International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

Realization of Reliability for Data and Software to Improve Quality

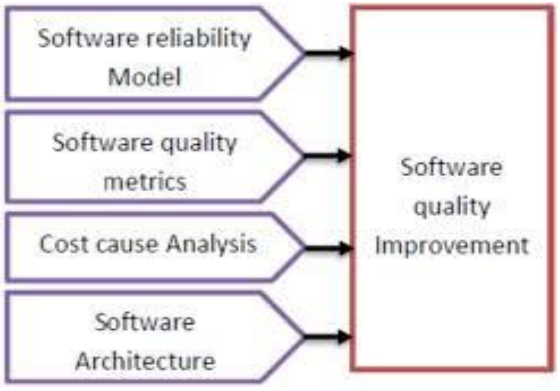
D.Naga Malleswari, Hemalatha Anumolu, Likitha Vampugadawala, Divya Sai Jonnalagadda

Abstract: ***The goal is to look for code performance metrics. Reliability is an important aspect of any program that cannot be ignored and difficult to measure. "Program reliability is defined as the probability of running programs without disruption in a specific environment for a specified period of time." The reliability of the technology differs from the performance of the hardware. Program reliability is difficult because the complexity of the program is high. Different methods can be used to increase system performance, but it is difficult to balance development time, budget, and software quality. But the best way to ensure technology consistency is to build high-quality programs throughout the life cycle of the program. We will discuss software reliability metrics in this paper. Metrics used early on can help detect and correct defects of requirements that will prevent program lifecycle errors later. It also provides consistency quality of the information system database with the help of RStudio, and we can also illustrate reliability based on the value of cyclomatic complexity and we can say whether the data or software is more reliable, less reliable or somewhat reliable.***

Keywords: Reliability, Consistency Quality, Cyclomatic Complexity

I. INTRODUCTION

Reliability can be described as the probability of an item for a specified period of time to perform a particular function under specific conditions. The reliability of a computer is described as the possibility that the free software will run in a specific environment for a specified period of time. Every material is unreliable due to system failure or errors. Uncertainty in the program is primarily due to software or design errors. Companies use scalable information retrieval systems as well as visualization and data analysis techniques. For more information about their companies and data analysis and modeling tools. These innovations represented a real-time work theory that uses current information and responds to events as they occur. This also applies to software engineering, where information analytics focuses on software development based on data, applications, and systems. Computer reliability is the possibility that for a specified period of time the computer will operate without failure. Reliability is a user-oriented view of the reliability of the code. Originally, the performance of code was calculated by counting the program's faults, so this practice is directed to the developer while consistency is user-oriented, as it is about operation rather than design. Reliability can be improved by avoiding failure of software development and performance through all stages of the technology life cycle. To do this, we need to make sure that the requirements clearly define the functionality of the final product (stage requirements). The second stage is among the reliability phases of the program, i.e. the most important is the useful life and care must be taken to maintain the product of the program. So, we must make sure that the generated code will support maintenance capability to avoid any other errors (coding phase). [1] First, we need to test whether all the specifications specified in the specification process have been met. (Test phase). Because reliability is a quality trait, we can conclude that reliability depends on the quality of the software. To build a highly reliable application, quality attributes added in each development cycle must be evaluated. To calculate these specific attributes, computer metrics are used. Specifications indicate what the code needs to contain. A clear understanding must therefore be established between the client and the developer of this requirements document. Otherwise, writing these specifications is critical. After that, the requirements should be comprehensive and detailed so that the design stage is easy. There should not be insufficient information about the requirements. Next is the interaction easily. Specifications should not contain any ambiguous information. In case of any unclear data, the programmer will have difficulty enforcing specifications.



Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
* Correspondence Author
D.Naga Malleswari, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.
HemaLatha Anumolu, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.
Likitha Vampugadawala, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.
DivyaSai Jonnalagadda, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.

