

Software Metrics Quality Testing (SMQT) Prediction using Logit Regression Model

Sanjeev Kumar Punia, PhD
Computer Science & Engineering Department
JIMS Engineering Management Technical
Campus, Greater Noida

Kuldeep Malik, PhD
Department of Computer Science
ITS Engineering College
Greater Noida

ABSTRACT

Software testing is always a hot field in software engineering for both industry and academia. Traditional research focused on software quality instead of quality testing which can be evaluated through organization software testing. Previously, software testing management was based on statistical methods completely. In recent years, machine learning algorithms are developing rapidly for software quality testing management.

In this reference, Logit regression (LR) model used widely in various domains like book classification, credit card investigation etc. In this research paper, the objective is to implement the Logit regression model in software testing quality management. Here, a latest metrics framework methodology is implemented for matrix testing management. Finally, the Logit regression model is testing on a financial unit data set.

Keywords

Logit regression model, software metrics testing management, total testing quality, WEKA

1. INTRODUCTION

Software testing is always the most important and crucial phase in software development life cycle. In every phase, testing is the main part involved thoroughly in software all project. Garrett et. al. [1] explained that testing activity also helps in reducing the working program cost. However, software testing cannot identify all available defects those may lead to big losses.

K. Miller [2] explains in National Institute of Standards report that software testing estimation costs is nearly \$69276.46 billion internationally per year. A. Bansal et. al.

[3] proposed a different strategy to improve the quality and efficiency in software quality testing management. In past, quality testing evaluation works was based on the traditional statistical methods. In the recent years, new techniques with machine learning and data analysis are very powerful in data analysis, modeling and visualization. Many machine learning technique developed rapidly for applications in different field like data mining, natural language processing, image recognition and expert systems etc. Therefore, in this paper, the primary objective is to build a framework for software metrics quality testing prediction management using Logit regression model.

In this research work, the total quality software metrics testing management theory is diversified using research tool. Later, the methodology is concluded based on the analysis of data sets result. This research paper is organized in four different sections. In second section, software metrics testing management framework is introduced. In third section, evaluation of software metrics testing quality framework is explained using Logit regression model. Finally, research work is concluded and discuss about the future scope in last section.

2. METRICS TESTING FRAMEWORK TRODUCTION

E. Allen et. al. [4] explained that in software development life cycle (SDLC), every software engineering project have different phases namely initiation, planning, requirement analyzation, designing, developing, testing, deployment, implementation, operation, maintenance and disposition. The software development life cycle starts from planning and ends with disposition. The different phases of software development life cycle are shown in Figure 1.

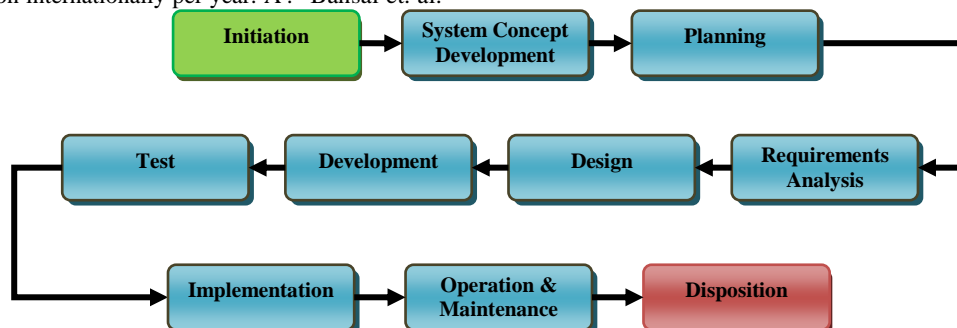


Figure 1: Software Development Life Cycle (SDLC) Phases

In software development life cycle, requirements analysis, design and development collectively called development stage (DS). The collection of different testing phase as unit testing, integration testing, system testing, stress testing, smoke testing and acceptance testing collectively called

testing stage (TS). The collection of all classic management theories like materials, manpower, machines, methods and environment is called total quality management (TQM) as shown through Figure 2.

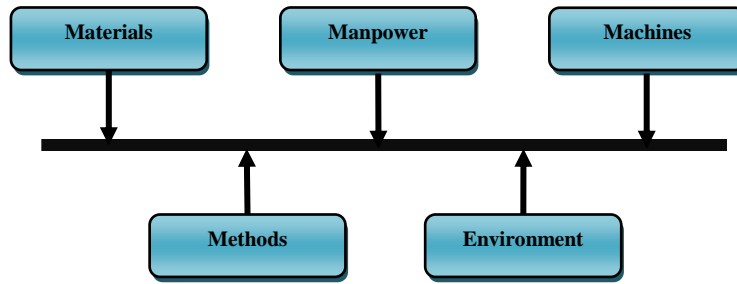


Figure 2: Total Quality Management Architecture

In software testing, total quality management metric can be classified in two different stages namely development stage and testing stages. In this research, three different frameworks namely (i) people strengths (PS) (ii) project attributes (PA)

and (iii) process testing (PT) are defined and used. In this research, eighteen different metrics with their sub class, attributes, type and range are considered as shown in Table 1.

