# Software Quality Engineering

Week 1

Engr. Muhammad Umer Haroon

# FOLLOW THE RULES

- Use of Mobile (no)
- Late coming    (no)
- Short Attendance (no)
- Plagiarism (no)
- Cheat (no)
- Disrespect (no)

# Course Content

- **Software Quality**, Software Quality Attributes, Quality Engineering., **Testing**: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing.

# Recommended Books

- Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed. CRC Press, Taylor and Francis Group, 2015

- Bernard Homes, Fundamentals of Software Testing, ISTE, Wiley, 2012

- Software Engineering, "Ian Sommerville, 9th Edition, Addison Wesley, 2011

# Evaluation Criteria

| Assessment | Weightage |
|---|---|
| Assignment | 10% |
| Quiz | 10% |
| Sessional 1 | 15% |
| Sessional 2 | 15% |
| Final Exam | 50% |

# Generic Flow

- **concepts**
- **Quality Engineering at Requirement Phase**
- **concepts**
- **Quality Engineering at Architecture Phase**
- **concepts**
- **Quality Engineering at Design Phase**
- **concepts**
- **Quality Engineering at Code Phase**
- **concepts**

# What is Software Quality ?

According to the IEEE

- Software quality is:
    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.*

# What is Software Quality?

- "Achieving high levels of user satisfaction, portability, maintainability, robustness, and fitness for use" by Dr. Barry Boehm.

- Quality means "conformance to user requirements" by Phil Crosby.

- Edwards Deming considers quality to be "striving for excellence" in reliability and functions by continuous improvement in the process of development, support by statistical analysis of the causes of failure.

# What is Software Quality?

- Watts Humphrey, of the SEI, tends to speak of quality as "achieving excellent levels of fitness for use, conformance to requirements, reliability and maintainability."

- James Martin said that software quality means being on time, within budget and meeting user needs.

- Tom McCabe, the software complexity specialist, defines quality as "high level of user satisfaction and low defect levels, often associated with low complexity.

# What is Software Quality?

- John Musa of Bell Laboratories states that quality means combination of "low defect levels, adherence of software functions to users needs, and high reliability"

- Bill Perry, head of Quality Assurance Institute has defined quality as "high levels of user satisfaction and adherence to requirements".

# Importance of Software Quality

- Software is a major component of computer systems
- (about 80% of the cost)
- – 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

# What is Software Quality Assurance?

According to the IEEE

Software quality assurance is:

1. *A **planned** and **systematic** pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.*

2. *A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with: quality control.*

# What is Software Quality Assurance?

According to D. Galin

Software quality assurance is:

*"A systematic, planned set of actions necessary to provide adequate confidence that the software development process or the maintenance process of a software system product conforms to established functional technical requirements as well as with the managerial requirements of keeping the schedule and operating within the budgetary confines."*

# Software Quality Definition

- In the most general sense, software quality can be defined as: An effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it.

- An effective software process establishes the infrastructure that supports any effort at building a high-quality software product. The management aspects of process create the checks and balances that help avoid project chaos—a key contributor to poor quality. Software engineering practices allow the developer to analyze the problem and design a solid solution—both critical to building high-quality software. Finally, umbrella activities such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.

- A useful product delivers the content, functions, and features that the end user desires, but as important, it delivers these assets in a reliable, error-free way. A useful product always satisfies those requirements that have been explicitly stated by stakeholders. In addition, it satisfies a set of implicit requirements (e.g., ease of use) that are expected of all high-quality software.

- By adding value for both the producer and user of a software product, highquality software provides benefits for the software organization and the enduser community. The software organization gains added value because
- high-quality software requires less maintenance effort, fewer bug fixes, and
- reduced customer support. This enables software engineers to spend more
- time creating new applications and less on rework. The user community
- gains added value because the application provides a useful capability in
- a way that expedites some business process. The end result is (1) greater
- software product revenue, (2) better profitability when an application
- supports a business process, and/or (3) improved availability of information
- that is crucial for the business.

**Barry William Boehm (May 16, 1935 – August 20, 2022) was an American software engineer, distinguished professor of computer science, industrial and systems**

Philip Crosby (1926-2001) was an influential author, consultant and philosopher who developed practical concepts to define and communicate quality and quality improvement practices. His influence was extensive and global.