Metrics	Reliability Specification
Probability of failure on demand(POFOD)	For systems where services request happens in an unpredictable way or when there is a long time interval between consecutive requests
Rate of occurrence of failures(ROCOF)	For systems when services are demand in more regular way
Mean time to failure(MTTF)	For systems involving long transactions, during which a guarantee of service continuity and delivery should be expected.
Availability(AVAIL)	For systems where continuous services delivery is a major concern

Fig: Metrics for reliability

Reliability is related to scale consistency. A student who performs a tool designed to assess motivation will have the same answers each time a test is taken. Although an accurate estimate of reliability cannot be given, reliability assessment can be achieved through various measures.

The details of the table below are the three performance attributes. The description explains how to evaluate each attribute. [2]

Using component comparison to total, half-split accuracy, Kuder Richardson's coefficient, and homogeneity are evaluated. The test or tool results are divided into half the reliability. Split in half. Comparing the two halves, the links are calculated. High links indicate high reliability, while the tool may not be accurate because of poor links. More complex version of the semester test is the Kuder-Richardson test. The average of all possible split half sets in this process is determined and a correlation between 0-1 is created. This test is more accurate than the half-chapter test, but with two answers can be completed only on questions.

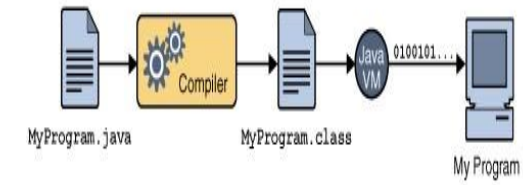
II. METHODOLOGY AND TECHNIQUE PROPOSED

Java programming language is an odd state language that mo st corresponding trend expressions will reflect:

Basic Architecture Unbiased
Article arranged Convenient
Dispersed High Performance
Multithreaded Powerful
Dynamic Secure

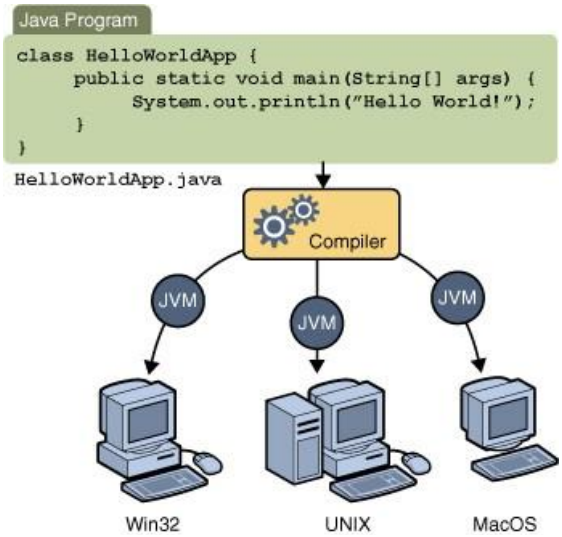
Step 1: The Java Language explains all past common articulations.

The source code is written in the Java programming language for the first time in flat substance repositories, ending with the.java extension. Such source documents are then assembled into. Class records by the javac compiler. A.class record does not contain code close to your processor; it contains bytecodes— the machine language of the Java Virtual Machine.



An overview of the software development process.

Because the Java VM is available to a wide range of operating frameworks, the same class records are suitable for running on Macintosh Operating System. For example, certain virtual machines, the Java Hotspot virtual machine, carry out additional steps at runtime to provide an introduction support for your application. It consolidates numerous activities, such as identifying execution bottlenecks and recompiling as often as possible.



A similar software is fit to run on multiple stages through the Java VM.

Step 2: R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R has a wide range of methods, both numerical and visual. It includes machine learning algorithm, linear regression, time series, statistical inference to name only a few. Most R libraries are written in R, but for heavy computational tasks, C, C++, and Fortran codes are preferred.**[5]**

R is not only costly for researchers, but R programming lang uage is also used by several large companies, like, etc. In a series of steps, data analyzes with R are performed; r esults are built, translated, found, modeled, and registered.

