ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

STUDY OF SOFTWARE QUALITY AND ITS ATTRIBUTES USING ERLANG DISTRIBUTION

Dr. P. K. Suri [*], Pooja

[*] CSE Department, Haryana College of Technology and Management, Kaithal, Haryana, India

ABSTRACT

Software quality is becoming an important part in software design, helping the designer to handle the complexity of large systems. While designing, the architect should analyze the system requirements before committing the resources to it. The analyzing process helps us to ensure the high quality of architecture design. For the past decade, there were many analyzing methods are used, which in turn to analyze only the views of single stakeholder. By doing so, there are many limitations that lead to critical situation in the development process. They elaborated this situation to excessive amount of time to perform the complete analysis. The scope of finding the key architectural decision is very difficult. Intend of these types of analysis gives the detailed information only after the designing phase, which makes the software unusable and not satisfied by the end-users. Generally, unusable software is useless. Customers and users won't accept unusable software, even if it provides the required features with the required operations.

KEYWORDS: Quality Attributes, Erlang distribution

INTRODUCTION

The first definition of quality history remembers is from Shewhart in the beginning of 20th century: There are two common aspects of quality: one of them has to do with the consideration of the quality of a thing as an objective reality independent of the existence of man. The other has to do with what we think, feel or sense as a result of the objective reality.

Software quality refers to some relations but distinct notions that exist wherever quality is defined in a business context:

Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software.

Software structural quality refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

QUALITY ATTRIBUTES

Software quality is defined as the degree to which software possesses a desired combination of attributes. The quality requirements to build the software architecture have to fulfill the stakeholders. They are commonly divided in two main groups based on the quality they are requesting, i.e., development and operational qualities. A **development quality requirement** is a requirement that is of importance for the developers work, e.g., maintainability, understandability, and flexibility. **Operational quality requirements** are requirements that make the system better from the user's point of view, e.g. performance and usability. Depending on the domain and priorities of the users and developers, quality requirements can become both development and operational, such as performance in a real-time system.

A quality attribute is the property of a software system. A quality requirement is a requirement that is placed on a software system by a stakeholder; a quality attribute is what the system actually presents once it has been implemented. During the development of the architecture it is therefore important to validate that the architecture has the required quality attributes, this is usually done using one or more architecture evaluations.

QULAITY ATTRIBUTES IN FOCUS

The focuses are on the following quality attributes: testability, portability, performance, maintainability,

Reliability, Usability, Correctness, Integrity, Interoperability, Reusability, Flexibility.

Testability: Testability is defined as: ―The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. The effort needed to validate the system against the requirements. A system with high testability can be validated quickly.

Portability: Portability is defined as: ―The ease with which a system or component can be transferred from one hardware or software environment to another. The portability is not only between different hardware platforms and operating systems, but also between different virtual machines and versions of frameworks.

Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. It is a multifaceted quality requirement. It incorporates aspects such as readability and understand ability of the source code. Maintainability is also concerned with testability to some extent, as the system has to be re-validated during the maintenance.

Performance: Performance is defined as: ―The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage. There are many aspects of performance, e.g., latency, throughput, and capacity.

Reliability: It is difficult to construct large software system which is correct. It might possible that few functions may not work in all scenarios, therefore software consider as incorrect. However the software still accepts able the failure rate is very small and it does affect so much.

Usability: It means software system is to be easy to use for human user. Normally client much emphasis on the user interfaces of software system.

Correctness: Software system is expected to satisfy its specification and fulfils the user's mission objectives.

Integrity: System integrity refers to extent to which access to software or secure data by unauthorised persons. It is play important role in network based application.

Interoperability: It is the ability of a system or different systems to operate successfully by communicating and exchanging information with other external systems written and run by external parties.

Reusability: It is the probability that a component will be used in other components or scenarios to add new functionality with little or no change. Reusability minimizes the duplication of components and the implementation time.

Flexibility: It is reflected in the cost of modifying an operational system.

QUALITY ATTRIBUTES AND THEIR DEFINITIONS

Correctness: The ability of a system to perform according to defined specification.

