Cloud based Application Development and Various Software Quality Model

Dr. Ganga Shirisha M S.
Asst Prof, Dept of Computer Science,
SSA Govt. First Grade College(Auto), Ballari,
Karnataka. (India)

Abstract:- A cloud application (or cloud app) can be a hybrid application and can be used for Android, IOS and window, with some characteristics of a pure desktop app and some characteristics of a pure Web app. A desktop app resides entirely on a single device at the user's location (it doesn't necessarily have to be a desktop computer). The quality of a paper can be evaluated on the basis of its complexity, response time and total cost. So that the quality measurement process consumes too much time. There are a number of models already introduced for measuring software quality based on its attributes. Such measurements help developer to improve software in next development. Such models also helps in reducing cost and time in next development. The main characteristics of the cloud models have covered in this paper.

Keywords: *Software Quality, Software Quality Models, Quality assurance plan.*

*****

I. INTRODUCTION

The term cloud application development to develop an multipurpose app for hybrid platform and the term software quality refers to the study of functions of software and measuring how well the software conforms to that design. The purpose of such evaluation is to provide better software in next development process. In this process the software requirements are the base from which the software quality is measured. The following shows a number of factors such as research, analysis, prototype of software, design, testing, planning, delivery and support. All the steps comes in a particular time and feedback of such process helps to measure quality of software. A basic model of measuring factors is given in the figure below:

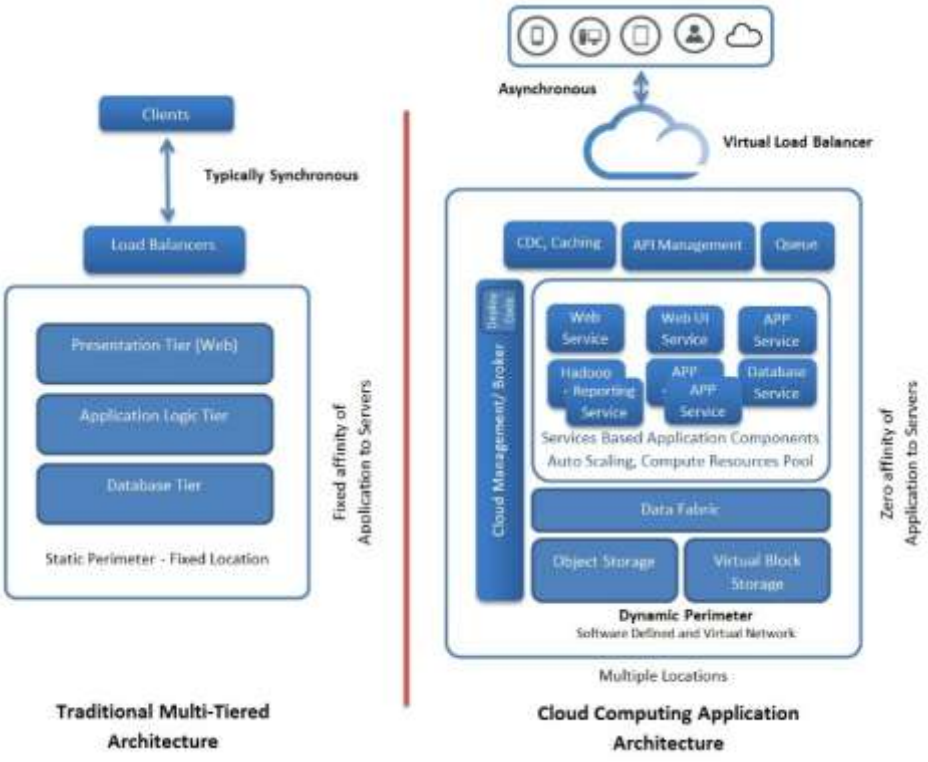

Figure 1: Comparative study of Cloud based Multi Tired Architecture and Traditional Architecture[1]

According to the control, assessment and other aspects, the software quality is a set of attributes of software and is defined as:

- Software Degree meets user expectations
- Feedback of user
- Security issues of software
- Execution Time
- Response time

