Java Institute for Advanced Technology

UNIT NAME: SOFTWARE ENGINEERING II

UNIT ID: HF2W 04

ASSIGNMENT ID: HF2W 04/AS/01

NAME: M.R.P.N.THARUKSHA RAJAPAKSHA

STUDENT ID: 2019/2020/CO/SE/I2/029

SCN NO: 207977608

NIC: 200019401866

BRANCH: JAVA INSTITUTE, COLOMBO

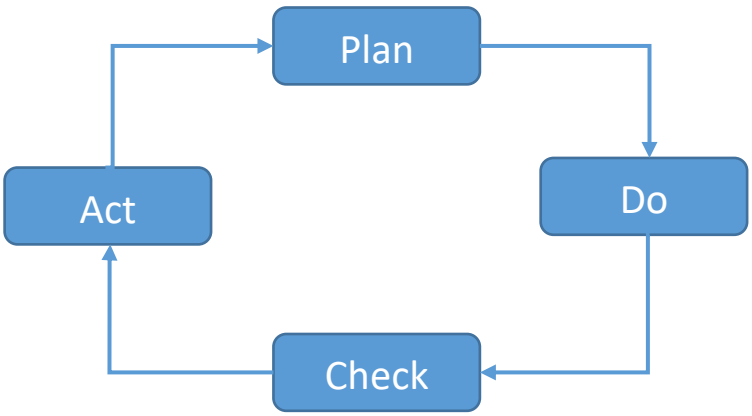
1. What is Software testing?

Software testing is a method of testing whether the actual software product meets the desired requirements and ensuring that the software product is flawless. It involves activating software/system components manually or using automated tools to evaluate one or more of the properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements as opposed to the actual requirements.

2. What are the things found when software testing?

- Architectural flaws
- Bugs
- Errors
- Failures
- Faults
- Invalid or incorrect functionality
- Poor design decisions
- Scalability issues
- Security vulnerabilities

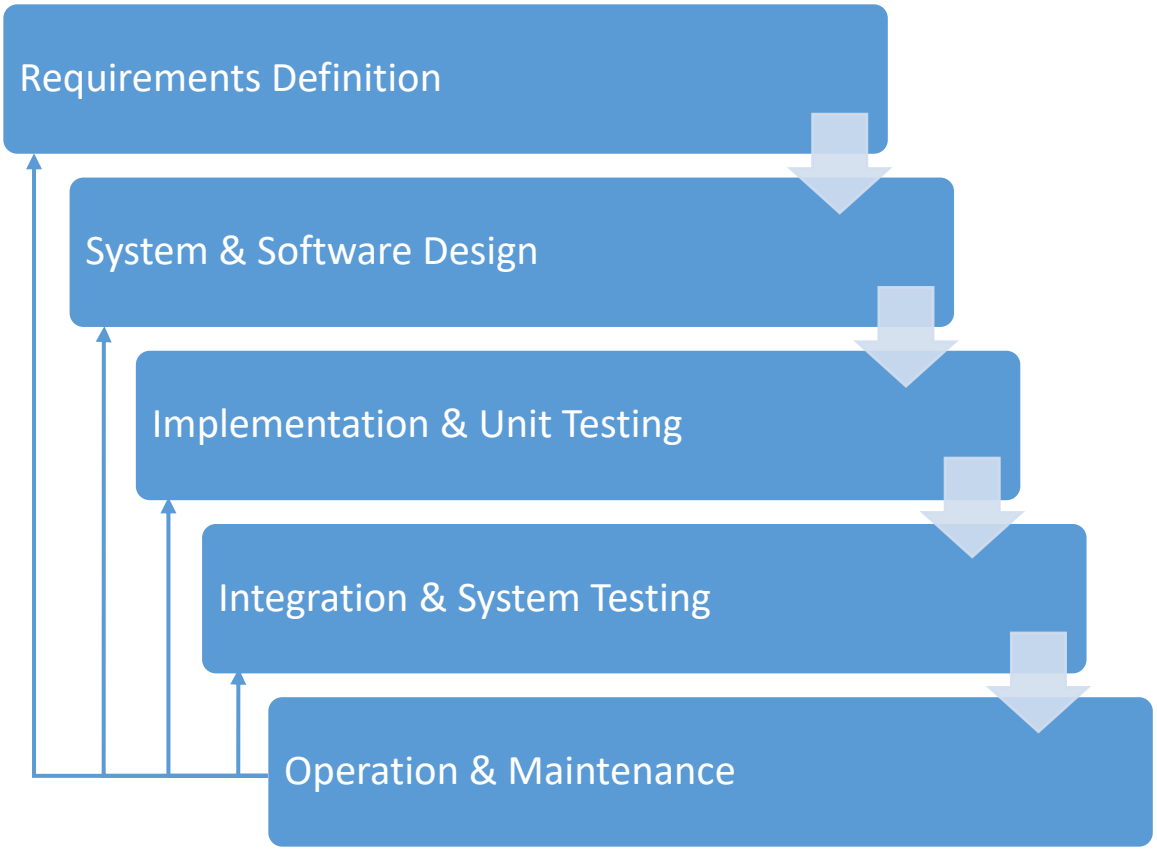
3. Explain the PDCA cycle and where testing fits in?



