) for defect tracing, has the advantage of popularity and low cost, and also improves the accuracy of tracking the identified defects. Work done by Stefan Wagner³ summarizes the work on defect classification approaches that have been proposed by two companies IBM and HP. The IBM approach is called Orthogonal Defect Classification (ODC) and the HP approach is based on three dimensions -Defect Origin, Types and Modes. Pankaj Jalote and Naresh Agarwal⁴ stressed on how analysis of defects found in first iteration can provide feedback for defect prevention in later iterations, leading to quality and productivity improvement. Ajit Ashok Shenvi⁵ worked under the philosophy that "capturing defects in the earlier stage of the life cycle" is a means of preventing defects in the later stages of the product life cycle and concentrated on finding out preventive action for functional defect types only. Suma V⁶ aimed to provide information on various methods and practices supporting defect detection and prevention based on three case studies and studied about the defect detection and defect prevention strategies adopted in these three projects only. All the above methodologies lacked some dimension in the early defect detection, defect detection process and defect prevention process and needed more attention.

In this study, the paper proposes to combine the above methodologies used such as ODC, Iteration defect reduction, capturing defects at early stage and finding out defect prevention for better classified type of defects and have attempted to come out with a defect detection process for continuous improvement of the Quality Processes and Defect Prevention.

Manuscript received Dec 20, 2016

Anjaneya Awasthi, Research Scholar, ShriVenkateshwara University, Gajraula, UP

IV. TYPE OF SOFTWARE DEFECTS

Whenever a software product is examined, different types of defects or bugs get encountered in software. It is important to identify the different types of defects that may arise in any stage of software development. The study has gone through the intensive literature review and software development processes and identified the various types of defects. These are mentioned in following paragraphs:

- **Defects at Requirement Analysis Phase:** This category of defects are the mistakes made in the definition or specification of the customer needs for a software product. This includes defects found in functional specifications; interface, design, and test requirements; and specified standards.
- **Defects at Design Phase:** The design phase is very important in software development lifecycle. Some mistakes may be in the design of a software product. These include defects found in functional descriptions, interfaces, control logic, data structures, error checking, and standards.
- **Defects at Coding Phase:** This category of defects is important to detect at the early stage. A mistake made in the implementation or coding of a program may be serious if not detected at an early stage. This includes defects found in program logic, interface handling, data definitions, computation, and standards.
- **Defects During Documentation:** A mistake made in software Product publication. This does not include mistakes made to requirements, design, or coding documents.
- **Test Case Defects:** The mistakes in the test cases results in unexpected results for the software product.
- **Miscellaneous Defects:** There are other different kinds of software defects that may arise during the software development process. These may include test tools, compilers, configuration libraries, and other computer-aided software engineering tools.

Defects at Requirement Analysis Phase	Defects at Design Phase	Defects at Coding Phase	Miscellaneous Defects
Functional Specifications	Functional descriptions	Defects in Program Logic	Test Case Defects
Test Requirements	Interface designs	Defects in Interface handling	Comptaion Issues
Design	Control Logics	Defects in Data Definitions	Configuration Libraries
Interface	Data Structures	Computations Defects	Defects related to Comcomputer-Aided Software Engineering tools
Specified Standards	Error Checking and standards		

Table 1: Types of software defects in Software development phases

V. NEED FOR EARLY DEFECT DETECTION

Defect prevention is an important activity in any software project. In most software development organizations, the project teams focus on defect detection and rework. Though defect prevention is also a very important activity in software development process. It is therefore advisable to make measures that prevent the defect from being introduced in the product right from early stages of the project. While the cost of such measures are the minimal, the benefits derived due to overall cost saving are significantly higher compared to cost of fixing the defect at later stage. Thus analysis of the defects at early stages reduces the time, cost and the resources required. The knowledge of defect injecting methods and processes enable the defect prevention. Once this knowledge is practiced the quality is improved. It also enhances the total productivity. The need and importance of early detection of software can be traced through following points:

1. Early Detection of Defects Reduces the Cost: “If an organization can prevent defects or detect and remove them early, it can realize significant cost and schedule benefits”⁷.
2. Early Detection of Defects Reduces Business Risks
3. It increase the Software Productivity
4. It improves the Security features of the Software
5. The early detection improves the overall quality of the Coding

VI. DEFECT DETECTION PROCESS

Early detection of software reduces the cost of the software projects. It is related to software development process improvement. “A software defect is a coding or logic error resulting in an application malfunction or incorrect results. A number of methods may be used to identify or detect defects at different phases of the development life cycle. It is well known that the later defects are detected the more expensive they are to fix or resolve. As such, organizations seek to identify potential defects as early as possible in the software development process. Early defect detection using static code analysis eliminates future costs and prevents problem expansion in the production or implementation phases”⁸. The defect can be prevented or detected at an early stage if each phase of software development phase is clearly defined and planned. The defect detection process simply includes the detailed development plan and to ensure that each step is implemented the way it has been planned. There are certain important points to be noted carefully in the development plan:

Process for Defect Detection in Requirement Analysis

- Systematic Noting of Customer operational requirements
- Noting the information about the restriction in the software
- Noting the requirements of Security features

Process for Defect Detection in design Phase

- Systematic and careful conversion of requirement into process diagrams
- **Data Dictionary:** All the data should be clearly defined with all the attributes so that deviation/ confusion can easily be eliminated.