Robustness: Appropriate performance of a system under cases not covered by the specification.

Extendibility: A system that is easy to adapt to new specification.

Reusability: Software that is composed of elements that can be used to construct different applications applications.

Compatibility: Software that is composed of elements that can easily combine with other elements.

Efficiency: The ability of a system to place as few demands as possible to hardware resources, such as memory, bandwidth used in communication and processor time.

Portability: The ease of installing the software product on different hardware and software platforms.

Timeliness: Releasing the software before or exactly when it is needed by the users.

Integrity: Releasing the software before exactly when it is needed by the users.

Verification and validation: How easy it is to test the system.

Ease of use: The ease with which people of various backgrounds can learn and use the software.

Maintainability: The ease of changing the software to correct defects or meet new requirements.

Performance: Low utilization of resources, lower response time and mean time of failure and recovery define the performance of the system.

Cost effectiveness: The ability of a system to be completed within a given budget.

ERLANG DISTRIBUTION

The shorthand X _ Erlang(k, μ) is used to indicate that the random variable X has the Erlang distribution with positive shape parameter k and positive scale parameter a. An Erlang random variable X with rate μ and k stages has probability density function

$$f(x)=\frac{x^{k-1}e^{-x/\mu}}{\mu^{k}(k-1)!} \quad x>0 \qquad (1)$$

The Erlang distribution can be used to model the time to complete k operations in series, where each

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

operation requires an exponential period of time to complete. The probability density function for various values of *k* and μ is illustrated below.

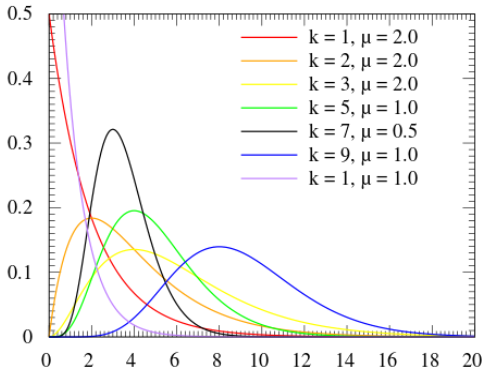

Figure 1: Probability Density Function

PROBABILITY DENSITY FUNCTION OF QULAITY FACTORS USING ERLANG DISTRIBUTION

Erlang-m distributed random variable is the sum of m independent and identically distributed exponential distribution. Its density function can be shown to be

$$H_m(t)= \frac{1}{(m-1)!}.(\frac{1}{\beta})^m.t^{m-1}.e^{-t/\beta} \qquad (2)$$

where β is the average value of each of the m constituent exponential distributions, and therefore mβ is the average value of the Erlang Distribution.

When m=1 , Above equation becomes

$H_1(t)=\frac{1}{\beta}.e^{-t/\beta}$ the exponential distribution.

Suppose there are 6 attributes and the average time is 3 days. Then the composite time or total service time to complete the project is 18 days.

ALGORITHM TO GENERATE RANDOM NUMBERS USING ERLANG DISTRIBUTION

1. Initialise int (m, n, i, j) and float(mean, beta, erlang, r, x)
2. Apply for loop on i, j.
3. Initialise prod=1.0
4. Calculate random number using
   R=rand()/32768.0
   prod=prod*r
5. Then take x=log(prod)
6. Erlang=mean*(-1.0/m)*x
7. Print all generated random numbers.

REFERENCES

[1] Staffordshire University, (*Software Quality: Definitions and Strategic Issues*), School of Computing Report, Ronan Fitzpatrick, April 1996. Websites

[2] Abran, A., Jacquet, J-P.(1997). From Software Metrics to Software Measurement Methods: A Process Model, *Third International Symposium and Forum on Software Engineering Standards(ISESS97)*, Walnut Creek CA, June 2-6.

[3] Arthur, L.J.(1992). *Improving Software Quality: an insider's guide to TQM*. John Wiley & Sons.

[4] System Simulation with Digitl Computer by Narsingh Deo

[5] Https://books.google.com/about/software-engg