

## **e-PG Pathshala**

**Subject : Computer Science**

**Paper : Software Quality Management**

**Module : Software Quality Assurance**

**Module No: CS/SQM/21**

**Quadrant 1 : e-text**

### **Software Quality Assurance**

The requirements document is one of the most important elements for achieving software quality. There is a need for a comprehensive definition of requirements that will cover all attributes of software and aspects of the use of software including:

- **Usability aspects**
- **Reusability aspects**
- **Maintainability aspects.**

The great variety of issues related to the various attributes of software and its use and maintenance can be classified into content groups called Quality Factors. Software requirement documents are expected to differ in the emphasis placed on the various factors, a reflection of the differences to be found among software projects. Thus, we can expect that not all the factors will be universally “represented” in all the requirements documents.

The classic model of software quality factors, suggested by McCall, consists of 11 factors.

McCall’s factor model

- **Product operation factors:**  
Correctness, Reliability, Efficiency, Integrity, Usability.

- **Product revision factors:**  
Maintainability, Flexibility, Testability.
- **Product transition factors:**  
Portability, Reusability, Interoperability.

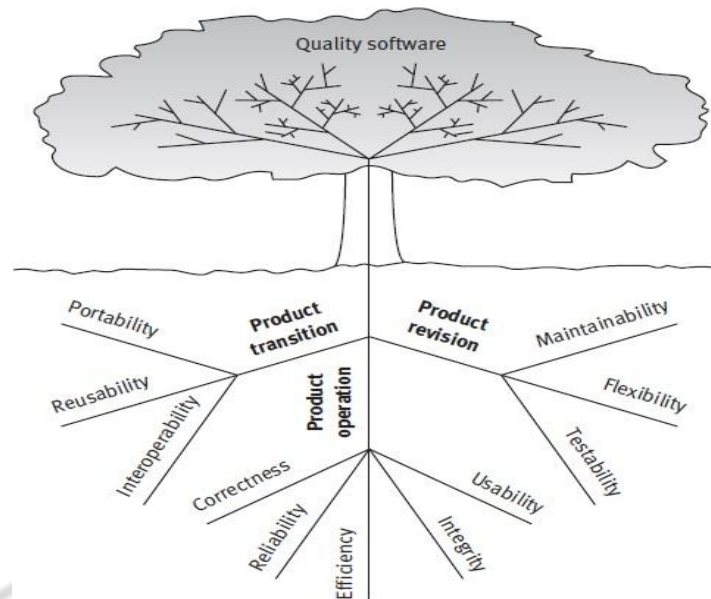


Figure 1: McCall's factor model tree

According to McCall's model, five software quality factors are included in the product operation category, all of which deal with requirements that directly affect the daily operation of the software.

- **Correctness**
- **Reliability**
- **Efficiency**
- **Integrity**
- **Usability**

**Correctness** is the extent to which a program satisfies its specifications and fulfills the user's mission objectives. Correctness requirements are defined in a list of the software system's required outputs.

**Example:**

The correctness requirements of a club membership information system can be the following:

**The output mission:** A defined list of 11 types of reports, four types of standard letters to members and eight types of queries, which were to be displayed on the monitor on request. The completeness of the output information: The probability of missing data about a member, his attendance at club events, and his payments must be zero.

**Reliability** can be defined as an extent to which a program can be expected to perform its intended function with required precision. Reliability requirements deal with failures to provide service. They determine the maximum allowed software system failure rate.

**Example:**

The failure frequency of a heart-monitoring unit that will operate in a hospital's intensive care ward is required to be less than one in 20 years. Its heart attack detection function is required to have a failure rate of less than one per million cases.

**Efficiency** deals with the hardware resources needed to perform all the functions of the software system. The requirements include the maximum values at which the hardware resources will be applied in a software system.

**Example:**

An outdoor meteorological unit, equipped with a 1000 milli-ampere hour cell, should be capable of supplying the power requirements of the unit for at least 30 days. The system performs measurements once per hour, logs the results, and transmits the results once a day to the meteorological center by means of wireless communication.

**Integrity** requirements deal with the software system security. The requirements to prevent access to unauthorized persons. Distinguish between the majority of personnel allowed to see the information.

**Example:**

The Engineering Department of a local municipality operates a Geographic Information System and is planning to allow citizens access to its GIS files through the Internet. The software requirements include the possibility of viewing and copying but not inserting changes in the maps of their assets as well as any other asset in the municipality's area. Access will be denied to plans in progress and to those maps defined by the Department's head as limited access documents.

**Usability** is the effort to learn, operate, prepare input and interpret output of a program. Deals with the scope of staff resources needed to train a new employee and to operate the software system.

**Example:**

**Software usability requirements document for the new help desk system:**

- (a) A staff member should be able to handle at least 60 service calls a day.
- (b) Training a new employee will take no more than two days (16 training hours), immediately at the end of which the trainee will be able to handle 45 service calls a day.

According to the McCall model of software quality factors, three quality factors comprise the product revision category. These factors deal with those requirements that affect the complete range of software maintenance activities:

- **Maintainability**
- **Flexibility**
- **Testability**

**Maintainability** is the effort required to locate and fix an error in an operational program. It determines the efforts needed by the users to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.

