# A Survey of Defect Prediction on Software Leads to Quality Product

Swathine.K[1],
Research scholar, SNR Son College, Coimbatore, Tamil Nadu, India.
Email:{swathinekrishna@gmail.com}

Dr. J.KomalaLakshmi[2],
Research supervisor, Department of Computer Science,SNR Son College, Coimbatore, Tamil Nadu, India.
Email:{jkomalalakshmi@gmail.com}

**ABSTRACT**

Software team always try to produce a product with zero defects. Defect prediction leads to high quality product and quality assurance. Software defects prediction aims to reduce software testing efforts by guiding the testers through the defect classification of software systems. If the defective product released to customer, at customer side defect became fault. Existence of defect reduces the software reliability. Predicting the defects need more practices and knowledge. So, defect prediction is very important in the field of software quality and software reliability.

**KEYWORDS:** Defect prediction, Regression analysis, Six sigma approach, Debugging, Boosting technique.

## 1. INTRODUCTION

A software defect is an error, flaw, mistake, failure, or fault in a system that produces an incorrect or unexpected result or causes it to behave in unintended ways. Software defects are expensive in terms of quality and cost. Moreover, the cost of capturing and correcting defects is one of the most expensive software development activities. Software defect prediction is not a new thing in software engineering domain. Predicting defects is the proactive process of characterizing many types of defects found in software's content, design and codes in producing high quality product. Predicting total number of defects for system testing phase especially functional defects is significant in test process improvement. In general, we should focus on the following different aspects of Defect prevention, Defect detection, and Defect correction.

## 2. REVIEW OF LITERATURE

**[1]** Software defect prediction is the process of locating defective modules in software. To produce high quality software, the final product should have as few defects as possible. Early detection of software defects could lead to reduced development costs and rework effort and mo- re reliable software. So, the study of the defect prediction is important to achieve software quality.

**[2]** Six Sigma provides a structured and systematic way of building a defect prediction model for the testing phase. The Design for Six Sigma (DfSS) methodology provides opportunities to clearly determine what needs to be achieved from the research, issues to be addressed, data to be collected, the needs to be measured and how the model is generated, constructed and validated.

**[3]** The classification algorithms used in this study is Boosting. The software complexity measures such as LOC measure, Cyclomatic complexity.

## 3. METHODOLOGY

Debugging, Regression analysis, Six sigma, Boosting technique are for defect prediction.

### 3.1 Debugging

Debugging is the process of removal of a defect. It occurs as a consequence of successful testing. Debugging process starts with execution of test cases. The actual test results are compared with the expected result. The debugging process attempts to find the lack of correspondence between actual and expected results.

Common approaches in debugging are:

**Brute force method-** The memory dumps and run time traces are examined and program with write statements is loaded to obtain clues to error causes.

**Backtracking method-** This method is applicable to small program. In this method, the source code is examined by looking backwards from symptom to potential causes of errors.

**Cause elimination method-** This method uses binary partitioning tto reduce the number of location where errors can exist.

### 3.2 Regression Analysis

Predicting defects is the proactive process of characterizing, many types of defects found in software's content, design and codes in producing high quality product, presented that size and complexity metrics are among the

earlier approaches to defect prediction. Lines of code (LOC) and McCabe's cyclomatic complexity were used to predict defects in software.

As regression analysis is selected as the technique to construct the defect prediction model for system testing, three criteria are used to evaluate the outcome of the analysis:

- P-value – It determines the significance of the predictors to the discovery of functional defects. P-value must be less than 0.05
- R-Squared (R-Sq) –It is the amount of variation explained by the regression equation which is used to predict future outcomes on the basis of other related information. It is a statistical term saying how good the particular generated equation is at predicting functional defects.R-Sq. value must be above 85%
- R-Squared adjusted (R-Sq (adj.)) –It is a modification of r-squared used in regression and multiple regression to compare models with different number of explanatory terms. R-Sq. (adj.) must be greater than 85%

