

# Software Quality Assurance & Software Testing

IS5113 & SE5102

---

LECTURE ONE

BY:

P.M.A.K.WIJERATNE

# Course Name: Software Quality Assurance

---

Course Code: IS5113

Credit Value: 02

Core/Elective: Elective

Hourly Breakdown: Theory – 15,

Practical – 30,

Independent Learning - 55

# Course Name: Software Testing

---

Course Code: SE5102

Credit Value: 02

Core/Elective: Core

Hourly Breakdown: Theory – 15,

Practical – 30,

Independent Learning - 55

# What is Software Testing

---

- Software testing is the process of evaluating the functionality of a software system or application.
- The main objective of software testing is to compare the actual function results with the expected functionality to ensure that the software application is free of defects and ready for use.
- It involves identifying product bugs and requirements that have not been met.

# Importance of Software Testing: Why is it Needed?

---

- Identify the poor design and architectural level decisions at early stages.
- Software testers who participate in every testing phase gain a deeper understanding of the project, identify issues earlier, collaborate with developers, and produce a higher-quality final product.
- This improves the efficiency of the development process, reduces the risk of negative feedback, and ultimately leads to greater user satisfaction.

# What is Quality?

---

- Quality is defined as the product or service that should be "fit for use and purpose."
- Quality is all about meeting the needs and expectations of customers concerning the functionality, design, reliability, durability, and price of the product.

# What is Assurance?

---

- Assurance is a positive declaration on a product or service.
- It is all about the product which should work well.
- It provides a guarantee that would work without any problem according to expectations and requirements.

# What is Quality Assurance (QA)?

---

- Quality Assurance is also known as QA Testing.
- QA is defined as an activity to ensure that an organization is providing the best product or service to the customers.
- Quality Assurance is a systematic way of creating an environment to ensure that the software product being developed meets the quality requirements.
- This process is controlled and determined at the managerial level.
- It is a preventive process whose aim is to establish the correct methodology and standard to provide a quality environment to the product being developed.



# A well-defined QA strategy should consider the following areas;

---

- Governance, financial reporting and stakeholder engagement and risk management
- Project team skill assessment and training requirements
- Communication and collaboration
- Methodologies
- Document control
- Requirements gathering process and definition of non-functional requirements
- Application / Service Architecture

# A well-defined QA strategy should consider the following areas;

---

- Testing strategy
- Test Environments
- Continuous integration and continuous delivery (CI/CD pipelines)
- Version control and branching strategy
- Design standards and reviews
- Coding standards, code quality checks and reviews

# What are the Quality Attribute of software?

---

The following six characteristics can define the quality of the software:

## 1. Functionality

- Quality of software is defined as how effectively the software interacts with other components of the system.
- The software must provide appropriate functions as per requirement, and these functions must be implemented correctly.

## 2. Reliability

- It is defined as the capability of the software to perform under specific conditions for a specified duration.

# What is the Quality Attribute of a software?

---

## 3. Usability

- Usability of software is defined as its ease of use.
- Quality of the software is also identified as how easily a user can understand the functions of the software and how much efforts are required to follow the features.

## 4. Efficiency

- The efficiency of the software is dependent on the architecture and coding practice followed during development.

# What is the Quality Attribute of a software?

---

## 5. Maintainability

- Maintainability is also one of the significant factors to define the quality of the software.
- It refers to identify the fault and fix in the software. It should be stable when the changes are made.

## 6. Portability

- Portability of the software, defined as how easily a system adapts to changes in the specifications.
- Quality of the software is also determined by the portability of the system how easy it is to install the software and how easy it is to replace a component of the order in a given environment.

# What is Debugging?

---

- Debugging is the action that the development team or a developer implements after receiving the test report related to the bugs in the software from the testing team.
- In the debugging process, the developer needs to identify the reason behind the particular bug or defect, which is carried out by analyzing the coding rigorously.
- The developer changes the code and then rechecks whether the defect has been deleted whenever the bug or error is found.
- Once the debugging is successfully finished, the application is again sent back to the test engineers, who remain in the process of testing.
- The debugging process allows us an earlier finding of an error and makes software development stress-free and easy.

# Testing Vs. Debugging

---

S.NO	Testing	Debugging
1.	It is the implementation of the software with the intent of identifying the defects	The process of fixing and resolving the defects is known as debugging.
2.	Testing can be performed either manually or with the help of some automation tools.	The debugging process cannot be automated.
3.	A group of test engineers executes testing, and sometimes it can be performed by the developers.	Debugging is done by the developer or the programmer.
4.	The test engineers perform manual and automated test cases on the application, and if they detect any bug or error, they can report back to the development team for fixing.	The developers will find, evaluates, and removes the software errors.
5.	Programming knowledge is not required to perform the testing process.	Without having an understanding of the programming language, we cannot proceed with the debugging process.

