24/08/2020

Mohd. Aftab Alam
02013302717

EXPERIMENT 1

Aim :

Introduction to Software Testing and Quality Assurance.

Explain the following terms in brief :

1. Software

Software is a collection of data, instructions or program that tell the computer how to work using the hardware attached with it. Without software, the hardware is just a useless piece of machine. There are two types of software : system software and application software.

Example : Visual Studio Code – it is a free source code editor made by Microsoft for Windows, Linux and macOS. It is the most popular developer environment tool.

2. Program

Program is a set of instructions given by the user to a computer to perform a specific task. It can be written in any programming language like C/C++, Python, Java, Swift, JavaScript, etc. A program in execution state is called process.

Example : Python program to print “Manchester United” is shown below –

```
print('Manchester United')
```

3. Software Engineering

Software Engineering is a discipline whose aim is the production of quality software, software that is delivered on time, within budget and that satisfies its requirements. The result of software engineering is an efficient and reliable software product.

4. Software Testing

Software Testing is a process to check whether the actual results match the expected results and to ensure that the software system is defect free. It is important because software bugs could be expensive or even dangerous.

5. Software Testing Process / V Model of Testing

This process involves 5 steps which are given below –

- Planning and Control

Planning – it involves producing a document that describes an overall approach and test objectives. It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and designing the test environment.

Control – this is the activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan. It involves taking actions necessary to meet the mission and objectives of the project.

- Analysis and Design – major tasks : to review the test basis, to identify test conditions, to design the tests, to design the test environment set-up and identify the required infrastructure and tools

- Implementation and Execution

Implementation – major tasks : to develop and prioritize test cases by using techniques and create test data for those tests, to create test suites from the test cases for efficient test execution, to re-execute the tests that previously failed in order to confirm a fix, to log the outcome of the test execution and to compare actual results with expected results.

Execution – it involves actually running the specified test on a computer system either manually or by using an automated test tool. It is a fundamental test process in which actual work is done.

- Evaluating Exit Criteria and Reporting – it is a process defining when to stop testing. It depends on coverage of code, functionality or risk. Basically it also depends on business risk, cost and time and vary from project to project. Major tasks : to assess if more tests are needed or if the exit criteria specified should be changed and to write a test summary report for stakeholders.

- Test Closure Activities – these activities are done when software is ready to be delivered. Major tasks : to check which planned deliverables are actually delivered and to ensure that all incident reports have been resolved, to finalize and archive test ware such as scripts, test environments, etc. for later reuse, to handover the test ware to the maintenance organization and to evaluate how the testing went and learn lessons for future releases and projects.

6. Why should we do Testing ?

Software Testing helps to identify errors, gaps and missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss.

7. What should we Test ?

We should test a program for all possible valid and invalid inputs or we can say all possible execution paths. For this we require effective planning, strategies and sufficient resources. We should also test the program for very large numbers, very small numbers, numbers that are close to each other, negative numbers, some extreme cases, characters, and some strange cases.

8. Who should do the Testing ?

For a small software, the developer should be responsible for testing and there are various plus points of testing done by the developer. Apart from this, there are professionals who work as full time software testers. There are various companies who hire these professionals to test their products and services.

9. Quality Assurance

Quality Assurance (QA) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering products or services to customers. It comprises administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled.

10. Quality Control

Quality Control (QC) is a process by which entities review the quality of all factors involved in production. Inspection is a major component of quality control, where physical product is examined visually (or the end results of a service are analyzed).

11. Verification and Validation

Verification – the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. It is a static practice of verifying documents, design, code and program. It will help to determine whether the software is of high quality, but it will not ensure that the system is useful. It is concerned with whether the system is well-engineered and error-free.

Validation – the process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. It is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

12. Error, Defect, Bug and Failure

Error – it is a human mistake. It appears not only due to the logical mistake in the code made by the developer. Anyone in the team can make mistakes during the different phases of software development.

