

# SOFTWARE QUALITY MANAGEMENT

# INTRODUCTION

## ■ UNIT I : INTRODUCTION TO SOFTWARE QUALITY.

Software Quality – Hierarchical models of Boehm and McCall – Quality measurement – Metrics measurement and analysis – Gilb's approach – GQM Model.

## ■ UNIT II: SOFTWARE QUALITY ASSURANCE.

Quality tasks – SQA plan – Teams – Characteristics – Implementation – Documentation – Reviews and Audits

## ■ UNIT III : QUALITY CONTROL AND RELIABILITY.

Tools for Quality – Ishikawa's basic tools – CASE tools – Defect prevention and removal – Reliability models – Rayleigh model – Reliability growth models for quality assessment

## ■ UNIT IV : QUALITY MANAGEMENT SYSTEM.

Elements of QMS – Rayleigh model framework – Reliability Growth models for QMS – Complexity metrics and models – Customer satisfaction analysis.

## ■ UNIT V : QUALITY STANDARDS.

Need for standards – ISO 9000 Series – ISO 9000-3 for software development – CMM and CMMI – Six Sigma concepts.

# OBJECTIVES

- Define Quality.
- Define software quality.
- Features of Quality software.
- Characteristics of software quality.
- Techniques for improving software quality.
- Views of quality.
- Hierarchical models of quality.
- Quality measurements.

# Quality

Source	Definition
OED(Oxford English Dictionary), 1990	The degree of excellence
Crosby, 1979	Zero defects
ISO, 1986	The totality of features and characteristics of a product/service that bear on its ability to satisfy specified/implied needs.

# Cont...

- Ability of the product/service to fulfill its function.
- Hard to define.
- Impossible to measure.
- Easy to recognize in its absence.
- Transparent when present.

## Cont...

- Quality is defined as the products or services capability to meet customer expectations.
- Quality that is defined as a matter of products and services whose measurable characteristics satisfy a fixed specification.
- Quality is defined as conformance to requirements.

## Cont...

- ❑ Quality consists of those product features which meet the need of customers and there by provide product satisfaction.
- ❑ Quality is multidimensional. some aspects of quality can be measured.

Example: Maximum speed, fuel, economy etc.

# Software quality

- Kitchen ham (1989 b) refers to software quality “fitness for needs” and claims quality involves matching expectations.

Two features of a piece of quality software:

- Conformance to its specification
- Fitness for its intended purpose.

The Department of Defense (DOD, 1985) in the USA defines software quality as “the degree to which the attributes of the software enable it to perform its intended end use”.



# Cont...

**Software was particularly problematical for the following reasons:**

- Software has no physical existence.
- The lack of knowledge of client needs at the start.
- The change of client needs over time.
- The rapid rate of change on both hardware and software.
- The high expectations of customers, particularly with respect to adaptability.
- Within the software quality area, the need to provide a solution that matches user needs is often considered as “design quality”, whilst ensuring a match to the specification is considered as “manufacturing quality”.

# External characteristics

- **Correctness**- Degree to which system is free from faults in specification, design and implementation.
- **Usability**- The Ease with which users can learn and use the system.
- **Efficiency**- Minimal use of system resource including memory and execution time.
- **Reliability**- The ability of a system to perform whenever required without/with few failures.
- **Integrity**- Prevention of unauthorized or improper use.
- **Adaptability**- Usability in other application than the original one.
- **Accuracy**- Degree of “quantitative” correctness.
- **Robustness**- Functioning of system in presence of invalid inputs, stress environment.

# Internal Characteristics

- **Maintainability:** Ease of modifying software for changing/adding capabilities, improving performance.
- **Flexibility:** Extend of modifying system for other uses/environments.
- **Portability:** Ease of modifying system for operating in different environment.
- **Reusability:** Extend of using parts in other systems.

# Cont...

- **External:**

What a systems user is interested in; typically properties of any single particular system.

- **Internal:**

What programmers/management are interested in; properties of the development of a collection of systems.

# Techniques for improving SQ

- Explicit software quality objectives.
- Explicit quality assurance activities.
- Testing strategy.
- Software Engineering guidelines.
- Informal technical reviews.
- Formal technical reviews.
- External audits.
- Development process.

# Cont...

- ❑ Change control procedures.
- ❑ Measurement of results.
- ❑ Prototyping.
- ❑ Mathematical proof.
- ❑ Modular programming techniques.