Table 1: Testing Management Metrics Framework

CLASS	SUB CLASS	ATTRIBUTES	TYPE	RANGE
People Strengths (PS)	General Information	People strength	Continuous	$[1, \infty)$
		Mean age	Continuous	$[0, \infty)$
	Field Strength	Work experience	Continuous	$[0, \infty)$
		Job performance	Continuous	$[0,10]$ out of 10
		Training time	Continuous	$[0, \infty)$
Project Attributes (PA)	Project Scaling	Function points	Continuous	$[0, \infty)$
		Total test cases	Discrete	positive integer
		Total modules	Discrete	positive integer
		Coupling degree	Discrete	positive integer
	Requirement Changing	Total requirements	Discrete	positive integer
Process Testing (PT)	Test Coverage Rate	Passing rate	Continuous	$[0,1]$
		Coverage rate	Continuous	$[0,1]$
	Defect Rate	Defect density	Continuous	$[0,1]$
		Average defect severity level	Discrete	$[4,1]$ (critical, major, minor, normal)
		Average defect age length	Continuous	$[0, \infty]$
	Test Resource	Test environment failure rate	Continuous	$[0,1]$
		Test automation rate	Continuous	$[0,1]$

2.1 People Strength

In every software engineering project, developers and testers are two most important factors for software testing management. The testers have a great and important decisive role during software quality assurance (SQA). The developers play a different role in software testing management due to coding part. In general, researchers are continuously working to design and find the best methodology from existing one through their skills and experience.

A. Lodhi et. al. [5] explained that software quality can be impacted through people different attributes. They proposed people metrics division in two different parts as (i) general information and (ii) field strength. In this, general information is the collection people strength, mean age while field strength

is the collection of work experience, job performance and training time. The work experience is measured through software testing service length. The job performance is measured through the end year assessment level and training time is measured through skills test training duration.

2.2 Project Attributes

I. Witten et. al. [6] shows that software project process testing depends on many testing factors like analysis, design, execution, implementation and maintenance. They divided the testing metrics framework in two different phases as (i) project scaling and (ii) requirement changing. The project scaling can be measured using function points and lines of code. They use function point for the estimation of functional size measurement. Arti Arya et. al. [7] measure project

scaling through total test cases, total modules and coupling degree in data mining and machine learning. The requirements changing can be calculated using total requirements and requirements certainty. The total requirement can be calculated using equation 1:

$$R = F + N \quad (1)$$

Where

R = Total requirements

F = Functional requirements

N = Non-functional requirements

The requirement certainty can be calculated using equation 2:

$$RC = \frac{RQ}{TQ} \quad (2)$$

Where

RC = Requirement certainty

RC = Reviewers requirements

TQ = Total requirements

2.3 Process Testing

J. Pekar [8] explained that testing process is a basic and core technique for software quality testing management. The process testing has three major quality testing measurement metrics as (i) test coverage rate and (ii) test defect rate and (iii) test resources. The test coverage rate is the collection of passing rate and coverage rate. The coverage rate is the total test cases performed and passing rate is the total correct result test cases. These two can be calculated through line of codes. The defect rate is the collection of defect density, average defect severity level and average defect age length. The defect density can be calculated using equation 3:

$$DD = \frac{TD}{FP} \quad (3)$$

Where

DD = Defect density

TD = Total defects

FP = Function points

The average defect severity level is divided in four different levels as normal, minor, major and critical. The average defect age length can be calculated using equation 4:

$$ADSL = DFD - DDD \quad (4)$$

Where

ADSL = Average defect level length

DFD = Defect fixed date

DDD = Defect detection date

The test resource depends on environment failure rate and test automation rate values. The test resource value can be calculated using equation 5:

$$TR = \frac{TEFR}{TAR} \quad (5)$$

Where

TR = Test resources

TEFR = Test environment failure rate

TAR = Test automation rate

The test environment failure rate depends on total failure test cases and total test cases. The test environment failure rate can be calculated using equation 6:

$$TEFR = \frac{TFTC}{TTC} \quad (6)$$

Where

TEFR = Test environment failure rate

TFTC = Test failure test cases

TTC = Total test cases

The test automation rate depends on total automated test cases and total test cases. The test automation rate value can be calculated using equation 7:

$$TAR = \frac{TATC}{TTC} \quad (7)$$

Where

TAR = Test automation rate

TATC = Total automated test cases

TTC = Total test cases

3. LOGIT REGRESSION MODEL

W. Hosmer et. al. [9] stated that Logit regression model is based on problem analysis techniques. The independent variable result is measured in discrete values. E. Allan et. al. [10] explained that the Logit regression testing management is choose the best available model that describes the relationship between software project quality testing and collected data management during testing process.