# Testing Vs. Debugging

6.	Once the coding phase is done, we proceed with the testing process.	After the implementation of the test case, we can start the Debugging process.
7.	Software Testing includes two or more activities such as validation and verification of the software.	Debugging tries to match indication with cause, hence leading to the error correction.
8.	It is built on different testing levels such as Unit Testing, Integration Testing, System Testing, etc.	It is built on different kinds of bugs because there is no such level of debugging is possible.
9.	Software testing is the presentation of defects.	It is a logical procedure.
10.	Software testing is the vital phase of SDLC (Software Development Life Cycle).	It is not a part of SDLC because it occurs as a subset of testing.



# What are Software Quality Assurance components?

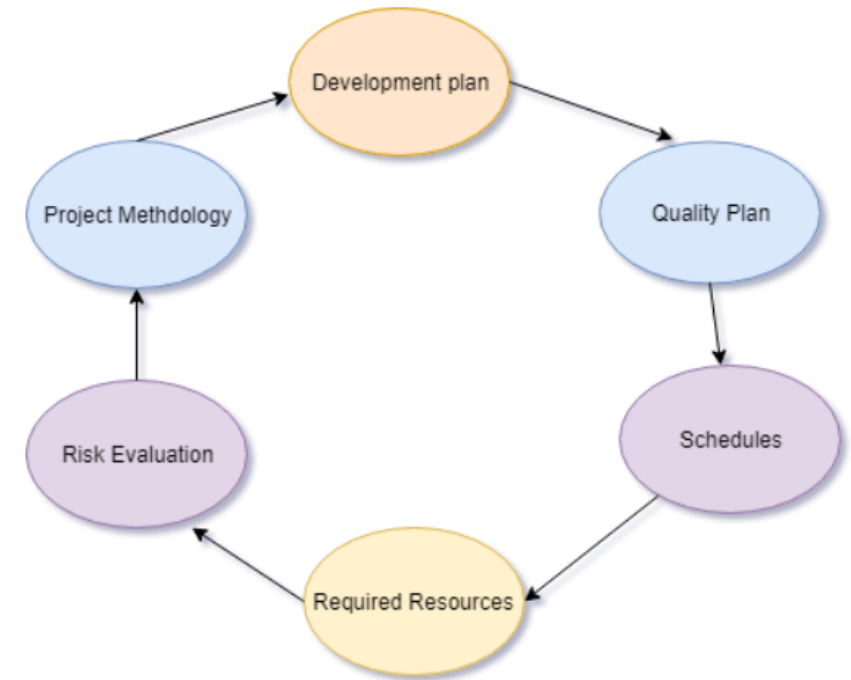
---

Software Quality Assurance has six classes of components.

## 1. Pre-project Plan

Pre-project Plan ensures that the resources required for project, schedule, and budget should be clearly defined.

Plan for development and ensuring quality has been determined.



*Diagrammatic Representation of Pre-project components*

# What are Software Quality Assurance components?

---

## 2. Project lifecycle component

A project lifecycle usually comprised of two stages:

### 1. Development Stage

- In the Development Stage Component, Software Quality Assurance helps to identify design and programming errors.
- Its Components are divided into the following sub-classes: Reviews, Expert Opinions, and Software Testing.

# What are Software Quality Assurance components?

---

## 2. Project lifecycle component

### 2. Operation Maintenance Stage

In Operation Maintenance Stage, the Software Quality Assurance components include the Development lifecycle component along with specialized components whose aim is to improve the maintenance tasks.



*Diagrammatic Representation of Project Lifecycle Component*

# What are Software Quality Assurance components?

---

## 3. Infrastructure error prevention and improvement components

The aim of this component is to the prevention of software faults and minimizes the rate of errors.

These components are as:

- Procedure and work instructions
- Templates and Checklists
- Staff Training, Retaining and Certification
- Preventive and Corrective Actions
- Configuration Management
- Documentation Control

# What are Software Quality Assurance components?

---

## 4. Software Quality Management Components

- This class of component consists of controlling development and maintenance activities.
- These components establish the managerial control of software development projects.
- The management component aims to prevent the project from going over budget and behind schedule.

The management components include:

- Project Progress Control
- Software Quality Metrics
- Software Quality Costs

# What are Software Quality Assurance components?

---

## 5. Standardization, Certification, and SQA assessment components

- Aim of these components is to implement international managerial and professional standards within the organization.
- These components help to improve the coordination among the Organizational Quality Systems and establish standards for the project process.