Software testing is an important part of the software development process. There are four important steps in general software development, also known as the PDCA (Plan, Do, Test, Act) cycle.

1. Plan: Define the goal and the plan to achieve that goal.
2. Do: At this stage, we will implement accordingly according to the planning strategy decided at the planning stage.
3. Check: Check/Test to make sure we are following the plan and getting the desired results.
4. Act: During the inspection cycle, if there are any issues, we will take appropriate action accordingly and revise your plan accordingly.

4. What is a software process? Explain with a diagram



A software process is a set of related activities that lead to software development. These activities may involve software development or modification of an existing system from scratch.

- Requirements Definition
  All the requirements of the system to be developed are captured at this stage and the requirements are documented in a specification document.

- System & Software Design
  The requirements specifications of the first stage are studied at this stage and the system design is prepared. This system design helps to specify the hardware and system requirements and helps to define the overall system architecture.

- Implementation & Unit Testing
  With system design applications, the system is first developed by small programs called units, which are then integrated into the next stage. Each unit is developed and tested for its functionality, called unit testing.

- Integration & System Testing
  All units developed during the activation phase are integrated into a system after each unit is tested. After integration, the whole system is checked for any errors and failures.

- Operation & Maintenance
  There are several issues that arise in the consumer environment. To resolve those issues, patches will be issued. Better versions will be released to improve production. Maintenance is done to deliver these changes in the customer environment.

5. Why testing is important?

Few people can argue against the need for quality control in software development. Delayed distribution or software errors can damage the brand's reputation - leading to frustrated and lost customers. In extreme cases, a fault or error can cause interconnected systems to degenerate or cause serious malfunctions.

6. What is the software bug and explain it

A software bug is a problem that causes a program to crash or produce invalid output. The problem is caused by inadequacy or misconceptions. An error can be an error, a mistake, an error, or an error, which can lead to failure or deviation from the expected results. Many errors are caused by human errors in the source code or its design. A program is said to be erroneous if it contains a large number of errors that affect the performance of the program and cause erroneous results.

7. What are the factors Testing can involve?

- Compatibility
- Correctness
- Integration
- Negative
- Performance
- Reliability
- Security
- Usability
- User Interface

8. Explain the software testing team Key Players and their Roles

- QA Engineer
  Checks software to identify bugs and errors. Checks whether a product meets requirements. A detective who knows where to hide bugs, even where no one expects. Focuses, reductions, and sometimes tests the system using special software.

- Test Analyst
  Project Document Teacher. The first to decide what and how to test. Knows exactly what needs to be produced. QA systematizes information to make the life of the engineer easier.

- Test Architect
  Finds final solutions that meet client requests and align with team resources. Have a complete overview of the software system. Knows every little element and how it interacts with other features.

- Test Manager
  Is fully responsible for the success (or failure) of the project. Forms test strategies, define the scope of work for other members, controls test execution.

- QA Team Lead
  Supervisor. You can participate in any of the above processes, but usually, the situation is checked and the team manages. Conducts interviews. Recruitment of new members and mentors. Deals with management tasks far more than technical tasks.

9. Who does the testing?

- Business Analysts
- End User
- Engineers In Test,
- Performance Testers
- Project Lead/Manager
- QA Analysts
- QA Engineer
- QA Testers
- Quality Assurance Engineers
- Software Developers
- Software Testers
- SQA Engineers
- Test Analysts
- Test Engineers
- Test Managers
- Usability Testers

10. Write short description for following topics?

- Error
  An error means an error that allows the program execution to complete and yield results that may be wrong but not easily identifiable as incorrect. An error means an error in coding or logic that causes a program not to substantially function as described in the applicable Specifications.

- Bug
  A bug is an error, flaw, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. Bugs may have subtle effects or cause the program to crash or freeze the computer.

- Fault
  A fault is a condition that causes a system to fail in performing its required function. A fault is the basic reason for software malfunction and is synonymous with the commonly used term bug.

- Failure
  Inability to set up a program due to incorrect logic. Things like crashes, bombs, and the like.