Quality of an software is always measured from the view of customer not the developer/ engineer. As once the user or customer uses the software he can give exact feedback by using software. Software quality measurement is about quantifying to what extend a system or software process desirable characteristics. The dependencies tree between software quality characteristics and their measurement attributes is represented in the diagram given below:

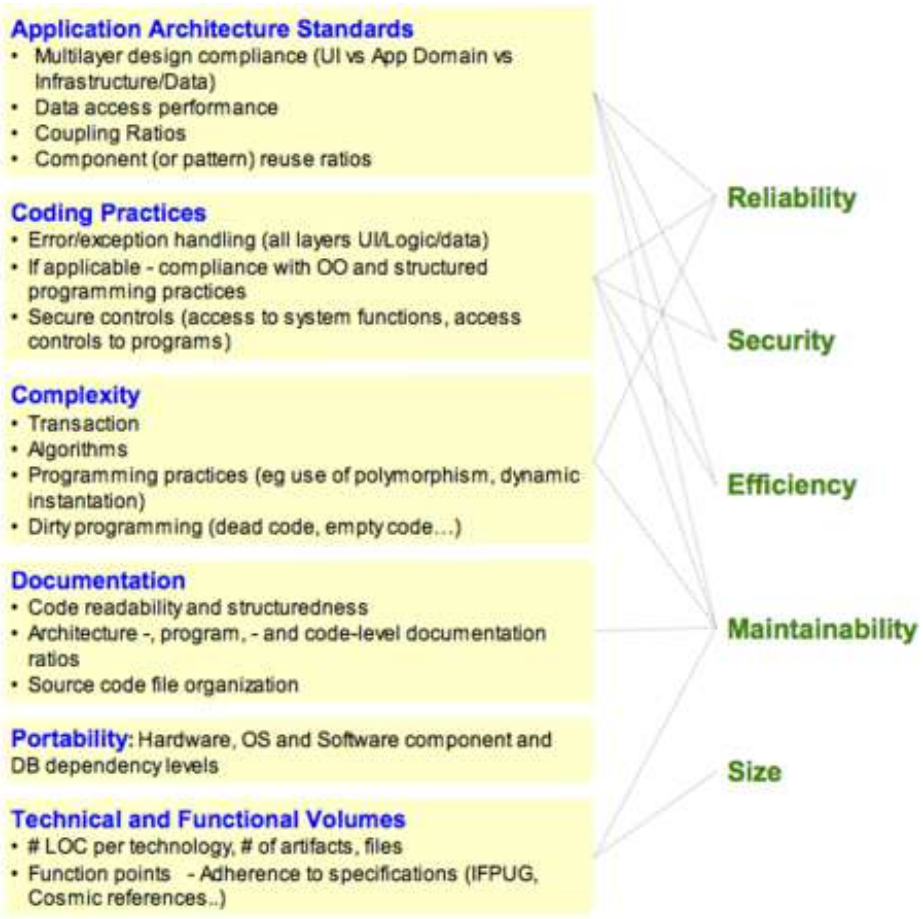

Figure 2: Relationship between software desirable characteristics (right) and measurable attributes (left).

II. MEASUREMENT

Quality assurance testing continuously uncover the measurements of software and helps to reduce the cost of fixing issues and bugs. Metrics have always been used to help guide managers with decisions about their organizations. However as technologies and methodologies have evolved, metrics must evolve as well. Defect tracking, for example, has traditionally been a metric used to measure software quality throughout the lifecycle. However, Agile methodologies suggest that pre-production defect tracking may, in fact, be detrimental to software teams. It can be difficult to determine which are the best for giving a true picture of software quality. This guide will provide insights for CIOs, senior managers and project managers in selecting the right set of metrics to effectively and efficiently manage their software development organizations. [3]

Some factors involved in measuring quality of software are given below: Some practices we rely on to address quality:

- Acceptance tests for every story (we use user stories for requirements)
- Automated integration tests.
- Automated performance tests.

- Code reviews (all our code is reviewed using a tool)
- Feature Level Tests (testing groups of stories)
- Manual testing for each story, immediately after it is implemented.
- Risk tracking
- Unit Tests.