# Views of Quality

- Quality is a multidimensional construct. It may therefore be considered using a polyhedron metaphor. Within this metaphor, a three-dimensional solid represents quality. Each face represents a different aspect of quality such as **correctness**, **reliability**, and **efficiency**.
- It has been classified according to a number of 'views' or perspective. These views are often diverse and may conflict with each other. Each view comes from a particular context.

# The software project Roles

- Project manager
- Business analyst
- Implementation programmer
- Quality auditor
- End user
- Line manager
- Project sponsor



# Cont...

## Views of Quality

### User

What I Want  
Fast response  
Control information  
Easy to use help menus  
Available as required  
Exception data  
Reacts to business change  
Input data once

### Designer

Good Specification  
technically correct  
Fits within systems structure  
Easy to maintain  
Difficult for user to manage  
Fast development  
Low maintenance  
well documents

# Garvin Classification

- 1. The transcendent view
  - Innate excellence
  - Classical definition
- 2. The product-based view
  - Higher the quality higher the cost
  - Greater functionality
  - Greater care in development

# Cont...

- 3. The user-based view
  - Fitness for purpose
  - Very hard to quantify
- 4. The manufacturing view
  - Measures quality in terms of conformance
  - Zero defects
- 5. The value-based view
  - Provides the data with what the customer requires at a price.

# HIERARCHICAL MODELS OF QUALITY

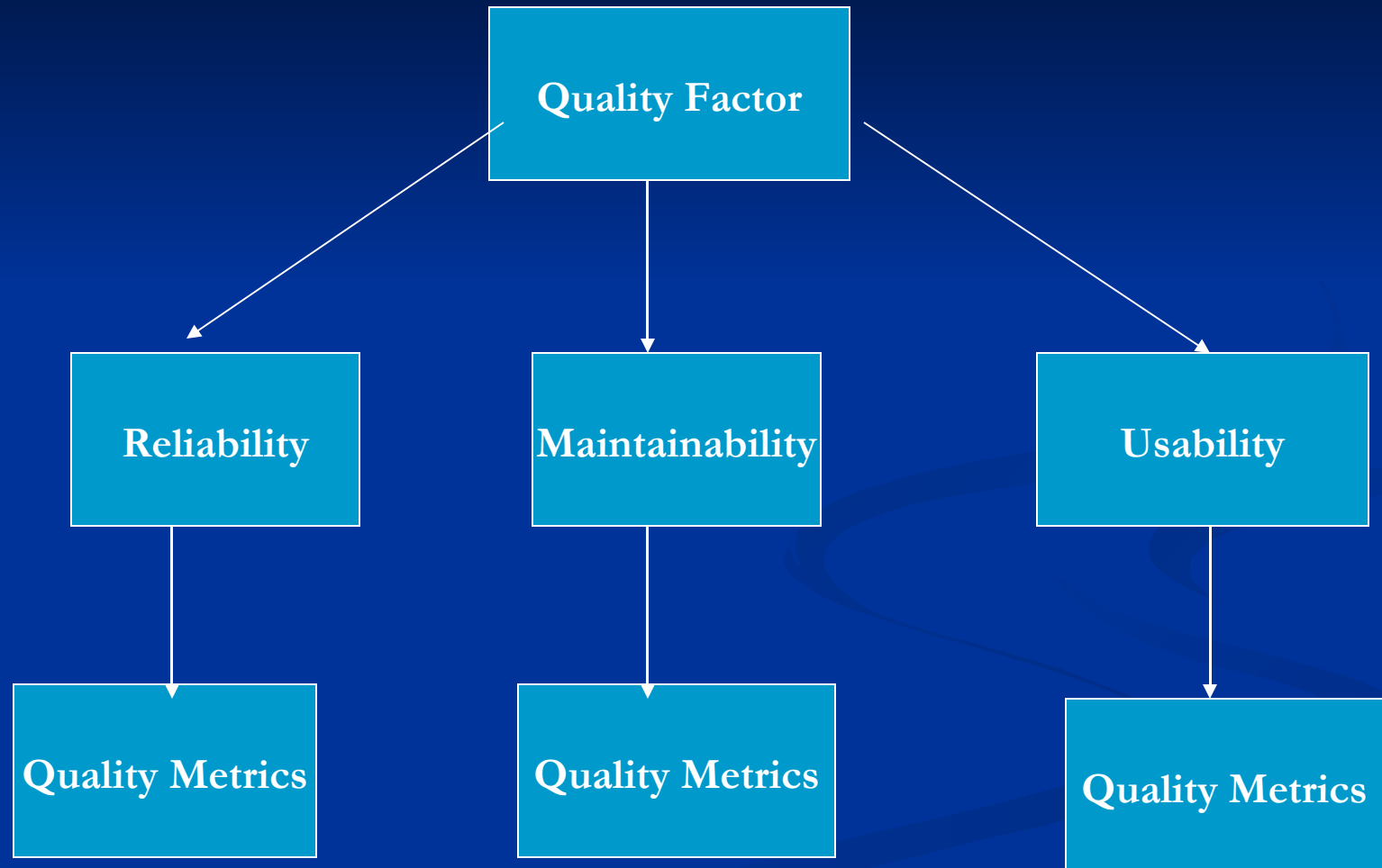
- A hierarchical model of software quality is based upon a set of quality criteria, each of which has a set of measures.