Watts Humphrey ,Known as the "Father of Software Quality," Humphrey dedicated the majority of his career to addressing problems in software development, including schedule delays, cost increases, performance problems, and defects.

James Martin was an English information technology consultant and author, known for his work on information technology engineering.

Tom McCabe (Thomas J. McCabe), founder of McCabe & Associates in Columbia, Maryland, is internationally known for his development of software metrics and his leadership in developing methodology and automation that foster the continued improvement of quality in software development, testing and maintenance.

John Musa served as a US Navy engineering officer in the late '50s. Afterward, he joined AT&T Bell Laboratories and worked there for 35 years until his retirement in 1996 as a technical supervisor. He continued working through his own company to provide SRE training and consulting.

Bill Perry founded the Quality Assurance Institute (QAI) in 1980 and was the Executive Director of the institute for 28 years.

Daniel Galin is the **head of information systems studies at the Ruppin Academic Center and an adjunct senior teaching fellow at the Technion's Faculty of Computer Science**. His main research interests include software quality assurance and analysis and the design of information systems.

William Edwards Deming was an American engineer, statistician, professor, author, lecturer, and management consultant.

# Software Quality Challenges

- Quality cannot be directly checked in the product; **it must be planned** right from the beginning.

- The project must focus on the quality issues of the project from the beginning, ensuring that quality criteria are consistent with defined requirements.

- Quality means meeting requirements and meeting customer needs, which means a defect-free product from both the producer's and the customer's viewpoint.

# Software Quality Engineering Road Map

- *Quality Engineering at every SDLC phase:*
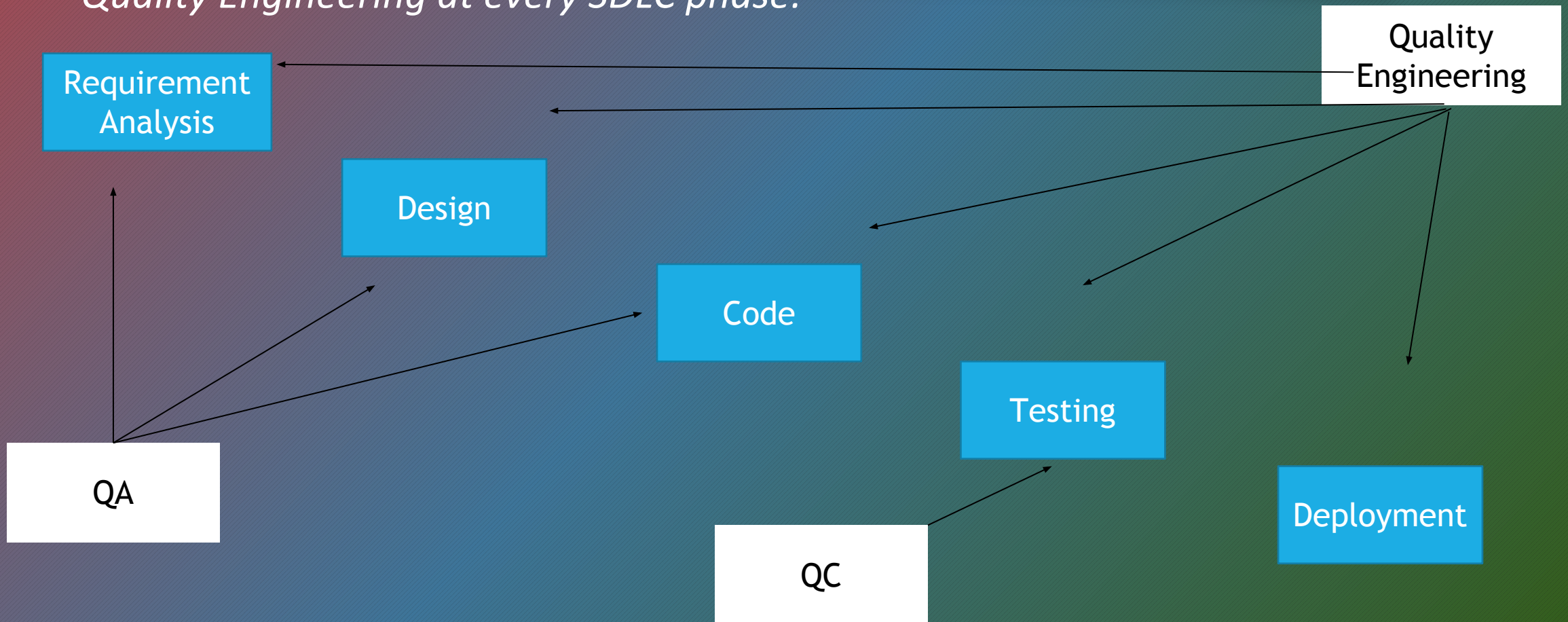
# What is an Error?