Defect – it is a variance between expected and actual results. An error that the tester finds is known as defect. A defect in a software product reflects its inability or inefficiency to comply with the specified requirements and criteria and, subsequently, prevent the software application from performing the desired and expected work. It is also known as ***Fault.***

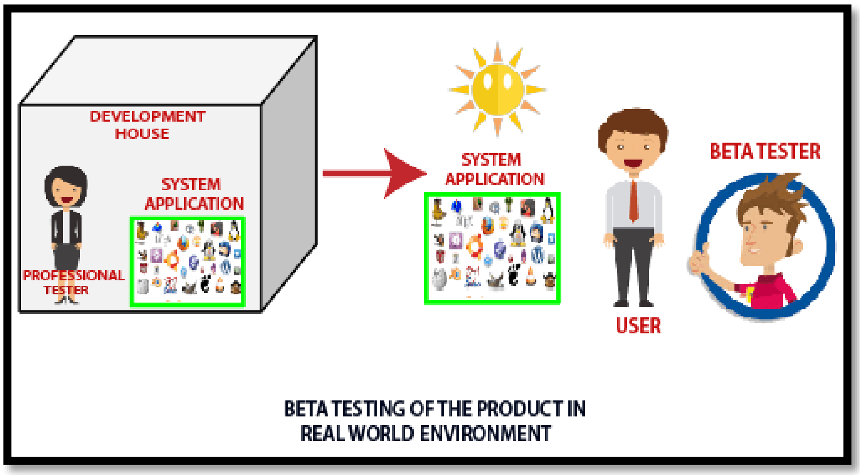
Bug – it occurs because of some coding error and leads a program to malfunction. It may also lead to a functional issue in the product. These are fatal errors that could block a functionality, results in a crash, or cause performance bottlenecks.

Failure – it is a consequence of a defect. It is the observable incorrect behavior of the system. It occurs when the software fails to perform in the real environment. Not all defects result in failures; some remain inactive in the code, and we may never notice them.

13. Alpha, Beta and Acceptance Testing

Alpha Testing – it is performed to identify all possible issues and bugs before releasing the final product to the end users. It is carried out by the testers who are internal employees of the organization. The main goal is to identify the tasks that a typical user might perform and test them.

Beta Testing – it is performed by "real users" of the software application in "real environment". It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of beta testing. This testing helps to test products in customer's environment.



Acceptance Testing – it is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of black box testing where the number of required users involved test the acceptance level of the system.

14. Deliverables and Milestones

Deliverables – it is a distinct, tangible or intangible outcome of your project that is produced during the project's course. These are developed by the project team members in alignment with the overall objectives of the project.

Milestones – these are checkpoints in the project that help you chart progress throughout the course of the project. These control points help identify that a number of tasks or key deliverables have been completed allowing you to move on to the next phase of your project.

15. Static and Dynamic Testing

Static Testing – testing which checks the application without executing the code. It is a verification process. It is performed in the white box testing phase, where the programmer checks every line of the code before handling over to the test engineer.

Dynamic Testing – testing which is done when the code is executed at the run time environment. It is a validation process where functional and non-functional testing is performed. It is performed to check whether the application or software is working fine during and after the installation.

16. White Box and Black Box Testing

White Box Testing – here the internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to software testers.

White Box Testing Techniques :

- Data Flow Testing
- Control Flow Testing
- Branch Testing
- Statement Testing
- Decision Testing

Black Box Testing – here the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. It mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as ***Behavioral Testing.***

Black Box Testing Techniques :

- Decision Table Technique
- Boundary Value Technique
- State Transition Technique
- All-Pair Testing Technique
- Cause-Effect Technique
- Equivalence Partitioning Technique
- Error Guessing Technique
- Use Case Technique

17. Why 100% Testing is not possible ?

100% Testing is not possible because of the following reasons :

- The domain of possible inputs of a program is too large to be completely used in testing a system. There are both valid inputs and invalid inputs.
- The design issues may be too complex to completely test. The design may have included implicit design decisions and assumptions.
- It may not be possible to create all possible execution environments of the system.

18. Limitations of Testing

Limitations of Testing are given below :

- Testing can be used to show the presence of errors, but never to show their absence. It can only identify the known issues or errors. It gives no idea about defects still uncovered. Testing cannot guarantee that the system under test is error free.
- Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.
- Software testing does not help in finding root causes which resulted in injection of defects in the first place. Locating root causes of failures can help us in preventing injection of such faults in future.

