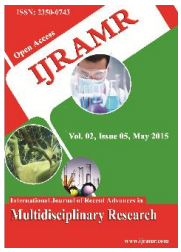
International Journal of Recent Advances in Multidisciplinary Research
Vol. 02, Issue 05, pp.0418-0420, May, 2015



Research Article

STUDY SOFTWARE RELIABILITY USING DEFECT DENSITY IN SOFTWARE ENGINEERING

***[1]Dr. Ramesh R. Manza and [2]Yogini B. Patil**

[1]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, 431004(MS) India
[2]Department of Computer Science, Shantidevi Chavan institute of Polytechnic, Chalisgaon 424101(MS), Jalgaon, India

ARTICLE INFO

Article History:
Received 19th February 2015
Received in revised form
21th March, 2015
Accepted 25th April, 2015
Published online 31th May, 2015

Keywords:
Defect density,
S/W reliability,
Functional requirements.

ABSTRACT

To fulfill software requirements and to satisfy the customer, software must be tested by all angles of users and developer .But the question is that when to stop testing or whether it is enough to test software, if it not tested properly by software engineer then it is handover to customer, if we want to give quality s/w product then we must calculate software reliability as it one of quality measure. s/w reliability is probability of failure free operation of software. usually tester focus on testing functional or normal behavior of units of s/w and it is in developer's environment but software reliability focus on operational behavior of software in natural setting that is in presence of customers and users, it is need of software community to combine both functional and operational approach so that testing should be done effectively and reliability should be accurately measured. In this paper issue is highlighted that using metrics defect density and statistical modeling software reliability can be measured.

INTRODUCTION

As we know software engineering is i) The application of systematic, disciplined, quantifiable approach to development, operation and maintenance of software that is application of engineering to software. ii) "The study of approaches as in the above statement" (IEEE). It contains different phases

i) Analysis
ii) Design
iii) Coding
iv) Support to develop software.

Software Requirements

It can be classified as

i) Functional
ii) Non-functional.

Functional Requirements specify which output should be produced from given inputs.They describe the relationship between the input and output of the system.

Corresponding author: Dr. Ramesh R. Manza
Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, 431004(MS) India

Non-functional requirements define system properties such as Reliability, Response time and storage requirements etc. Here some requirements are product requirements and some are external requirements. Product requirements specify that delivered product must work in a way such as execution speed, reliability etc. (as in tree dia.1). The external requirements are external to system and development process e. g interoperability requirements, legislative requirements etc. Requirements are validated by testing phase of software engineering. Every time customer/users executes program it tested. System is useless if non-functional requirements are not met.

Software reliability

It is amount of time that the software is available for use as indicated sub attributes such as maturity, fault tolerance, recoverability etc. It is also defined as probability of failure free software operations for specified period of time in specified environment (ANSI/IEEE, 1991). A simple measure of reliability is mean time between failure (MTBF). It is calculated as MTBF=MTTF+MTTR here, MTTF=mean time to failure MTTR=mean time to repair. In addition to reliability measure, we must develop a measure of availability. It is probability that a program is operating according to requirements at a given point in time and is defined as,

Availability=(MTTF/(MTTF+MTTR))*100%. S/W Reliability field includes to calculate it and develop it to improve software development process it is calculated by number of field failures along with some additional information such as time at which the failure was found, in which part or module it was found, the state or mode of software at that time, the nature of failure etc.ISO 9000-3 (ISO, 1991). S/W reliability is one of quality factors of ISO 9126 which define reliability as amount of time that the software is available for use as indicated by the sub attributes: maturity, fault tolerance, recoverability.