- Error is usually refer to some syntax mistake by developer.

- Refer to both Syntax and semantic error.

- Syntax error are usually fixed by developer.

- Semantic error give rise to unusual output.

# What is Defect?

- Defect is deviation from customer requirement.
- Mostly defects are found in the software after software is shipped to the customer at production site.

- Example:
- In online shopping, the option of searching a debit card for making payment is missing.

# What is Bug?

- Error found before the software is shipped into production.

- "Defects" accepted by developers are bugs.

# What is Fault?

- Fault is a result of error.

- Fault is an incorrect step or process due to which unanticipated result arises.

# What is Failure?

- Failure is a result of faults.

- Failure is inability of the program to behave as expected within given performance requirement.

## VERIFICATION

- Verification addresses the concern: "Are you building it right?"
- Ensures that the software system meets all the functionality.
- Verification takes place first and includes the checking for documentation, code, etc.
- Done by developers.
- It has static activities, as it includes collecting reviews, walkthroughs, and inspections to verify a software.
- It is an objective process and no subjective decision should be needed to verify a software.

## VALIDATION

- Validation addresses the concern: "Are you building the right thing?"
- Ensures that the functionalities meet the intended behavior.
- Validation occurs after verification and mainly involves the checking of the overall product.
- Done by testers.
- It has dynamic activities, as it includes executing the software against the requirements.
- It is a subjective process and involves subjective decisions on how well a software works.

# *Quality Engineering Economics*

- High concerns and challenges in the software quality engineering, one must realize the following facts in order to cope with the quality task:

  - Everything in the process of software development ends up in the user's satisfaction.

  - Satisfaction of the user is dependent on the overall behavior of the system.

  - Features and quality of the software product are defined/determined through requirements.

# *Function Quality Cost (FQC)*

In most development projects, functionality and quality ( QA precisely) are natural enemies.

Projects with open budgets are very rare, usually the budget is fixed and here the functionality and quality compete with each other in order to get a bigger share from budget.

- The Function-Quality-Cost comes out to be

$$\text{Cost} = AF + BQ$$

- Where A & B = Level of investment
- F = Features/Functions
- Q = Quality

# Function Quality Cost (FQC)

- It is very much clear that increasing feature in a closed-budget project will certainly decrease the budget share for quality of the product.
- The following example will elaborate the concept more clearly.

(Dev + QA)

| Quality vs. Pre-defined Budget Scenario | |
|---|---|
| Total Budget | PKR 100,000 |
| Total Features | 4 |
| Cost per Feature | 100,000/4 = PKR 25,000 |

2. اَللّٰهُمَّ صَلِّ عَلٰى مُحَمَّدٍ وَّعَلٰى آلِ مُحَمَّدٍ کَمَا صَلَّيْتَ عَلٰى اِبْرَاهِيْمَ وَعَلٰى آلِ اِبْرَاهِيْمَ اِنَّکَ حَمِيْدٌ مَّجِيْدٌ اَللّٰهُمَّ بَارِکْ عَلٰى مُحَمَّدٍ وَّعَلٰى آلِ مُحَمَّدٍ کَمَا بَارَکْتَ عَلٰى اِبْرَاهِيْمَ وَعَلٰى آلِ اِبْرَاهِيْمَ اِنَّکَ حَمِيْدٌ مَّجِيْدٌ.

اے اللہ! اپنی رحمت نازل فرما محمد پر اور آل محمد پر جیسا کہ تو نے اپنی رحمت نازل فرمائی ابراہیم پر اور آل ابراہیم پر۔ بیشک تو تعریف والا اور بزرگی والا ہے۔ اے اللہ! برکت نازل فرما محمد پر اور آل محمد پر جیسا کہ تو نے برکت نازل فرمائی ابراہیم پر اور آل ابراہیم پر۔ بیشک تو تعریف والا اور بزرگی والا ہے۔

(صحیح البخاری: 3370)