Quality model are,

Bohem model.

McCall model.

# HIERARCHICAL VIEWS OF SQ



# GE model (MacCall 1977&1980)

- McCall model

- 1978, James McCall
- Project manager at General Electric

Three area addressed by MacCall model.

Product operation.

Product revision.

Product transition.

# Cont...

## Product Operation:

Here factors are related to the Operational performance, convenience, ease of usage and correctness.

## Product Revision:

These factors pertain to the Testing and Maintainability of Software. It gives idea about maintenance, flexibility and Testing effort

## Product Transition:

To transfer a product from one platform to another platform or from one technology to another technology.

# Cont....

**Product Transition**

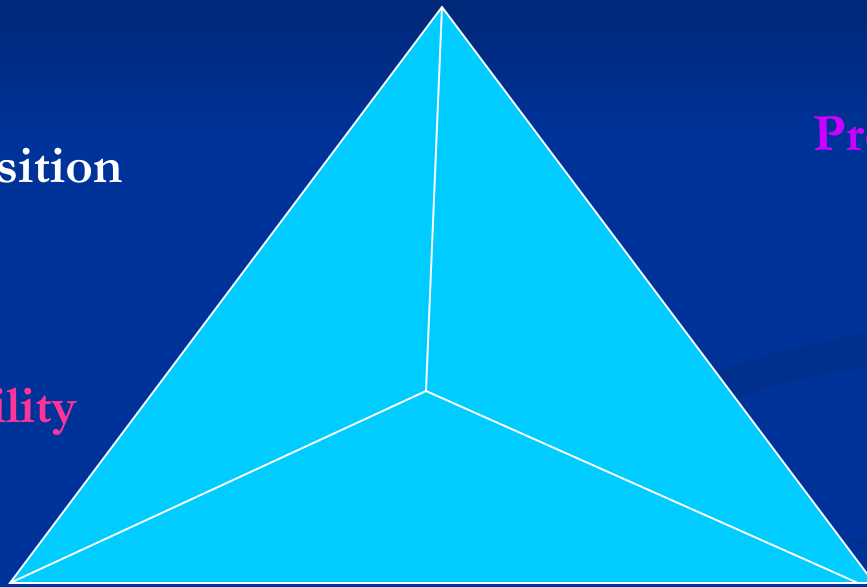
**Portability**  
**Reusability**  
**Interoperability**

**Product Revision**

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

**Product Operation**

**Correctness, Efficiency, Integrity, Reliability, Usability**





# Cont...

- Usability: Ease of use of software.
- Integrity: Protection of the program from unauthorized user access.
- Efficiency: concerned with the resources.  
(ex): Process time, storage.
- Correctness: The extent to which the program fulfills its specifications.
- Reliability: Ability not to fail.

# Cont...

- Maintainability: The effort required locating an  
fixing a fault in the  
program with its  
operating environment.
- Flexibility: Ease of making changes with its  
operating environment.
- Testability: Ease of testing the program to  
ensure that it is error free.

# Cont...

- Portability: The effort to transfer a program from one environment to another.
- Reusability: The ease of using the software in different context.
- Interoperability: It is the effort required to couple the system to another system.

