ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 5, May 2013

Review on Software and Hardware Reliability and Metrics

Kiranjit Kaur and Sami Anand

Abstract—Reliability is one of the important parts of any software that cannot be ignored and hard to measured. Software Reliability is defined as probabilistic function of time it is not a direct function of time. Various approaches can be used to improve the reliability of software and hardware. However it is hard to balance development time and budget with software reliability. Software does not have moving parts and does not physically wear out as hardware, but is does outlive its usefulness and becomes obsolete. Software reliability is not as well defined as hardware reliability, but the Software Assurance Technology Center (SATC) at NASA is striving to identify and apply metrics to software products that promote and assess reliability. But in this paper we are focus on some reliability metrics that define the reliability using some systematic form. In this paper we define two types of reliability like Hardware reliability and Software reliability. Software reliability is different from Hardware reliability. Software Reliability is defined as probabilistic function of time. In this paper we will discuss how software metrics can improve the reliability and quality of the software product. Software quality directly related to software reliability. Metrics used early can aid in detection and correction of requirement faults that will lead to prevention if errors later in the life cycle. 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. The IEEE defines reliability as "The ability of system or component to perform to perform its required functions under stated conditions for a specified period of time".

Index Terms **—Software, Software Reliability and Reliability Metrics**

I .INTRODUCTION

Reliability refers to the consistency of a number of measurements taken using the same measurement method on the same subject.

Kiranjit kaur,*Computer Science & Engineering Department, Lovely Professional University Lalandhar India. Lalandhar (Punjab) 9417382006 /*

Sami Anand*, Computer Science & Engineering Department, Lovely Professional University Lalandhar India. Jalandhar (Punjab) 9216757681*

If repeated measurements are highly consistent, then there is a high degree of reliability with the measurement method or the operational definition. According to ANSI, software Reliability is defined as: the probability in which failure free software operation for specified period of in a specified environment [1].

For the purpose of measuring the reliability of a system, the main problem which facing to gathering the accurate and complete failure data that is necessary for determine the reliability. The failure data is getting through a Product Service Organization, where users can report failures when they encounter them [2]. Reliability is an important part of software engineering, the software engineering use various type of parameters (knowledge) to improve the reliability, measurement is play an important role for measuring the reliability of any software, with the help of measure we can able to achieve a high reliability of software [3].

Software reliability related to itself with how well the software functions to meet the requirements of the customer. If the software fulfills the customer requirements then the software is reliable mean fewer Problems face to during the development and execution state [4]. This paper is organized as follows: Section II describes the related work for software Reliability, Section III review part and IV references.

II. RELATED WORK

Software Reliability is the key task for achieving the high reliability of any software industry. It applies the attributes that are helpful for achieving the reliability and it focus on metrics [4]. It is related to the real world problems in which the system is failed it use the reliability growth test reliability problems are found for the purpose of increasing the system reliability in Mean Time Between Failure (MTBF) [5].It present quantitative model in software reliability. It explains these model and their characteristics. It compares the models results on two different open source projects [6].It focus on each phase of the development life cycle, metrics identify the problem area how to lead to problems and errors. It decreases the cost and increase the reliability [7]. It describes the reliability analysis on observational data. It define how to analyze at the right level of precision [8].It examine the reproducibility of age –related muscle force production. Four force –time parameters are used [9].It define the various types of reasons that become the reason of unreliability. It tried to fill the gap by considering a three state repairable complex system with three types of failure using a mathematical model [10] .It focus on traditional and new object-oriented metrics for the purpose increasing the reliability [11].It analysis

component based software reliability. Which technique is used also known Scenario-Based Reliability Analysis [12].It present the approach to predicting software reliability based on some systematic identification of software process failure models and their likehoods13].It develops a new statistical software reliability tool which analyses the statistical software reliability and use a GUI tool [14].

III. REVIEW ON RELIABILITY

Software Reliability Definitions:

Formal definition:-Software Reliability is defined as the probability of failure free operation of computer program in a specified environment for a specified time**.**

Informal definition: - Reliability is a measure of how related system matches its starting acceptances.

Another Informal definition:-Software Reliability is defined as how the system fulfills the user requirements.

Software Reliability Vs Hardware Reliability

Software reliability- The software reliability not measured on the basis of time, because the software is never wear out. There is no problem of rust as like in case of hardware.

Hardware Reliability- Electronic and mechanical parts may become "Old" and wear out with time and usage. In the hardware reliability 'time' is used to define the reliability of hardware .It means how much time the hardware remain working without any defect.

Distinct Characteristics of Software and Hardware

Fault- Software faults are mainly design faults where as hardware faults are mostly physical.
Wear out-It is an important point, software remain reliable overtime instead of wearing out like hardware. It become obsolete (out of fashion) if the environment for which it is developed changes. Hence software may be retired due to environmental changes, new requirements, new expectations etc.
Software is not manufactured- A software is developed it is not manufactured like hardware. It depends upon the individual skills and creative abilities of the developer which is very difficult to specify and even more difficult to quantify and virtually impossible to standardize.
Time dependency and life cycle- Software reliability is not a function of operational time. But it is applicable on hardware reliability.
Reusability of components- In software there is graphical user interfaces are built using reusable components that are enable the creation of graphics windows, pull-down menus and wide variety of interaction mechanism. But in software GUI is not used, it used various types of devices. Good software has following features
-It should be fault free.
-It should posses' tolerance power.
-It should be user friendly.
-It should be within the budget of the customer.