Entity relationship diagram should properly develop which shows all the entities, their attributes and relationships between various entities.

Process for Defect Detection in Coding Phase

- Careful selection of Programming Language that fulfills the customers need as taken in requirement analysis phase.
- Dividing the software into different modules
- Developing the flow charts and algorithms module wise
- Analysis of Lines of Code and determination of developers requirements

Process for Defect Detection in Coding Phase

- Inspection of all the modules independently
- Integration Testing
- Acceptance Testing

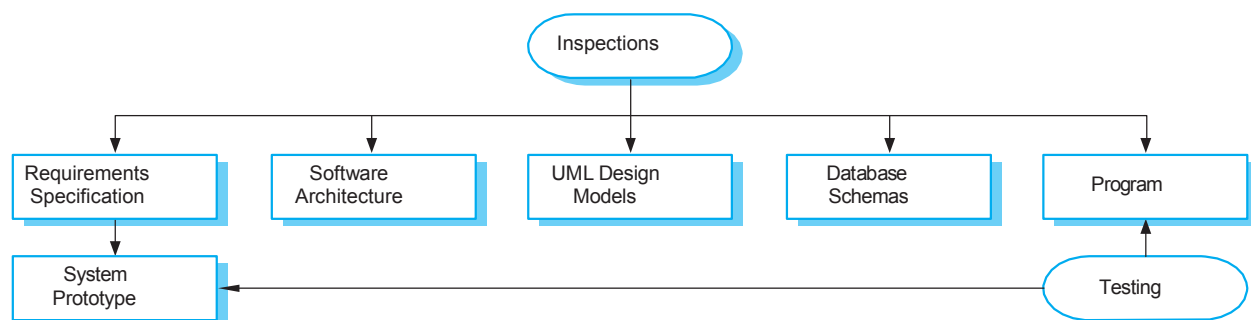


Diagram1: Inspections and Testing
 Source: Software Engineering by Ian Sommerville⁹

Process for Defect Detection in Maintenance Phase

Noting and correcting the defects as gap in requirements and development of software

With addition of new features in software as may be needed by the customer when it is in use, the software should be ensured for proper integration of new features with no conflict with earlier procedures in the software

The above mentioned processes will not only be helpful in early defect detection of defects but also it will be helpful in preventing the defects.

CONCLUSION

Regardless of the industry, the software has become a necessary in today environment. The software is useful only when it is fulfilling the requirements for which it has been acquired. The defects in software can be very harmful if they remain undetected upto the last phase. The early detection of software is very important for the success of software development projects. Implementation of defect preventive and detection action not only helps to give a quality project, but it is also a valuable investment. Defect prevention and early detection practices enhance the ability of software developers to learn from those errors and, more importantly, learn from the mistakes of others. The benefits of adopting strategy would be enormous and to list a few, Early detection of defects reduces development time and cost, satisfaction, reduces rework effort, thereby improves product quality.

REFERENCES

[1]Kumaresh S &Baskaran R (2010). "Defect Analysis and Prevention for Software Process Quality Improvement". International Journal of Computer Applications Vol. 8 ISSN:0975-8887

[2]Pan Tiejun, ZhengLeina, Fang Chengbin, 2008, "Defect TracingSystem Based on Orthogonal Defect Classification" published in Computer Engineering and Applications, vol 43, PP 9-10, May 2008.

[3]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

[4]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.

[5]Ajit Ashok Shenvi,2009,"Defect Prevention with Orthogonal Defect Classification", In Proc- ISEC '09, February 23-26, 2009

[6]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

- [7] Gary McGraw et.al.(June 2008). “Why is Security a Software Issue?” retrieved on 18 Dec 2016 from <http://www.informit.com/articles/article.aspx?p=1193473&seqNum=6>
- [8] The Benefits of Early defects Detection in Software Development. Retrieved on 18 December 16 from <http://www.castsoftware.com/glossary/early-defect-detection>
- [9] Ian Sommerville. Software Engineering. Published by Pearson Education, Inc., publishing as Addison-Wesley, Ninth Edition p 208