The component includes:

- Quality management standards
- Project process standard

# What are Software Quality Assurance components?

---

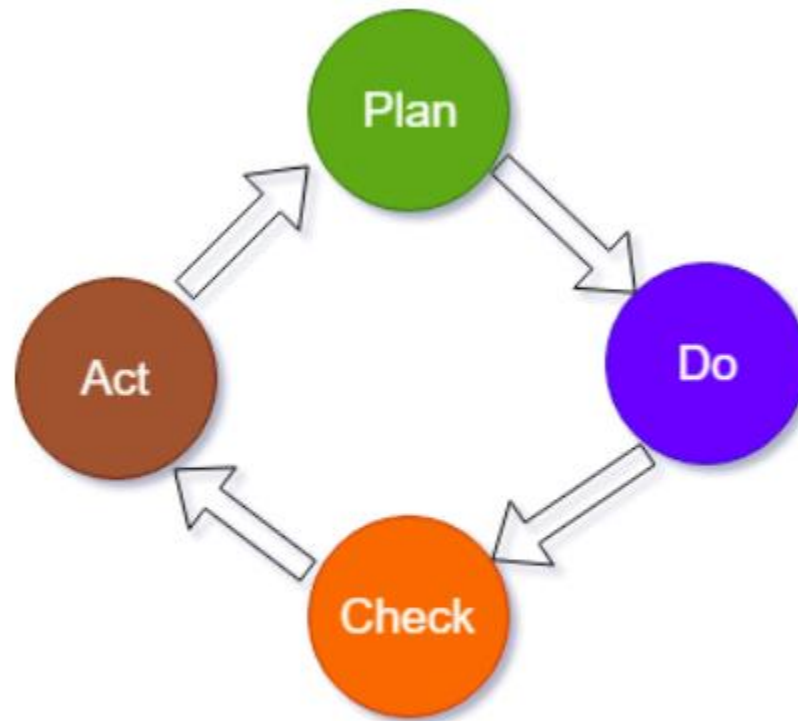
## 6. Organizing for Software Quality Assurance (the human elements)

- The main aim of this class of components is to initiate and support the implementation of Software Quality Assurance components, identify any deviations from the predefined Software Quality Assurance procedures, methods, and recommended improvements.
- The Software Quality Assurance organizational team includes test managers, testers, SQA unit SQA committee, and SQA forum members.

# How to do Quality Assurance?

---

The whole process of quality assurance has to define the cycle called the PDCA cycle.



*Graphical Representation of Process of Quality Assurance*



# How to do Quality Assurance?

---

- **Plan:** The organization should **plan and establish the process related objectives** and **determine the process** that is required to deliver a high-quality end product.
- **Do:** **Development and testing of processes** and also **change in the methods**.
- **Check:** **Monitoring of processes, modify the methods**, and **check whether it meets the predetermined objectives**.
- **Act:** **Implement actions** that are necessary to **achieve improvements** in the process.

# What is Quality Control?

---

- Quality Control is a software engineering process used to ensure the quality of a product.
- It does not deal with the processes used to create a product.
- Instead, it examines the quality of the end product and the outcome.
- The main aim of **Quality Control is to check whether the product meets the specification and requirement of the customer.**

# What is the difference between Quality Control and Quality Assurance?

---

Quality Assurance	Quality Control
Quality Assurance prevents defects.	Quality Control provides identification of defects.
Quality Assurance is process oriented.	Quality control is product oriented.
Quality Assurance is proactive in the process and protective.	Quality Control is a reactive.
Quality Assurance is a managerial tool.	Quality Control is a corrective tool.
Each developer is responsible for Quality Assurance.	The testing team is responsible for Quality Control.
Verification is an example of QA.	Validation is an example of QC.
The focus of QA is to prevent defects in the developing software by paying attention to processes.	The focus of QC is to identify deficiencies in the developed software by paying attention to testing processes.

# Challenges in Software Quality Assurance

---

- **Time constraints:** Often, the SQA team has limited time to test the software application thoroughly, which can lead to incomplete testing and the possibility of bugs going unnoticed.
- **Lack of resources:** The SQA team may not have access to the necessary tools, equipment, or personnel to perform comprehensive testing.
- **Communication issues:** Miscommunication between the SQA team and other stakeholders, such as developers or project managers, can lead to misunderstandings and errors.

# Challenges in Software Quality Assurance

---

- **Complexity of the application:** As software applications become more complex, it becomes increasingly difficult to test all possible scenarios and edge cases.
- **Changing requirements:** Changes in project requirements can make it difficult for the SQA team to keep up with testing and ensure that all new features are thoroughly tested.
- **Automation challenges:** While automation can help streamline testing, it also comes with its own set of challenges, such as maintaining test scripts and ensuring that they are up-to-date.

---

Thank You!