# Boehm Model

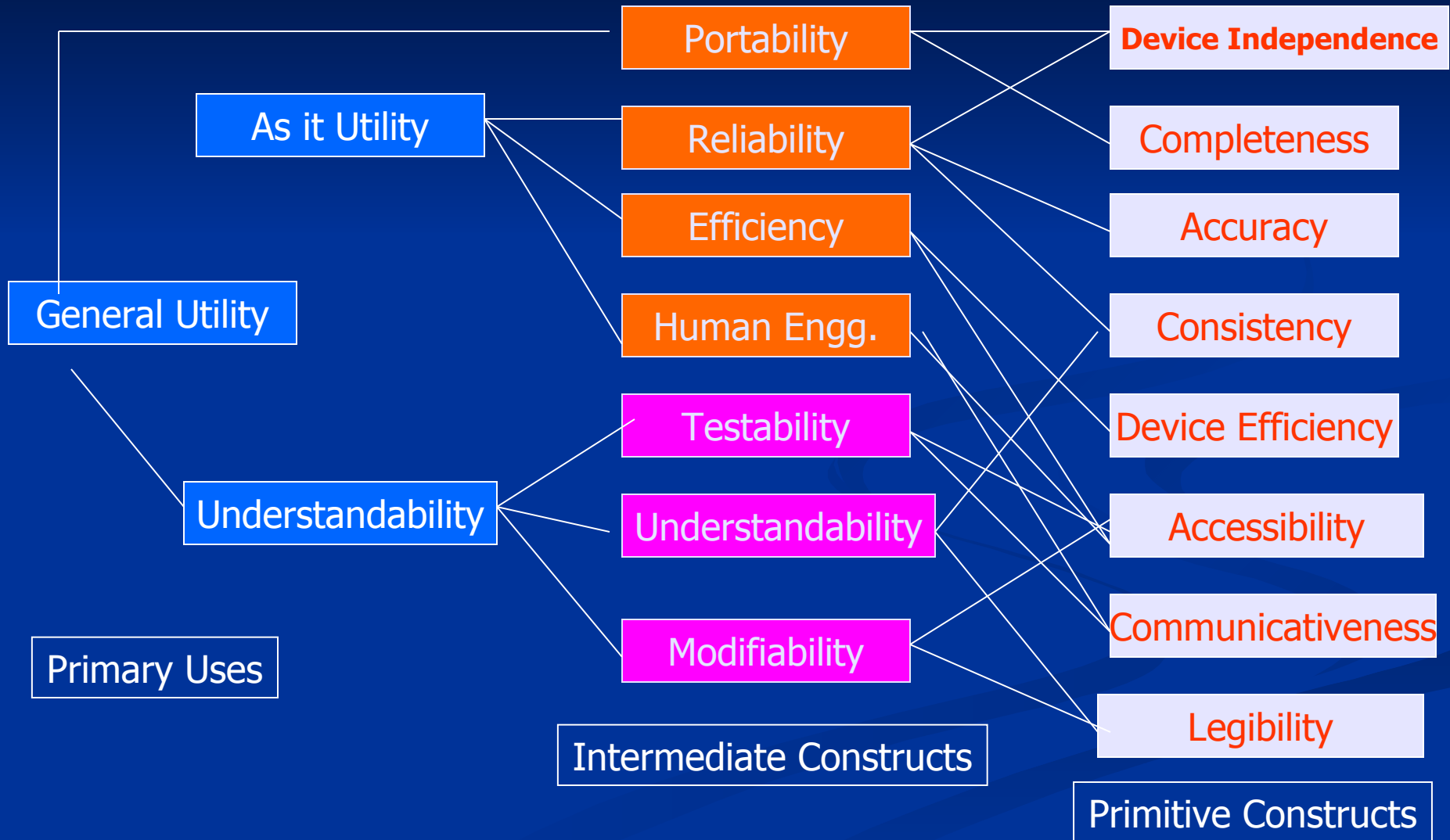
- The Boehm model

- 1977, Barry Boehm (  
[http://sunset.usc.edu/Research\\_Group/barry.ht](http://sunset.usc.edu/Research_Group/barry.ht)  
)
- a TRW Systems and Energy Inc. (“Thompson RamoWooldridge”) software research lab.

# Boehm's Software Quality Model

- Introduced in 78. Boehm has defined three levels of quality attributes:
  - Primary uses
  - Intermediate constructs
  - Primitive constructs
- This model is similar to Mc Call but also includes hardware performance that are missing in Mc Call.
- Model reflects:
  - What user want with S/w Product
  - Uses of Resources
  - Is Software easy to learn and use
  - Is well designed, well coded, easily tested & Maintained

# Cont...



# ISO-9126

- Single Model to standardize the Quality factors since 1980.
- It makes easier to compare products.
- It encompasses Six major attributes contributing the Quality

**Functionality:** Characteristics related with the achievement of purpose

**Reliability:** Capability of Software to maintain the performance of S/w

**Usability:** Effort required to use the software

**Efficiency:** Relationship b/w level of performance & amount of resources

**Maintainability:** Effort needed to make modification, improvement

**Portability:** Transfer of One software to another one









































