



International Journal of Allied Practice, Research and Review

Website: www.ijaprr.com (ISSN 2350-1294)

A Review of Software Reliability

¹Chandni Mahajan,

Teaching Assistant Government MAM College Jammu, Jammu and Kashmir, India

²Jai Kumar,

Teaching Assistant, GDC Samba, Jammu, Jammu and Kashmir, India

³Bindu Jamwal,

Teaching Assistant GDC Paloura, Jammu, Jammu and Kashmir, India

Abstract: According to ANSI, "Software Reliability is defined as software opportunities that may not fail for a period of time".

Software Reliability is different from Hardware Reliability. Although Software Reliability is defined as the probability of a time job, it is not a specific time function.

Gaining the credibility of the Software is difficult because the complexity of the software is often high. Various methods can be used to improve software reliability, however, it is difficult to estimate development time and budget and software reliability. But the best way to ensure the reliability of the software is to develop high quality software at all stages of the software life cycle. In this paper we will discuss how software metrics can improve the reliability and quality of software products.

IEEE defines trust as "the ability of a system or component to perform tasks required under certain specified circumstances." It is therefore necessary that the reliability of the software should be measured and tested. Although it is difficult to gain the reliability of the software as we do not fully understand the software, it is always tempting to measure something related to the integrity to reflect the features. Reliability is a quality product, and the quality of the software can be measured. So in this paper we will show you how quality metrics are used to test software reliability. And this paper discusses various ways to measure software reliability.

Keywords: Software Reliability, integrity Reliability analysis.

I. Introduction

Software reliability is an opportunity for the software to operate without fail for a specified period of time. Failure means that the system in its operation does not meet the user's needs in any way. Software reliability worries about how well the software works to meet customer needs.

Reliability represents a user-focused view of software quality. Initially, the quality of the Software was measured by counting errors in the system so this method is directed at the developer where reliability is inclined towards the user, because, it is related to performance rather than design.

II. Software reliability and Hardware reliability:

Software reliability is not a timely task. Hardware components may age and degenerate over time, but software will not change over time unless the software is modified or modified.

In the reliability of Hardware, in the first stage of production, there may be a high number of errors. But after finding and removing errors this number may decrease and gradually in the second phase (Healthy), there are only a few errors. After this stage, there will be an aging phase in which, part of the body ages with time and use and the number of defects will increase.

But in terms of software reliability, in the first stage, i.e. while testing and integration will have a high number of errors, but after removing errors, there are only a small number of errors and this process of debugging continues at a small rate. Software products will not wear out over time and use, but may become obsolete over time.

III. Need for Software Reliability Measurement

In any software industry, system quality plays a key role. Combines hardware quality and software quality. We know that hardware quality is always high .So if the system quality changes, it is only because of the software quality differences. Software quality can be measured in many ways. Reliability is a user-centered measure of "software quality".

Let us imagine that there are 3 systems that work to solve the problem. By determining the reliability of each system we can determine which system is less reliable and we can put more effort into repairing that system to improve the overall reliability of the system. So there is always a need to measure loyalty. In later sections we will discuss how to improve the reliability and quality of software.

IV. Increasing the reliability and quality of the software

a) Faults and Failures :

software failure:

The flow of external system performance results from the requirements. So failure is motivating. It depends on performance and behavior.

Differences between error and failure:

Failure is not the same as “bug” or “error”. There is a big difference between the two.

An error in the program occurs when the editor makes a mistake and causes a failure when it is done under certain circumstances.

b) Increasing Reliability

Reliability can be increased by preventing the errors and improving quality software at all stages of the software life cycle. We must ensure that the requirements clearly state the effectiveness of the final product or not. Among the reliability categories of software, the second is that the useful life is the most important so the software product should be carefully maintained. We must therefore ensure that the generated code can support storage to avoid any additional errors.

(Code category)

Next we must ensure that all the requirements mentioned in the requirement section are met or not met. (Test category)

Since reliability is a quality attribute, we can say that reliability depends on the quality of the software. So building a highly reliable software is necessary to measure the quality attributes used in each development cycle. Software metrics are used to measure these functional attributes. The following sections outline the different types of metrics used to improve system reliability.

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 should have. Therefore in this document of need, a clear understanding between the client and the developer should be present. If not it is important to document these requirements.

Requirements must have a valid structure to avoid the loss of important information.

Next, the requirements should be complete and detailed to facilitate the design phase. Requirements should not contain insufficient information.

The following is easy to communicate. There should not be any vague data on requirements. If there is any vague data, it will be difficult for the developer to apply that specification. The Requirement Reliability metrics checks the quality features mentioned above in the requirement document.

b) Design and Code Reliability Metrics

The quality features available in design and coding are complexity, size and modularity.

If there are complex modules, then it is difficult to understand and there is a high probability of errors. Therefore the complexity of the modules should be minimal. Next size, depending on factors such as line content, comments, usable statements etc. According to SATC, the most effective test is a combination of size and complexity. Reliability will decrease if the modules have a combination of high density and high density or high density and low density. In recent combinations and reliability decreases because, the smaller size results in shorter code that is harder to change.

These metrics also apply to object-directed code, but in this case, additional metrics are required for quality assurance..

c) Testing Reliability Metrics:

Test reliability metrics use two methods of reliability testing.

First, it ensures that the system is fully equipped with the functions specified in the requirements. As a result, errors due to lack of performance are reduced.

The second method is nothing more than checking code, finding errors and correcting them.

Current software to measure software reliability can be divided into four categories.

- 1) Product metrics
- 2) Project management is busy
- 3) Analyze metrics
- 4) Error and failure metrics

software size and complexity play an important role in compiling and coding. One of the product metrics called function point metric is used to measure the size and complexity of a system.

Project Management Metrics enhances reliability by evaluating the Management process while process metrics can be used to measure, monitor and improve software integrity and quality.

Lastly, Fault and Failure Metrics determines, in which the software performs all the functions specified in the required documents without errors. It takes errors and failures from coding and analyzes to achieve this function.

VI. Conclusion

In any software industry, gaining the trust of the software is a vital task. Gaining the credibility of the Software is difficult because the complexity of the software is often high. Reliability is a quality attribute and the quality of the software can be measured .So reliability depends on the quality of the software. So in each development phase, some quality features are used and the reliability and quality of the software can be improved by using software metrics in each development phase. These metrics measure the reliability of software Requirements, Design and coding, and test categories.

VII. 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/**