



IMPROVE THE RELIABILITY AND QUALITY OF SOFTWARE BY USING SOFTWARE METRICS

D.Jyothirmai¹, M.Suresh Kumar², Dr. K. Subba Rao³

¹Asst.Professor, BVRIT-Narsapur-Medak-India

²DRDO Scientist-E-Hyderabad

³Professor, BVRIT-Narsapur-Medak-India

ABSTRACT

In the Software Engineering, The Software Reliability is an important facet of software quality. “Software Reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment”. Software Reliability is different from Hardware reliability. Though Software Reliability is defined as probabilistic function of time, it is not a direct function of time.

Achieving Software reliability is hard because the complexity of software tends to be high. Various approaches can be used to improve the reliability of the software, however, it is hard to balance development time and budget with software reliability. But the best approach to assure software reliability is to develop a high quality software through all of the stages of software life cycle. In this paper we will discuss how software metrics can improve the reliability and quality of the software products.

Keywords: Software Engineering, Software Reliability, Quality, Software Products,

1.INTRODUCTION

Software reliability is the probability that the “Software will work without failure for a specified period of time”. Failure means the program in its functionality has no met user requirements in some way. Software reliability concerns itself with how well the software functions to meet the requirements of the customer.

Reliability represents a user oriented view of software quality. Initially, Software quality was

measured by counting the faults in the program and so this approach is developer oriented where as reliability is user oriented, because, it relates to operation rather than design.

2. SOFTWARE RELIABILITY AND HARDWARE RELIABILITY

Software reliability is not a direct function of time. Hardware parts may become old and wear out with time, but software will not change over time unless the software is changed or modified intentionally.

In Hardware reliability, in the first phase of the manufacturing, there may be a high number of faults. But after discovering and removing faults this number may decrease and gradually in the second phase (Useful life) there exists only a few number of faults. After this phase, there will be wear out phase in which, the physical component wear out due to the time and usage and the number of faults will again increase. The Phases of hardware when considering reliability is shown in the below Figure-1

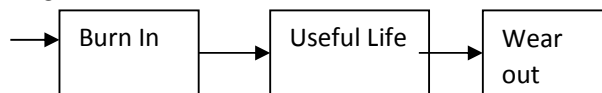


Figure-1: Phases of hardware when considering reliability

The Phases of software when considering reliability is shown in the below Figure-2

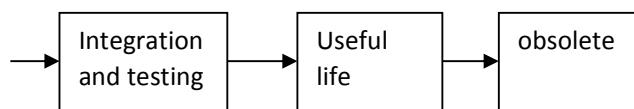


Figure-2: Phases of software when considering reliability

But in software reliability, at the first phase, i.e while testing and integration there will be high number of faults, but after removing the faults , there exists only a few number of faults and this process of removing the faults continues at a slower rate . Software products will not wear out with time and usage , but may become outmoded at a later stage.

3. NEED FOR SOFTWARE RELIABILITY MEASUREMENT

In any software industry , system quality plays an important role . It comprises of hardware quality and software quality. We know that hardware quality is constantly high . So if the system quality changes , it is because of the variation in software quality only. Software quality can be measured in many ways. Reliability is an user – oriented measure of “software quality”. Suppose assume that there are 3 programs that are executing to solve a problem. By finding the reliability of each program we can find which program has less reliability and we can put more effort to modify that program to improve the overall reliability of the system. So always there is a need to measure the reliability. In the later sections we will discuss about how to improve reliability and quality of the software

4. INCREASING THE RELIABILITY AND QUALITY OF THE SOFTWARE

a) Faults and Failures :

Software failure: It is the departure of the external results of program operation from requirements . So failure is dynamic. It depends upon the operation and behaviour.

