

SOFTWARE METRICS AND RELIABILITY

*Mr. Deepak Kumar Mohanty.
Asst. Prof., Dept of IT, Damits,
Jagda, Rourkela
Email: deemohanty@gmail.com*

Abstract

According to ANSI, “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.

. The IEEE defines reliability as “The ability of a system or component to perform its required functions under stated conditions for a specified period of time.” So it is necessary that software reliability should be measured and evaluated. Though it is hard to achieve software reliability as we don’t have a good understanding of the software, it is always tempting to measure something related to reliability to reflect the characteristics. Reliability is a by-product of quality, and software quality can be measured. So in this paper we will demonstrate how quality metrics are used in evaluating the reliability of the software. Also this paper discusses different practices of software reliability measurement.

I .Introduction:

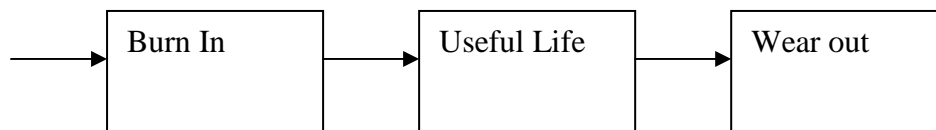
Software reliability is the probability that the software will work without failure for a specified period of time. Failure means the program in its functionality has no met user requirements in some way. Software reliability concerns itself with how well the software functions to meet the requirements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

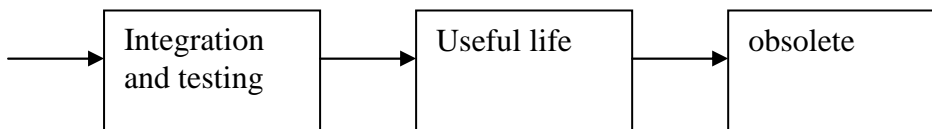
II .Software reliability and Hardware reliability:

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.



Phases of hardware when considering reliability



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.

III. 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 .

IV. Increasing the reliability and quality of the software

a) Faults and Failures :

What is a 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.

V. 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:-

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 uses two approaches to evaluate the reliability.

First, 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 evaluating the code , finding the errors and fixing them.

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

- 1) Product metrics
- 2) Project management busy
- 3) Process metrics
- 4) Fault and failure metrics

As discussed earlier 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.

Project Management metrics increases reliability by evaluating the Management process whereas process metrics can be used to estimate , monitor and improve the reliability and quality of the software.

The final one, Fault and Failure Metrics determines, when the software is performing the whole functions that are specified by the requirement documents with out any errors. It takes the faults and failures that arises in the coding and analyzes them to achieve this task.

VI. Conclusion:

For any software industry , 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.

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](http://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/