### 3.3 Six sigma approach

Six Sigma is chosen as the approach in building up the mathematical model for predicting functional defects in system testing phase. Software projects are selected from those adopting V-Model software development process.. This is due to its emphasis on voice of customers and systematic approach in building the model. Six Sigma provides a structured and systematic way of building a defect prediction model for the testing phase. The Design for Six Sigma (DfSS) methodology provides opportunities to clearly determine what needs to be achieved from the research, issues to be addressed, data to be collected, the needs to be measured and how the model is generated, constructed and validated. Design for Six Sigma DfSS, the research is organized according to DMADV phases: Define (D), Measure (M), Analyze (A), Design (D) and Verify (V). In general, DMADV phases are described as below:

- Define - identify the project goals and customer (internal and external) requirements
- Measure - determine customer needs and specifications; benchmark competitors and industry
- Analyze – study and evaluate the process options to meet customer needs
- Design – detailing the process to meet customer needs
- Verify – confirm and prove the design performance and ability to meet customer needs

### 3.4 Boosting technique

In software defect prediction, predictive models estimation is based on code attributes to assess software modules containing errors likelihood. The classification accuracy of Boosting techniques for software defect prediction based on the KC1 dataset is investigated. Identify defects based on existing software metrics using data mining techniques and thereby improve software quality which ultimately leads to reducing the software development cost in the developing and maintenance phase. Boosting works through classification algorithms use sequentially on training data reweighted versions.

The defect detectors are assessed as follows:

a = Classifier predicts no defects and module actually has no error.

b= Classifier predicts no defects and module actually has error.

c = Classifier predicts some defects and module actually has no error.

d = Classifier predicts some defects and module actually has no error.

The accuracy, probability of detection (pd) or recall, probability of false alarm (pf), precision (prec) and effort is calculated as

Accuracy = (a+b) / (a+b+c+d)

Recall = d/ (b+d)

Pf = c/ (a+c)

Prec = c/ (c+d)

Effort = ( c*LOC +d*LOC )/TotalLOC

## 3. COMPARISON

| Constraints\ Methods | Base | Activity |
|---|---|---|
| Debugging | Comparison of results from test cases | Removal of defect |
| Regression Analysis | Design,code,content | Verification and validation |
| Six sigma | Function | Defect density |
| Boosting technique | Code attribute | Defect detection |

## 5. RESULT

By using statistical approach such as regression analysis, the research can justify the reasons and significance of metrics from requirement, design and coding phase in predicting defects for system testing. Six Sigma provides a structured and systematic way of building a defect prediction model for the testing phase. Boosting technique investigate the classification performance of for defect prediction. KC1 dataset was used for evaluation of the boosting algorithms.

## ACKNOWLEDGEMENT

I express my heartfelt thanks to Dr.H.Balakrishnan, Principle, S.N.R son college for making available in excellent infrastructure. I thank my guide Dr.J.Komala Lakshmi, for her support, effort to guide and encourage me. I thank Dr.G.P.Ramesh Kumar, Head of the department of Computer science for providing necessary facilities. I thank Dr. AnnaSaro Vijendran, Head of department of Computer Application.

## REFERENCE

[1] Naheed Azeem, Shazia Usmani, "Defect Prediction Leads to High Quality Product" Journal of Software Engineering and Applications, 2011, 4, pp:639-645.

[2] Muhammad Dhiauddin Mohamed Suffian, Suhaimi Ibrahim, "A Prediction Model for Functional Defects in System Testing using Six Sigma", ARPN Journal of Systems and Software, Volume 1 No. 6, September 2011.

[3] V.Jayaraj, PhD," Software Defect Prediction using Boosting Techniques", International Journal of Computer Applications,Volume 65– No.13, March 2013.

[4] Lalit Kumar Singh, Anil Kumar Tripathi, Gopika Vinod," Software Reliability Early Prediction in Architectural Design Phase: Overview and Limitations" Journal of Software Engineering and Applications, 2011, 4, pp181-186.

[5] Shaik Nafeez Umar," software testing defect prediction model-a practical approach" International Journal of Research in Engineering and Technology, ISSN: 2319-1163.

[6] Muhammad Dhiauddin Mohamed Suffian, Suhaimi Ibrahim," A Prediction Model for System Testing Defects using Regression Analysis", International Journal of Soft Computing And Software Engineering, Vol.2,7, 2012,July 25, 2012, e-ISSN: 2251-7545.

[7] Mrinal Singh Rawat, Sanjay Kumar Dubey, "Software Defect Prediction Models for Quality Improvement: A Literature Study", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012