III. THEORETICAL ANALYSIS

• The concept of reliability is the accuracy of test results.

In theory, each assessment includes some errors - a component of test points that does not apply to the system you are trying to calculate. - The file may result from poorly designed test, deviations from the time the measure was measured by the student or marking the results of the assessment. [4]

• Reliability indexes attempt to determine the percentage of errors in the test score.
There are four ways to evaluate the reliability of the tool: -

Split-Half Reliability: Determine how much error is due to poor test construction in a test score.

To calculate: perform one test once, then calculate alpha, formula 20 KuderRichardson or formula Spearman Brown reliability index.

Reliability in retest test: Determines the amount of test score error due to test management issues (for example, subject interrupted due to too much noise).

Calculation: Take the same test for the same topics on two different occasions. Associate test scores for both departments in the same test.

Parallel Type Reliability: Determines the equivalence of two different versions of the same calculation.

To calculate: Take the two tests for the same participants in a short period of time. The test results are comparable between the two tests.

Inter-Rater reliability: Determines the accuracy of two different instrument raters.

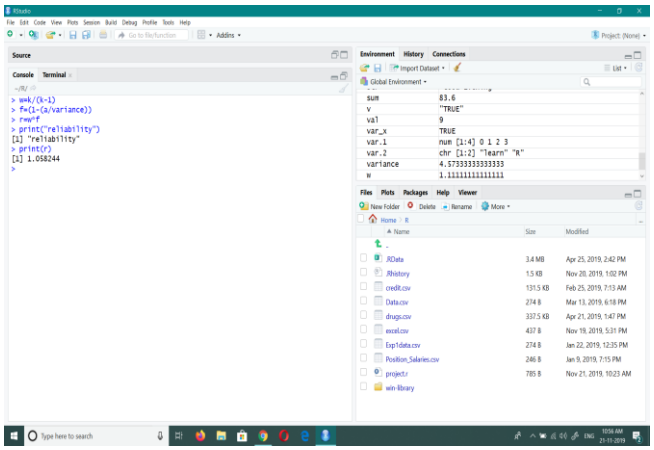
To calculate: to give the results of one test administration to two evaluators and to compare the two raters' points.
The Kuder and Richardson Formula 20 experiment is tested with internal binary consistency measurements. It is similar to a semi-divided technique for all question sets and is true when each question is true or incorrect. Score incorrect question 1 and 0 score incorrect question. The test statistic is
Formula: rKR20 = (k/k-1) (1-∑pq/σ^2)
– rKR20 is the Kuder-Richardson 20 equation
– k is the total number of test items
– p is the proportion of test takers passing the item
– q is the proportion of test takers failing the item
– σ2 is the variance of the whole test item.

IV. CAPABILITY STUDY

Secure: One of the fundamental sources of weakness in mod ern CGI is that shells that are uniformly useful to the job env ironment never execute the projects.In this way, the CGI software engineer must be cautious to sift through objects, such as back quotes and semicolons, which are treated by the shell exceptionally. Updating this safeguard is more difficult than one might expect, and vulnerabilities from this problem are always exposed in commonly used CGI libraries. A second source of problems is how some CGI projects are prepared by dialects that do not test cluster or string limits as a consequence. In C and C++, for example, it is highly acceptable to distribute a 100-component cluster and then compose it into the 999 th "component," which is an incredibly unusual piece of program memory. Software engineers who fail to perform this test open up their system to think about or accidental cradle flood assaults along these lines. Servlets are not experiencing the ill effects of any of these problems. It does not use a shell to do as well, irrespective of whether a servlet performs a system call to summon a program to the neighborhood work environment. Furthermore, obviously, cluster limits checking, and other memory assurance highlights are a focal piece of the Java programming language.

