

## **Hierarchical models of Boehm and McCall:**

In 1978, B.W. Boehm introduced his software quality model. The model represents a hierarchical quality model similar to McCall Quality Model to define software quality using a predefined set of attributes and metrics, each of which contributes to overall quality of software.

The difference between Boehm's and McCall's model is that McCall's model primarily focuses on precise measurement of high-level characteristics, whereas Boehm's quality model is based on a wider range of characteristics.

### **Example –**

Characteristics of hardware performance, that are missing in McCall's model.

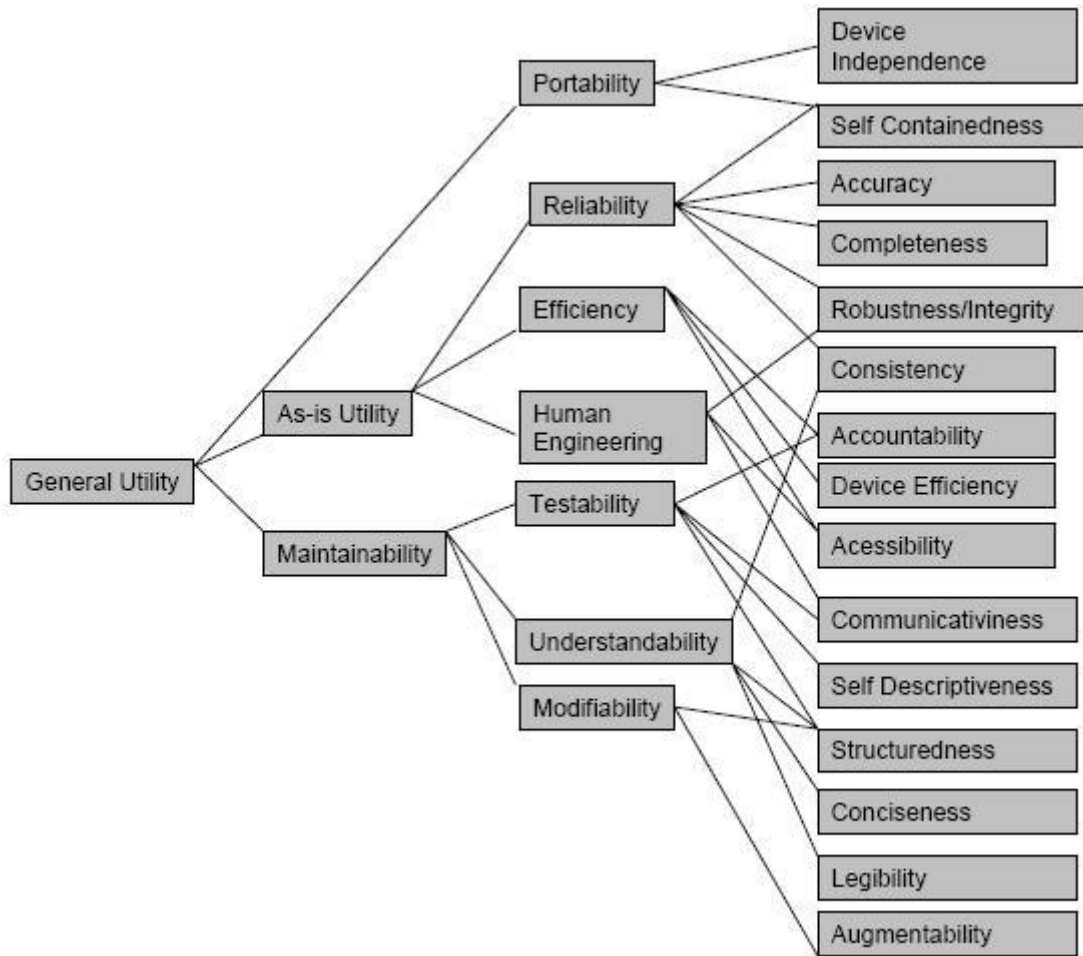
The Boehm's model has three levels for quality attributes. These levels are divided based on their characteristics. These levels are primary uses (high level characteristics), intermediate constructs (mid-level characteristics) and primitive constructs (primitive characteristics).

The highest level of Boehm's model has following three primary uses stated as below –

1. **As is utility** –  
Extent to which, we can use software as-is.
2. **Maintainability** –  
Effort required to detect and fix an error during maintenance.
3. **Portability** –  
Effort required to change software to fit in a new environment.

The next level of Boehm's hierarchical model consists of seven quality factors associated with three primary uses, stated as below –

1. **Portability** –  
Effort required to change software to fit in a new environment.
2. **Reliability** –  
Extent to which software performs according to requirements.
3. **Efficiency** –  
Amount of hardware resources and code required to execute a function.
4. **Usability (Human Engineering)** –  
Extent of effort required to learn, operate and understand functions of the software.
5. **Testability** –  
Effort required to verify that software performs its intended functions.
6. **Understandability** –  
Effort required for a user to recognize logical concept and its applicability.
7. **Modifiability** –  
Effort required to modify a software during maintenance phase.



