

Benghazi University  
Faculty of Information Technology  
**Software Engineering Department**

# ***Software Quality***

## ***Part2***

### ***(Software Quality Definitions & Concepts)***

***Prepared by: Mohamed A. Elorfi***

***Instructor: Salah Abdelsattar***

# Quality—A Pragmatic View

The *transcendental view* argues that quality is something that you immediately recognize, but cannot explicitly define.

The *user view* sees quality in terms of an end-user's specific goals. If a product meets those goals, it exhibits quality.

The *manufacturer's view* defines quality in terms of the original specification of the product. If the product conforms to the specification, it exhibits quality.

The *product view* suggests that quality can be tied to inherent characteristics (e.g., functions and features) of a product.

Finally, the *value-based view* measures quality based on how much a customer is willing to pay for a product.

In reality, quality includes all of these views and more.

# Quality definitions

- **Webster's Definition of Quality**

*The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs*

- **IEEE**

1. *The degree to which a system, component, or process meets specified requirements.*
2. *The degree to which a system, component, or process meets customer or user needs or expectations.*

- **Juran, (Manufacturer's Perspective)**

*Software quality is fitness for intended use*

- **Phil Crosby, (Management's perspective)**

*"Quality means "conformance to user requirements".*

- **Edwards Deming, (Consumer's perspective)**

*"striving for excellence" in reliability and functions by continuous improvement in the process of development, support by statistical analysis of the causes of failure.*

- **Watts Humphrey**, of the SEI,  
*“achieving excellent levels of fitness for use, conformance to requirements, reliability and maintainability.”*
- **Barry Boehm**,  
*“Achieving high levels of user satisfaction, portability, maintainability, robustness, and fitness for use”.*
- **Pressman**  
*Conformance to:*
  1. *explicitly stated functional and performance requirements,*
  2. *explicitly documented development standards, and*
  3. *implicit characteristics that are expected of all professionally developed software.*

- **James Martin**,  
“software quality means being on time, within budget and meeting user needs”
- **Tom McCabe**, the software complexity specialist,  
“high level of user satisfaction and low defect levels, often associated with low complexity”
- **John Musa** of Bell Laboratories,  
“combination of low defect levels, commitment of software functions to users needs, and high reliability”
- **Bill Perry**, head of Quality Assurance Institute  
“high levels of user satisfaction and adherence to requirements”.

## **Comment:**

It is apparent from a first look at these definitions that quality is a subjective term with many definitions – covering different perspectives of various stakeholders. However, a closer look and analysis of these definitions from the perspective of intent reveals that they all essentially pointing in the same direction – “quality is non-negotiable”. It is therefore important, for the organizations - to manage quality using a defined and structured approach to quality management.

## **Why business should be concerned with Quality?**

- Customers demand quality.
- Quality is critical for success.
- Quality is competitive issue now
- Quality gives you the global reach
- Quality is cost effective
- Quality helps retain customers and increase profits
- Quality is the hallmarks of world-class business

## Importance of Software Quality

- Software is a major component of computer systems
  - (about 80% of the cost)
- Software is used for
  - communication (e.g. phone system, email system)
  - health monitoring,
  - transportation (e.g. automobile, aeronautics),
  - economic exchanges (e.g. ecommerce),
  - entertainment,
  - etc.
- Software defects are extremely costly in term of
  - money
  - reputation
  - loss of life



## **Software Quality**

- Is achieved through a disciplined approach - called software engineering.
- Can be defined, described, and measured.
- Can be assessed before any code has been written.

## **Software quality challenges**

- Defining it.
- Describing it (qualitatively).
- Measuring it (quantitatively).
- Achieving it (technically).