In this research, the Logit regression model testing quality estimation on a financial unit data set is performed. The financial unit has more than 350 systems to generate information. The data set updating frequency was 21 days. In this experiment, 450 data set sample instances are selected for the verification of result. During this, every instance have same i.e. 18 attributes metrics as mentioned in Table 1. In this, testing quality (a new attribute) is predicted using different software testing methodologies. WEKA tool is used for machine learning techniques during the research. The research data set is shown in Figure 3.

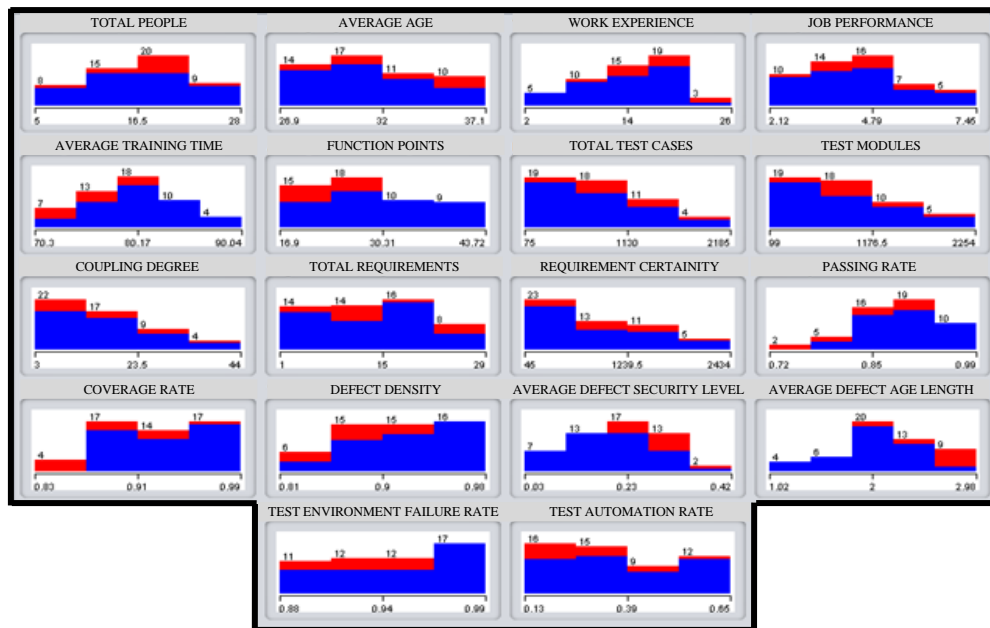


Figure 3: Distribution Data Visualization

4. CONCLUSION

In this paper, eighteen attribute metrics are demonstrated in software quality testing management framework through total quality management during testing process. The result also showed that the software quality testing management metrics framework is flexible by implementing the Logit regression model on financial unit data sets. The quality testing accuracy improved greatly using Logit regression model. The analysis shows that a new metric can be added i.e. total quality management metrics in a financial unit for testing accounting projects. In future, this framework can be implemented on a large financial unit data set for quality metrics testing management. The Logit regression model may be used for comparing different testing machine learning algorithms.

5. REFERENCES

- [1] Garrett and Dustin, 2018 Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality. 4th edition, Addison-Wesley Professional.
- [2] K. Miller, 2017 Testing Management Theories - Critical Realist Philosophy and Research Methods. Strategic Management Journal.
- [3] A. Bansal, A. Singhal and S. Kukreja 2016 A Critical Survey on Test Management in IT Projects. International Computing Communication Automation Conference (ICCAC).
- [4] E. Allen and M.Tyagi 2018 Logistic Regression Modeling Of Software Quality, International Journal of Reliability, Quality and Safety Engineering
- [5] A. Lodhi and K. Turowski, 2018 Test Management Framework for Managing IT Projects in Industry. International Conference on E-Business Engineering (ICEBE), IEEE 25th International Conference.
- [6] I. Witten and G. Holmes, 2018 WEKA - A Machine Learning Workbench. Seventh Australia and New Zealand International Conference on Intelligent Information Systems, Brisbane, Australia.
- [7] Arti Arya and A Manjula 2017 Study on Software Metrics based Software Defect Prediction using Data Mining and Machine Learning Techniques, International Journal of Database Theory and Application
- [8] J. Pekar 2017 Total Quality Management: Guiding Principles for Application. IEEE 16th International Conference.
- [9] W. Hosmer and S. Lemeshow 2018 Applied Logit Regression. 15th Edition, Wiley.
- [10] Fadel and B. Mourad 2016 Empirical Analysis of Object-Oriented Design Metrics for Predicting Unit Testing Effort of Classes, Journal of Software Engineering and Applications