Boehm further classified characteristics into Primitive constructs as follows- device independence, accuracy, completeness, consistency, device efficiency, accessibility, communicativeness, self-descriptiveness, legibility, structuredness, conciseness, augmentability. For example- Testability is broken down into:- accessibility, communicativeness, structuredness and self descriptiveness.

**Advantages:**

- It focuses and tries to satisfy the needs of the user.
- It focuses on software maintenance cost effectiveness.

**Disadvantages:**

- It doesn't suggest, how to measure the quality characteristics.
- It is difficult to evaluate the quality of software using the top-down approach.

## McCall Model:

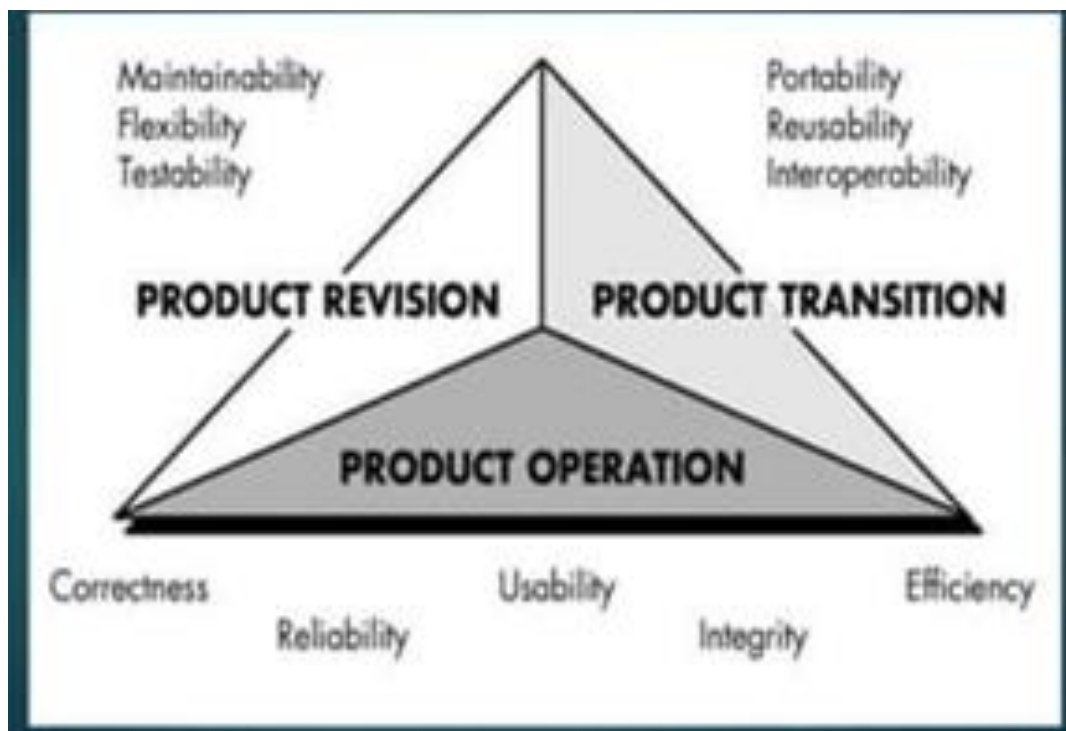
McCall software quality model was introduced in 1977. This model is incorporated with many attributes, termed as software factors, which influence a software. The model distinguishes between two levels of quality attributes :

### 1. Quality Factors

The higher level quality attributes which can be accessed directly are called quality factors. These attributes are external attributes. The attributes in this level are given more importance by the users and managers.

### 2. Quality Criteria

The lower or second level quality attributes which can be accessed either subjectively or objectively are called Quality Criteria. These attributes are internal attributes. Each quality factor has many second level of quality attributes or quality criteria.



### 1. Product Operation :

It includes five software quality factors, which are related with the requirements that directly affect the operation of the software such as operational performance, convenience, ease of usage and its correctness. These factors help in providing a better user experience.

- **Correctness** –  
The extent to which a software meets its requirements specification.
- **Efficiency** –  
The amount of hardware resources and code the software, needs to perform a function.
- **Integrity** –  
The extent to which the software can control an unauthorized person from the accessing the data or software.
- **Reliability** –  
The extent to which a software performs its intended functions without failure.
- **Usability** –  
The extent of effort required to learn, operate and understand the functions of the software.

## 2. Product Revision :

It includes three software quality factors, which are required for testing and maintenance of the software. They provide ease of maintenance, flexibility and testing effort to support the software to be functional according to the needs and requirements of the user in the future.

- **Maintainability** –  
The effort required to detect and correct an error during maintenance phase.
- **Flexibility** –  
The effort needed to improve an operational software program.
- **Testability** –  
The effort required to verify a software to ensure that it meets the specified requirements.

## 3. Product Transition :

It includes three software quality factors, that allows the software to adapt to the change of environments in the new platform or technology from the previous.

- **Portability** –  
The effort required to transfer a program from one platform to another.
- **Re-usability** –  
The extent to which the program's code can be reused in other applications.
- **Interoperability** –  
The effort required to integrate two systems with one another.