Related work and contribution

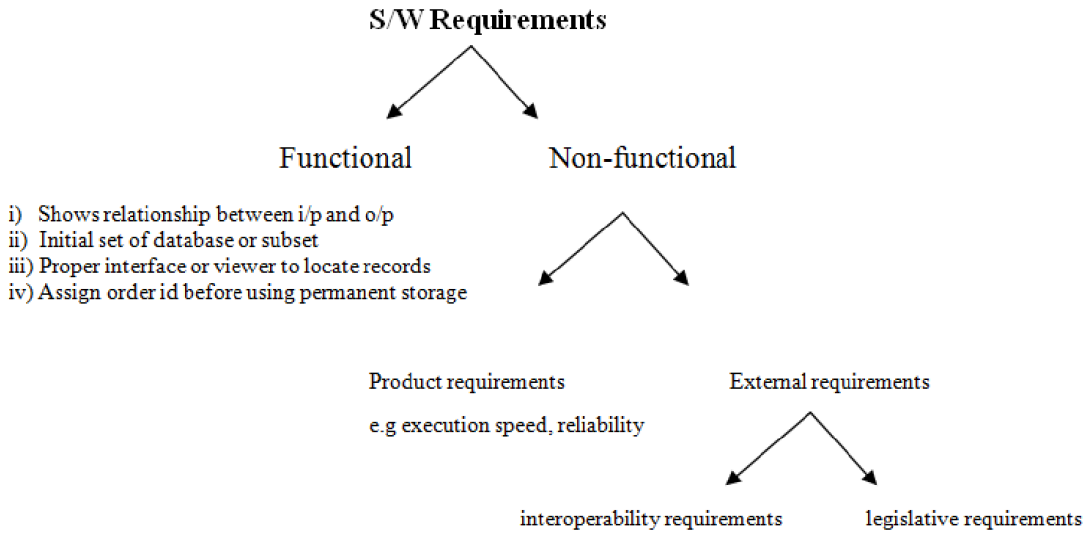
Software reliability is important to achieve a ideal level of performance in limited time frame. It is used to measure defects that cause software failures in different mode and behavior of software with fixed time. As software is present in different forms like programs, rules, data, documentation .If defects occurs it affects on transportation, telecommunication, military, entertainment offices, aircraft and business software reliability is very important to study. In this paper classification on software reliability models based on various dimension is given (Latha Shanmugam and Dr. Lilly Florence, 2012). In this paper different models are discussed as Time Between Models, Fault Count Models, Fault seeding models, Input Domain based models, Identical Error Behavior Models, Independently Distributed inter-failure models. Here it is reflected that testing and reliability are different. Testing doesn't focus how it will operate in the field, here test case are designed for boundary conditions and software is tried to break or trace than normal condition to find different failure paths .Software reliability on other side focus on operation profile of software so that accurate reliability estimation and prediction occurs so in future testing and reliability should apply together (Software Reliability Engineering: A Roadmap, ?).

In this paper it is reveal that collecting metrics data for reliability prediction is difficult. It can be obtain from organization such as NASA, Microsoft, IBM, Cisco etc. So extracting information of metrics data using statistical and probability foundation, gives right direction for investigation in software engineering (Cheng et al., 1997).

To achieve predictable reliable software traditional software reliability models takes defect density metrics as input and produce reliability quantity as the output. In the future function should be developed what should the reliability process for what goal and what will be metrics such feedback is useful for s/w reliability engineering process (Software Reliability Engineering: A Roadmap, ?). In this research paper data from several projects show a significant relationship between size of module and its defect density. Here it is shown that defects may be due to interconnections of module or due to complexity of individual modules. controlling module size distribution defect density can be reduced. Exponential distribution shows that when modules broken into number of modules and bring then together then defect density is minimal (Yashwant K. Malaiya and Jason Denton, 2000).