III. QUALITY ASSURANCE MODEL AND PHASES

The software quality assurance is categorized into four phases:

1. **Initiation phase:** The requirement of software is gathered including cost and time measurement.
2. **First Review phase:** Another important phase analyzing the feedback of phase first.
3. **Repetition phase:** This phase check regular testing of the various software versions.
4. **Control phase:** Within this phase the software is examined to check it is ready for releasing. [3].

Depending on your definition quality is highly subjective, and therefore difficult to measure. As the project is being developed, it will be difficult to measure "How well the product meets the consumers needs".

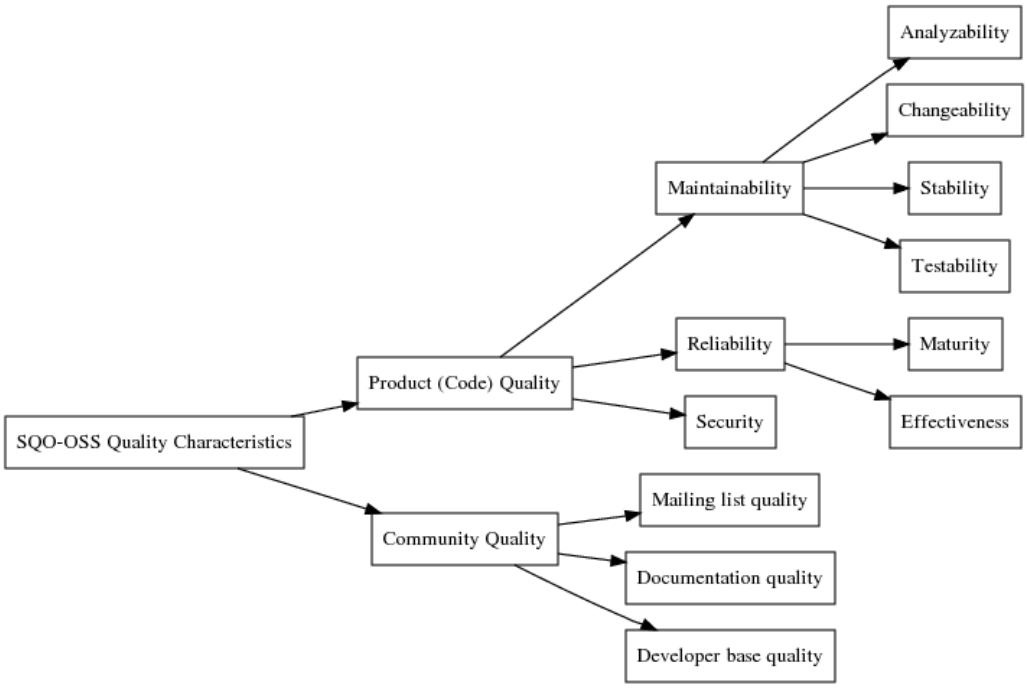

Figure: Software quality model as tree[4]

One of the traditional measures of quality is the bug detection rate. This is relatively easy to measure, and somewhat less subjective. It is believed that a relatively fixed percentage of bugs will go undetected. Projects with a high rate of detected bugs will likely have a high rate of undetected bugs. The rate of bug detection will vary depending on the techniques used and skill of your testers.There are a variety of bug tracking tools which can be used. Most will allow you to track bug counts over time. Aiming for zero bugs at all times may improve overall quality. One method to do this is "Fix bugs first". This may have multiple benefits, including lowering overall bug rate.

IV. CONCLUSION

In this paper many software quality models are studied. Different models use different approach in relation to functional and non functional requirements. But reliability is most important factor in software quality measurement. It should be noticed that phases like requirements, coding implementation should be done in proper manner to have quality.

REFERENCES

[1] http://www.kandasoft.com/home/softwaredevelopment/quality-assurance.html
[2] https://en.wikipedia.org/wiki/Software_quality
[3] http://searchsoftwarequality.techtarget.com/guides/Quality-metrics-A-guide-to-measuring-software-quality
[4] http://www.programmingresearch.com/blog/measuring-metrics-in-safety-critical-systems/