Environmental Factors- Environment factors do not affect software reliability, but it affect to the hardware.

Basic Concept of Faults, failure, time and Failure Functions:

Failure Definition: It is the departure of the external results of program operation from requirements. So failure is dynamic. It depends upon the operation and behavior**.** A system is said to have a failure if the service it delivers to the user, not fulfill specification for a specified period of time
Detection of failure:
– How can you determine if there is a fault?
– Can we determine what fault caused the Failure?
Fault Definition: It is defect in the program that, when executed under particular condition causes a failure. The cause of the failure or the internal error is said to be fault.
Faults occur for many reasons:
– Incorrect requirements.
– Incorrect implementation of requirements.
– Unforeseen situations
Fault avoidance
-Use of information-hiding, strong typing, good Engineering, principles'.
Time: Reliability quantities are defined with respect to time, although it would be possible to define them with respect to other variables. We are concerned with three kind of time.
(i)Execution time
(ii)Calendar time
(iii)Clock time

Failure Functions: Can be expressed in several ways.
Cumulative failure function: It denotes the average cumulative failure associated with each point of time.
Failure functions: Failure can be expressed in several ways.
Failure intensity: Rate of change of the cumulative failure function.
Failure rate: The probability of failure in a per unit time is called failure rate.

Reliability Metrics: - Software reliability is defined as the probability of failure free operation of computer program in a specified environment for a specified time.
Important software reliability metrics include the following:

(i)MTTF (ii) MTBF (iii) AVAIL (IV) POFOD (v) ROCOF (VI) POFGO
Mean time to failure (MTTB):- It defines as the time interval between successive failures. MTTF of 200 means and average of 200time units passes between failures.
Mean Time Between Failure (MTBF):-It defined as the summation of MTTF and MTTR
AVAIL: - How much time the system continuously work, provides the availability of software. Availability of .998 means that system is available 998 out of 1000 time units.
Probability of failure on demand (POFOD):-It is defined as the probability that the system will when a service is required. A POFOD of indicates that at least one of 10 services requests will fail.

Rate of failure occurrence (ROCOF):-It is the number of failure occurring in unit time interval. A ROCOF of 0.02means 2 failure will occur for every 100 operational time units.
Probability of failure for a given output: - It is used to measure the probability of particular output, which is not met by the software. A lower probability of failure is desired for events with high priority.

Following steps to a Reliability Specification:

- Each subsystem, analyze the result of possible system failures.
- Some Partition failure in to appropriate classes.
- Each failure identified, describe the acceptable using an appropriate metrics.

IV.CONCLUSION

Software reliability is one of the key tasks for any software industry. Achieving Software reliability is hard because the complexity of software tends to be high. Metrics to measure software reliability do exit and can be used starting in the requirements phase. In each and every phase of the development life cycle of software the metrics can identify potential areas of problems that may lead to problems or errors. Hardware reliability is differing to software reliability. It is related to the wear out with time and usage. As hardware is become more and more cheap and powerful, software is become more and more complex and important in computer based information system.

V.REFERENCES

[1] Deepak *Pengoria, Saurabh Kumar,"*A Study on Software Reliability Engineering Present Paradigms and its Future Considerations" 2009.
[2] Pankaj Jalote, Brendan Murphy, Mario Garzia, Ben Errez, "Measuring Reliability of Software Products".
[3] Norman F. Schneidewind, "SOFTWARE RELIABILITY MEASUREMENT" R&M Engineering journal, volum23, number2, june 2003.
[4] Suman Kumar Kalavagunta,"Software Metrics and Reliability".
[5]Larry H.Crow"Demonstrating Reliability Growth Requirements with Confidence", 2012 IEEE.
[6] Cobra Rahmani, "Exploitation of Quantitative Approaches to Software Reliability,"2008.
[7] Dr.Linda Rosenberg,Ted Hammer and Jack Shaw, "Software Metrics and Reliability".
[8] Roland g.Jansen,Leon F.Wiertz,Erik S.Meyer and Lucas P.J..Noldus "Reliability analysis of observational data:Problem,Solutions, and Software implementation"2003.
[9] Michael G.Bemben, Benjamin H.Massey,Richard A.Boileau and James E.Misner, "Reliability of Isometrics Force-time Curve Parameters for Men Aged 20to 79Years,"journal of Applied Sport Science Research, Volume6, Number 3, 1992.
[10] Sachin Kumar and Anand Tyagi,"Evaluation of some reliability parameters of a three state repairable system with environmental failure,"international Journal of Research and Reviewa in Applied Sciences Volume, Issue 2 (November 2009).
[11] Dr.LindaH.Rosenberg, "Applying and Interpreting Object-Oriented Metrics".
[12] Sherif Yacoub,Bojan Cukic and Hany H.Ammar, "A Scenario-Based Reliability Analysis Approach for Component-Based Software,"2004 IEEE.
[13] Carol Smidts,Marith Stutzke and Robert W.Stoddard, "
[14]M.A.A.Boon,E.Brandt,ICorro Ramos,A.Di Bucchianico and R.Henezn, "A New Statistical Software Reliability Tool".

Kiranjit kaur M.tech, research work, Software Engineering.

Sami Anand Asst. Prof.(L.P.U.) M.tech ,member of IEEE&IEEE Computer Society, research work in area of Network Security and Software Engineering..