MATERIALS AND METHODS

In this research if S/W reliability is to be study then different testing strategies must be addressed specify product requirements in a quantifiable manner long before testing commences. State testing objectives explicitly in measurable terms e.g test effectiveness, test coverage, mean time to failure, the cost to find and fix defects, defect density or frequency of occurrence, test work hour per regression test etc..Understand the user of the software (through use cases) and develop a profile for each user category. Develop a testing plan that emphasizes rapid cycle testing to get quick feedback to control quality levels and adjust the test strategy. Build robust software that is designed to test itself and can diagnose certain kinds of errors. Use effective formal technical reviews as a filter prior to testing to reduce the amount of testing required. specify product requirements in a quantifiable manner long before testing commences. Understand the user of the software (through use cases) and develop a profile for each user category. Develop a testing plan that emphasizes rapid cycle testing to get quick feedback to control quality levels and adjust the test strategy. Build robust software that is designed to test itself and can diagnose certain kinds of errors. Use effective formal technical reviews as a filter prior to testing to reduce the amount of testing required. State testing objectives explicitly in measurable terms as mention above (Glib, 1995).



tree dia :1

Defect density=Total number of known defects/KLOC, where KLOC=LOC*1000.Here relationship between defect density and three repository metrics, i.e number of downloads, software size and number of developers (Cobra Rahmani and Deepak Khazanchi, 2010). In this paper regression analysis tools used which offers many statistical options such as Histogram plot, Analysis of Variance, R-squared and the significance of observed regression line (p-value). P-value is significance level to accept or reject tests. Here defect density is compared with number of downloads, software size, and number of developers. Regression analysis doesn't show relationship between defect density and number of downloads in OSS Project (i. e Open Source Software Project) (Cobra Rahmani and Deepak Khazanchi, 2010).

Research question

Using non linear regression and different statistical tools like p-value, R-value, we can explain different trend in OSS Project or exhibit relationship between Defect Density and number of downloads in OSS Project which is not by linear regression this may be good research question.

Conclusion

Using statistical modeling and software reliability theory, models of software failures as function of execution time can also be developed. It is observed that by collecting metrics during software testing and making use of existing software reliability models, it is possible to develop meaningful guidelines for S/W reliability and it may gives new rule, new model, or hypothesis that will be topic of research.

REFERENCES

"Software Engineering: A Practitioner's Approach, 6/e; Chapter 14: Software Testing Techniques", R.S. Pressman and Associates, Inc., 2005.

Institute of Electrical and Electronics Engineers. ANSI/IEEE standard glossary of software engineering terminology, IEEE Std. 729-1991.

ISO 9000-3. Quality management and quality assurance standard- part 3: guidelines for the application of ISO 9001 to the development, supply and maintenance of software. Switzerland: ISO; 1991

Volume 2, *Issue 10*, *October 2012*. *ISSN*: *2277 128X*. *International Journal of Advanced Research in. Computer Science and Software Engineering.*

Software Reliability Engineering: A Roadmap

Michael R. Lyu *Computer Science and Engineering Department The Chinese University of Hong Kong, Hong Kong lyu@cse.cuhk.edu.hk*

Cheng, J., D.A. Bell, and W. Liu, 1997. "Learning Belief Networks from Data: An Information Theory Based Approach,**"** *Proceedings of the Sixth International Conference on Information and Knowledge Management,* Las Vegas, pp. 325-331.

International Journal of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 4 2250-3005(online) August| 2012 Page 1060 "A Selective Survey and direction on the software of Reliability Models" By, Vipin Kumar.

Latha Shanmugam, Dr. Lilly Florence. Volume 2, Issue 10, October 2012 ISSN: 2277 128X *International Journal of Advanced Research in Computer Science and Software Engineering Research* Paper "An Overview of Software Reliability Models ".

(GIL 95) Glib, T. "what we fail to do our current testing culture" Testing Techniques news letter, Software Research, Inc. San Francisco, 1995

Westfall, L. 2010. "Defect Density," Available: http://www. westfallteam.com/Papers/defect_density.pdf.

Yashwant K. Malaiya and Jason Denton, 2000. "Module Size Distribution and Defect Density", 0-7695-0807-3/00 by IEEE.

Cobra Rahmani and Deepak Khazanchi, "A Study on Defect Density of Open Source Software", DOI 10.1109/ICIS. 2010.11.

****