Difference between fault and failure: Failure Is not the same thing as a “Bug “ or “ fault” . There is a lot of difference between these two terms. A “fault” is a defect in a program which arises when programmer makes an error and causes “failure” when executed under particular conditions.

b) Increasing Reliability:

Reliability can be increased by preventing the above said errors and developing quality software through all of the stages of software life cycle. To do this, we have to ensure that whether the requirements are clearly specifying the functionality of the final product or not (Requirement phase)

Among the phases of the software reliability , the second one i.e useful life is the most important one and so the software product must be maintained carefully. So we have to ensure that the code generated can support maintainability to avoid any additional errors.(Coding phase).

Next we have to verify that all the requirements specified in the requirement phase are satisfied or not . (Testing phase)

As reliability is an attribute of quality , we can say that reliability depends on software quality .So to build a high reliable software there is a need to measure the attributes of quality that are applied at each development cycle. Software metrics are used to measure these applicable attributes. The following sections shows different types of metrics that are applied to improve the reliability of system.

5. SOFTWARE METRICS FOR RELIABILITY:

The Metrics are used to improve the reliability of the system by identifying the areas of requirements (for specification),Coding (for errors),Testing (for verifying) phases.

The different types of Software Metrics that are used are

a) Requirements Reliability Metrics:

The Requirements indicate what features the software must contain. So for this requirement document, a clear understanding between client and developer should exist. Otherwise it is critical to write these requirements .The requirements must contain valid structure to avoid the loss of valuable information. Next , the requirements should be thorough and in a detailed manner so that it is easy for the design phase. The requirements should not contain inadequate information .

Next one is to communicate easily .There should not be any ambiguous data in the requirements. If there exists any ambiguous data , then it is difficult for the developer to implement that specification. Requirement Reliability metrics evaluates the above said quality factors of the requirement document.

b) Design and Code Reliability Metrics:

The quality factors that exists in design and coding plan are complexity, size and modularity. If there exists more complex modules, then it is difficult to understand and

there is a high probability of occurring errors. So complexity of the modules should be less.

Next coming to size, it depends upon the factors such as total lines, comments, executable statements etc. According to SATC, the most effective evaluation is the combination of size and complexity.

The reliability will decrease if modules have a combination of high complexity and large size or high complexity and small size. In the later combination also the reliability decreases because, the smaller size results in a short code which is difficult to alter. These metrics are also applicable to object oriented code, but in this, additional metrics are required to evaluate the quality.

c) Testing Reliability Metrics:

Testing Reliability Metrics are uses two approaches to evaluate the reliability.

First approach is, it ensures that the system is fully equipped with the “functions” that are specified in the requirements. Because of this, the errors due to the lack of functionality decreases.

Second approach is nothing but to evaluating the code, finding the errors and fixing them.

The current practices of software reliability measurement can be divided into four categories.

i) Product Metrics

ii) Project Management Metrics

iii) Process Metrics

iv) Fault and Failure Metrics

The above four metrics are used as follows:

i)Product Metrics: The software size and complexity plays an important role in design and coding phase. One of the product metrics called function point metric is used to estimate the size and complexity of the program.

ii)Project Management Metrics: These are increases reliability by evaluating the Management process

iii) Process Metrics: These can be used to estimate, monitor and improve the reliability and quality of the software.

iv)Fault and Failure Metrics: These can be determines, when the software is performing the whole functions that are specified by the requirement documents without any errors. It takes the faults and failures that arises in the coding and analyzes them to achieve this task.

6. CONCLUSION:

In any Software Industry, to achieving software reliability is the key task. Achieving Software reliability is hard. Because the complexity of the software tends to be high. Reliability is an attribute of quality and software quality can be measured. So reliability depends on high software quality. So at each development phase, some quality attributes are applied and the reliability and quality of the software can be improved by applying software metrics at each of these development phases. This metrics measures software reliability in Requirements, Design and coding, and testing phases. In the future, The software metrics are used for improve the Quality and Reliability of Software is more better than present in the Software Industry.