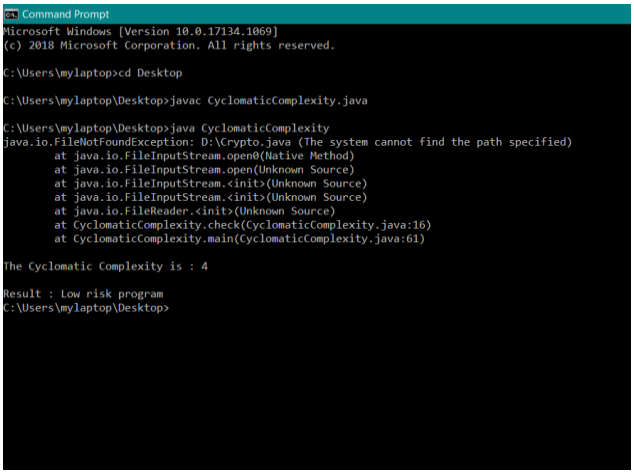
V. DISCUSSION OF RESULTS

1. Output for Kuder Richardson 20 formula



The Value of **r=1.05**, Therefore the dataset we taken is low reliable according to
Kuder Richardson 20 formula result gives the reliability value which checks the internal consistency
• If the r value ranges from 0 to 1.05, then it is low reliable.
• If the r value ranges from 1.06 to 2.98, then it is moderately reliable
• If the r value ranges from 2.99 to 4.0, then it is high reliable [3]

2. Output for Cyclomatic Complexity



The Value of **Cyclomatic Complexity =4(<10)**, Therefore the program we taken is at low risk according to
Cyclomatic Complexity value tells whether the program has low risk or moderate risk or high risk or complex risk.
• If the cyclomatic complexity value is 50 then it is most complex and highly unstable method
• If the value of cyclomatic complexity ranges between 21 and 50, this is a high-risk program.
• If the value of cyclomatic complexity ranges from 11 to 20, it will be a moderate risk program
• If the value of cyclomatic complexity is less than 10, then the program is low risk [3]

VI. RESULTS TAULAR FORM:

Value of r	Reliability	Result
0 - 1.05	Low	✓
1.06 - 2.98	Moderate	×
2.99 – 4.0	High	×

Cyclomatic Complexity	Risk	Result
>50	complex	×
21-50	High	×
11-20	Moderate	×
<10	Low	✓

VII. CONCLUSION

We conducted a case study to integrate quality management in technology analysis methods. For this we used RStudio Software for Continuous evaluation of system performance and improvement using computer analysis tools from industry, i.e. inclusiveness, consistency, effectiveness, appropriate level of detail, relevance, perceived value, and behavioral purpose. We can say whether the information system and computer data are low reliable, medium reliable or high reliable and also find how much risk of the program.

REFERENCES

1. Capucci, Federico, and Vijaygauri Gumaste." Test Case Writing (Creation)" Test University.
2. UTest, 16 Oct. 2013. Web. 17 Mar. 2014. http://help.utest.com/testers/crash-courses/functional/test-casewriting -creation-101 Rosenberg, Linda H., PhD, Theodore F. Hammer, and Lenore L. Huffman.
3. Essential QA Metrics for Determining Solution Quality White paper. Web. 17 march. 2014
4. Reporting Defect Trends and Status. Tool Mentor. Rational Software, 1998. Web. 17 Mar.2014.
5. http://www.interface.ru/rational/rup51/toolment/clearquest/tm_repdd. htm Hoffman,

AUTHORS PROFILE

D. Naga Malleswari, Computer Science and Engineering, Assistant Professor, Koneru Lakshmaiah Education, Vaddeswaram, India.

HemaLatha Anumolu, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.

Likitha Vampugadawala, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.

DivyaSai Jonnalagadda, Computer Science and Engineering, Koneru Lakshmaiah Education, Vaddeswaram, India.