Software Quality.

Ortiz Vega Angelo
Course Code: CE2103
Name: Algorithms and Data Structures II,
Academic Area of Computer Engineering.
Cartago, Costa Rica.

-Keywords: Software Quality, ISO 9126, ISO 25010, Testing, Software Requirements.

-Content:

Defined by ISO 9126, ISO 25010: 2011.
a. Good practices to tie common problems.
b. Quality: degree which the requirements (functional and non-functional) are accomplished.
c. Defines metrics focused on the product (reliability, efficiency, usability, maintainability, portability)
d. Related to:

1. QA: Quality Assurance, improves the process.
2. QC: Quality Control, check the product "testing"

SOFTWARE QUALITY ATTRIBUTES APPROACH

This approach to software quality is best exemplified by fixed quality models, such as ISO/IEC 25010:2011. This standard describes a hierarchy of eight quality characteristics, each composed of sub-characteristics:

1. Functional suitability
2. Reliability
3. Operability
4. Performance efficiency
5. Security
6. Compatibility
7. Maintainability
8. Transferability

Additionally, the standard defines a quality-in-use model composed of five characteristics:

1. Effectiveness
2. Efficiency
3. Satisfaction
4. Safety
5. Usability
6.

A fixed software quality model is often helpful for considering an overall understanding of software quality. In practice, the relative importance of particular software characteristics typically depends on software domain, product type, and intended usage. Thus, software characteristics should be defined for, and used to guide the development of each product.

Testing:

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

Principles of Testing:

(i) All the test should meet the customer requirements

(ii) To make our software testing should be performed by third party

(iii) Exhaustive testing is not possible.As we need the optimal amount of testing based on the risk assessment of the application.

(iv) All the test to be conducted should be planned before implementing it

(v) It follows pareto rule(80/20 rule) which states that 80% of errors comes from 20% of program components.

(vi) Start testing with small parts and extend it to large parts.

Types of Testing:

1. Unit Testing

It focuses on the smallest unit of software design. In this we test an individual unit or group of interrelated units.It is often done by the programmer by using sample input and observing its corresponding outputs.

2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design.Integration testing is testing in which a group of components are combined to produce output.

Integration testing are of two types: (i) Top down (ii) Bottom up

3. Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

4. Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing

It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.

5. Alpha Testing

This is a type of validation testing.It is a type of*acceptance testing* which is done before the product is released to customers. It is typically done by QA people.

6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment

7. System Testing

In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.

In this we have security testing, recovery testing , stress testing and performance testing

8. Stress Testing

In this we gives unfavorable conditions to the system and check how they perform in those conditions.

9. Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system.It is used to test speed and effectiveness of program.