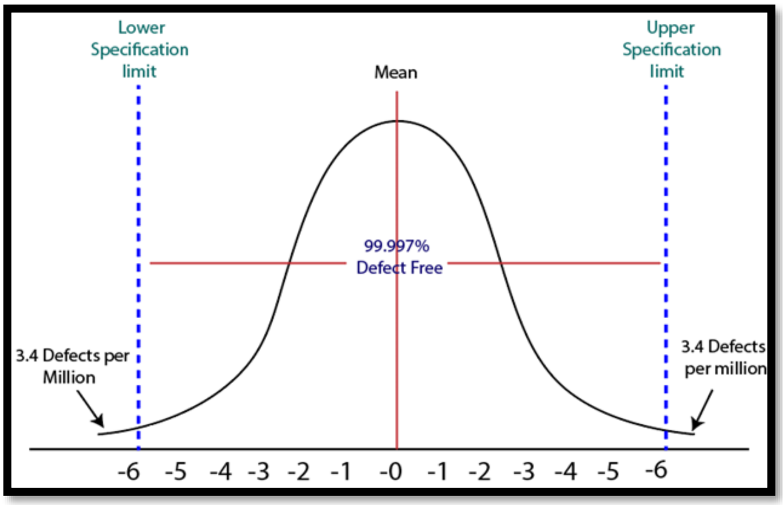
19. Quality and Reliability

Quality – it is defined in terms of its fitness of purpose; that is, a quality product does precisely what the users want it to do. Characteristics of quality are – portability, usability, reusability, correctness and maintainability.

Reliability – it means operational reliability. It is described as the ability of a system or component to perform its required functions under static conditions for a specific period. It is also defined as the probability that a software system fulfills its assigned task in a given environment for a predefined number of input cases, assuming that the hardware and the input are free of error.

20. Six Sigma Concepts

Six Sigma is the process of improving the quality of the output by identifying and eliminating the cause of defects and reduce variability in manufacturing and business processes. A six sigma method is one in which ***99.997%*** of all the opportunities to produce some features of a component are statistically expected to be free of defects.

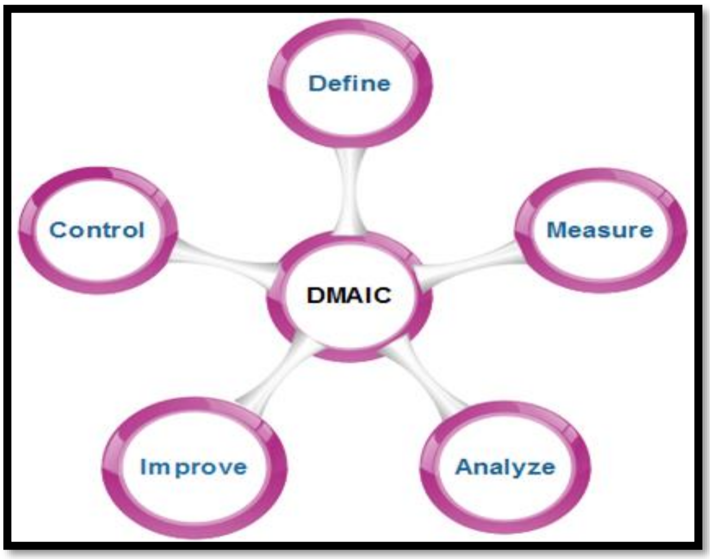


Characteristics of Six Sigma :

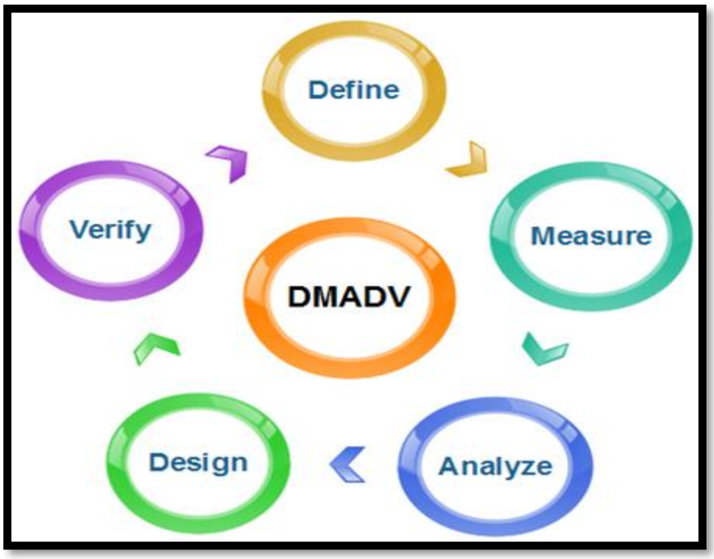
- Statistical Quality Control
- Methodical Approach
- Fact and Data-Based Approach
- Project and Objective-Based Focus
- Customer Focus
- Teamwork Approach to Quality Management

Six Sigma Methodologies :

DMAIC - It specifies a data-driven quality strategy for improving processes. This methodology is used to enhance an existing business process.



DMADV - It specifies a data-driven quality strategy for designing products and processes. This method is used to create new product designs or process designs in such a way that it results in a more predictable, mature, and detect free performance.


Define
Verify
Measure
DMADV
Design
Analyze