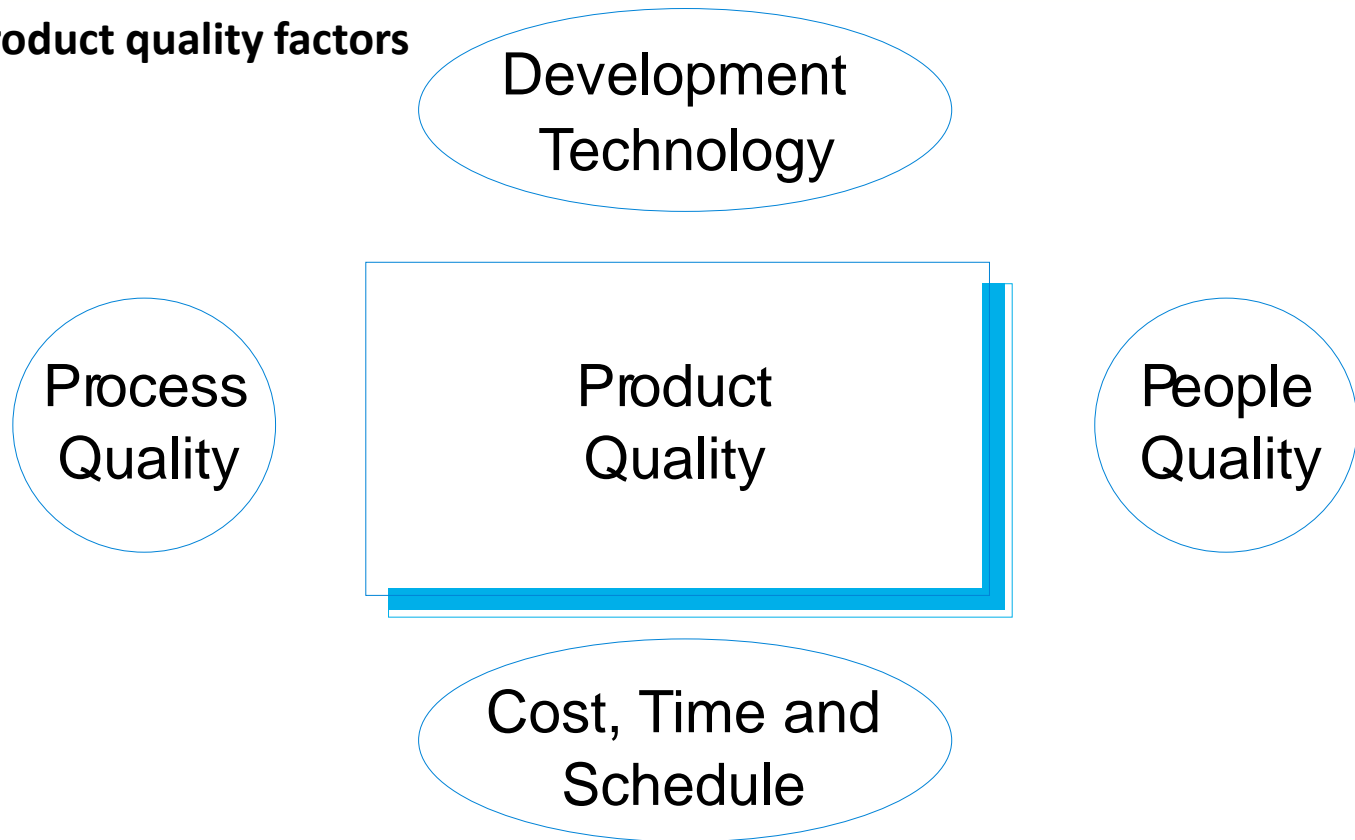
## Process and Product Quality

- ❖ **Product:** delivered to the customer
- ❖ **Process:** produces the software product
- ❖ **Underlying assumption:** a quality process leads to a quality product
- ❖ Process quality and product quality are closely related and process improvement benefits arise because the quality of the product depends on its development process.
- ❖ A good process is usually required to produce a good product.
- ❖ For manufactured goods, process is the principal quality determinant.
- ❖ For design-based activity, other factors are also involved especially the capabilities of the designers.

## What is product quality?

- Quality, simply, means that a product should meet its specification
  - The software product should deliver the required functionality (*functional requirements*) with the required **quality attributes** (*non-functional requirements*)

### Software product quality factors



# Attributes of quality software product

- ***Auditability***

The ease with which conformance to standards can be checked.

- ***Accuracy***

The precision of computations and control. A qualitative assessment of freedom from error. A quantitative measure of the magnitude of error. The correct data values are recorded.

- ***Communication commonality***

The degree to which standard interfaces, protocols, and bandwidths are used.

- ***Completeness***

The degree to which full implementation of required function has been achieved. All data items are captured and stored for use. Data items are properly identified with time periods.

- ***Conciseness***

The compactness of the program in terms of lines of code.

- ***Consistency***

The use of uniform design and documentation techniques throughout the software development project.

- ***Data commonality***

The use of standard data structures and types throughout the program.

# Attributes of quality software product

- ***Error tolerance***

The damage that occurs when the programs encounters an error. Suitable error prevention and detection procedures are in place. There are procedures for reporting and correcting errors. Various audit procedures are applied.

- ***Execution efficiency***

The run-time performance of a program.

- ***Expandability***

The degree to which architectural, data, or procedural design can be extended.

- ***Generality***

The breadth of potential application of program components.

- ***Hardware independence***

The degree to which the software is decoupled from the hardware on which it operates.

- ***Instrumentality***

The degree to which the program monitors its own operation and identifies errors that do occur.

- ***Modularity***

The functional independence of program components.

- ***Operability***

The ease of operation of a program.

# Attributes of quality software product

- ***Robustness***

The extent to which software can continue to operate correctly despite the introduction of invalid inputs.

- ***Security***

The availability of mechanism that control or protect programs and data. The system and its operations are protected from various environmental and operation risks. There are provisions for recovery in the event of failure or destruction of part or all system.

- ***Self-documentation***

The degree to which the source code provides meaningful documentation.

- ***Simplicity***

The degree to which a program can be understood without difficulty.

- ***Software system independence***

The degree to which the program is independent of nonstandard programming language features, operating system characteristics, and other environmental constraints.

- ***Traceability***

The ability to trace a design representation or actual program component back to requirements.

- ***Training***

The degree to which the software enables new users to apply the system.