7. REFERENCES:

- 1) Musa, J.D, A. Iannino and K.okumoto Software Reliability : Measurement, Prediction, Application, professional Edition : Software Engineering Series, McGraw- Hill, NewYork, NY.,1990
- 2) Software Reliability by John Musa.
- 3) Software Metrics and Reliability by Linda Rosenburg, Ted Hammer, Jack Shaw satc.gsfc.nasa.gov/support/ISSRE_NOV98/software_metrics_and_reliability.html - 26k
- 4)Software Reliability by Jiantao. http://www.ece.cmu.edu/~koopman/des_s99/sw_reliability/
- 5 .Goel A.L. and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” IEEE Transactions on Reliability, vol. 28, no. 3, pp. 206–211, 1979.
- 6 Gokhale S.S., Triwedi K.S., “A Time Structure based Software Reliability Model”, Annals of Software Engineering, 8, 85-121, 1999.
7. Gupt R. D, Kundu D. Exponentiated exponential family; an alternative to gamma and Weibull. Biometrical Journal; 43: pp.117-130, 2001.
8. Huang, C. Y., Lyu, M. R. and Kuo, S. Y, A unified scheme of some non-homogeneous Poisson process models for software reliability estimation, IEEE Transactions on Software Engineering, Vol. 29, pp.261–269, 2003.
9. Huang C.Y., “Performance Analysis Of Software Reliability Growth Models With Test

- Efforts And Change-Point” Journal of Systems and Software , 76, pp. 181-194.2005a
10. Iannino A., Musa J.D., Okumoto K., Littlewood B. “Criteria for Software Reliability Model Comparisons”, IEEE Transactions on Software Engineering, SE-10(6): pp687-691, 1984
11. Inoue S and Yamada,S “ A software reliability growth modeling based on infinite server queuing theory, Proceedings of the 9th ISSAT International Conference on Reliability and Quality in Design, Hawaii , pp. 305-309, 2002.
12. Inoue, S. and Yamada, S. ‘Testing-coverage dependent software reliability growth modeling’, Int. J. Reliability, Quality and Safety Engineering, Vol. 11, No. 4, pp.303–312., 2004.
13. Ishii, T. and Dohi, T. “Two-dimensional software reliability models and their application’, (PRDC) 12th Pacific Rim International Symposium on Dependable Computing, pp.1–8, 2006.
14. Inoue, S. and Yamada, S. ‘Two-dimensional software reliability assessment with testing coverage’, *Second International Conference on Secure System Integration and Reliability Improvement*. 14–17 July, .150–156,2008.
15. Inoue, S. and Yamada, S. “Two-dimensional software reliability measurement technologies’, Proceedings of IEEE IEEEM, pp.223–227, 2009.
16. Jelinski Z and Moranda PB, “Software Reliability Research”, Freiburger W., (Ed.) Statistical Computer Performance Evaluation, New York, Academic Press, pp. 465-497, 1972.
17. Puli Nageswararao, D.Jyothirmai, Dr.K.Subba Rao Published a international Journal on” Improvement in the Software reliability by using the Software Reliability Characteristic Model and Measures of Defect Control ” International Journal of Engineering Science AND Research Technology(IJESRT), Vol: 7 Issue: 3,ISSN No: 2277-9655, Mar 2018(UGC Approved Journal).
18. Puli Nageswararao, D.Jyothirmai, Dr.K.Subba Rao Published a international Journal on” Develop the Software Reliability using Software Reliability Improving Techniques” ” International Journal of Current Engineering Scientific Research (IJCESR), Vol:5 Issue: 4,ISSN No: 2394-0697(Online), 2393-8374(Print), April 2018(UGC Approved Journal).
19. D.Jyothirmai, Dr.K.Subba Rao Published a international Journal on “Improve The Software Reliability Using Software Reliability Growth Models and Reliability Activities” International Journal of Innovative Research in Technology(IJIRT) Vol:5 Issue:1 , ISSN : 2349-6002 ,June 2018(UGC Approved Journal).
- 20.D.Jyothirmai, Dr.K.Subba Rao Published a international Journal on “ Evaluate the Required level of Software Reliability Using Reliability Prediction” International Journal of Innovative Research in Technology(IJIRT) Vol:5 Issue:3 , ISSN : 2349-6002 ,Aug 2018(UGC Approved Journal).