**Example:**

- a) The size of a software module will not exceed 30 statements.
- (b) The programming will adhere to the company coding standards and guidelines.

**Flexibility** is the Effort required to modify an operational program. The capabilities required to support adaptive maintenance activities are covered by the flexibility requirements. Include the resources required to adapt a software package to a variety of customers of the same trade, of various extents of activities, of different ranges of products etc.

**Example:**

Teacher support software deals with the documentation of pupil achievements, the calculation of grades, the printing of term grade documents, and the printing of warning letters to parents of failing pupils. The flexibility requirements are: The software should be suitable for teachers of

all subjects and all school levels. Non-professionals should be able to create new types of reports according to the schoolteacher's requirements.

**Testability** is the effort required to test a program to ensure that it performs its intended function.

It deals with the testing of an information system as well as with its operation. Testability requirements software operation include automatic diagnostics performed by the software system prior to starting the system. It finds out whether all components of the software system are in working order and obtain a report about the detected faults.

**Example:**

An industrial control unit is programmed to calculate various measures of production status, report the performance level and operate a warning signal in predefined situations. One testability demanded was to develop a set of standard test data with known system expected correct reactions in each stage. This standard test data is to be run, before production begins, to check whether the unit reacts properly.

According to McCall, three quality factors are included in the product transition category, a category that pertains to the adaptation of software to other environments and its interaction with other software systems.

The Various Product Transition Software Quality Factors are:

- **Portability**
- **Reusability**
- **Interoperability**

**Portability** is the effort required to transfer a program from one hardware configuration or software system environment to another. Makes it possible to continue using the same basic software/hardware in diverse situations.

**Example:**

A software package designed and programmed to operate in a Windows 2000 environment is required to allow low-cost transfer to Linux and Windows NT environments.

**Reusability** is the extent to which a program can be used in other applications. Deals with the use of software modules originally designed for one project in a new software project currently being developed. The reuse of software saves development resources, shortens the development period, and provides higher quality modules.

### **Example:**

A software system is required for the operation and control of a hotel swimming pool that serves hotel guests and members of a pool club. Although the management did not define any reusability requirements, the unit's team leader, after analyzing the information processing requirements of the hotel's spa, decided to add the reusability requirement that some of the software modules for the pool should be designed and programmed in a way that will allow its reuse in the spa's future software system, which is planned to be developed next year

These modules will allow:

- Entrance validity checks of membership cards and visit recording.
- Restaurant billing.
- Processing of membership renewal letters.

**Interoperability** is the effort required to couple one system with another. Focuses on creating interfaces with other software systems or with other equipment firmware. Interoperability requirements can specify the name(s) of the software or firmware for which interface is required.

### **Example:**

The firmware of a medical laboratory's equipment is required to process its results (output) according to a standard data structure that can then serve as input for a number of standard laboratory information systems.

Two factor models, appearing during the late 1980s, considered to be alternatives to the McCall classic factor model.

- **The Evans and Marciniak factor model (Evans and Marciniak, 1987).**
- **The Deutsch and Willis factor model (Deutsch and Willis, 1988).**

**Verifiability** defined in Evans and Marciniak requirements define design and programming features that enable efficient verification of the design and programming. Most verifiability requirements refer to modularity, to simplicity, and to adherence to documentation and programming guidelines.

**Expandability** defined in both the models (Evans and Marciniak, and Deutsch and Willis) refers to future efforts that will be needed to serve larger populations, improve service, or add new applications in order to improve usability.

**Safety** defined in (Deutsch and Willis) are meant to eliminate conditions hazardous to operators of equipment as a result of errors in process control software. These errors can result in

inappropriate reactions to dangerous situations or to the failure to provide alarm signals when the dangerous conditions to be detected by the software arise.

**Manageability** in (Deutsch and Willis) refers to the administrative tools that support software modification during the software development and maintenance periods, such as Configuration Management and Software Change Procedures.

**Survivability** defined in (Deutsch and Willis) refer to the continuity of service. Defines the minimum time allowed between failures of the system, and the maximum time permitted for recovery of service. These requirements may refer separately to total and to partial failures of services, they are especially geared to failures of essential functions or services.

**Reusability** Requirements are defined in cases where the client anticipates development in the near future of an additional software system having strong similarities to the present software, the client will himself initiate reusability requirements. The client is interested in reusing parts of software systems that were developed earlier in a new system.

**Verifiability** Requirements are meant to improve the design reviews and software tests carried out during software development. Their aim is to save development resources and they are, therefore, of interest to developers. The client, however, is usually uninterested in placing requirements that deal with the internal activities of the developer team.

### **Summary**

Issues related to the various attributes of software and its use and maintenance can be classified into content groups called Quality Factors. McCall's classic factor model classifies quality factors into three categories namely Product operation factors, Product revision factors and Product transition factors. Two factor models The Evans and Marciniak factor model and the Deutsch and Willis factor model are considered to be alternatives to the McCall classic factor model.