QA Software Testing

Manual Testing Content Slides

What is Software Testing?

Software testing is a process used to identify the correctness, completeness, and quality of developed computer software.

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided it's design and development.

Is Testing Important?

Quality software is reasonably bug or defect free, delivered on time and within budget, meets requirements and/or expectations, and is maintainable.
Key aspects of quality for the customer include:

Good design – looks and style
Good functionality – it does the job well
Reliable – acceptable level of breakdowns or failure
Consistency
Durable – lasts as long as it should
Good after sales service
Value for money

IT Company Structure

IT Company Structure have following below main departments which are explained with there work and roles in coming slides.

- Functional Manager
- Project Manager
- Business Analyst
- QA Manager
- Developer
- Client

IT Company Structure Continues...

Functional manager (Role)

- Assign project
- Discuss how well a person is doing that work and if a person wants to continue doing it (providing opportunities for growth)
- Gather information from other PMs to write the evaluation
- Work with the employee to set and coach on career goals

IT Company Structure Continues...

Project Manager (Role)

- Develop a project plan
- Manage deliverables according to the plan
- Recruit project staff
- Lead and manage the project team
- Determine the methodology used on the project
- Establish a project schedule and determine each phase
- Assign tasks to project team members
- Provide regular updates to upper management

IT Company Structure Continues...

Business Analyst (Role)

- Assist in defining the project
- Gather requirements from business units or users
- Document technical and business requirements
- Verify that project deliverables meet the requirements
- Test solutions to validate objectives

IT Company Structure Continues...

QA Manager (Role)

- Convert the requirements and design documents into a set of testing cases and scripts, which can be used to verify that the system meets the client needs.
- This collection of test cases and scripts are collectively referred to as a test plan.

Developer (Role)

- The Developer is responsible for the actual building of the solution.

IT Company Structure Continues...

Test Analyst/Tester/Test Engineer (Role)

- The Tester ensures that the solution meets the business requirements and that it is free of errors and defects.
- Identifying the Target Test Items to be evaluated by the test effort
- Defining the appropriate tests required and any associated Test Data
- Gathering and managing the Test Data
- Evaluating the outcome of each test cycle

Client (Role)

- They are the people for whom the project is being undertaken

Tester Qualities

1. Curiosity

- Always curious to know what will be next output.
- Always curious to check the flow.
- As a tester, you should never assume anything.

2. **Logical Thinking**

- Testing needs smart and outboxes thinking.
- Logical thinking helps in finding out the most unexpected bugs.
- As a tester, you need to focus on nonfunctional requirements also, that is not clearly said.

Tester Qualities Continues...

3. **Attentive**
   - A tester should keep a sharp eye on each and every detail of the product.
   - Every small detail is helpful in solving and finding big bugs.

4. **Imagination**
   - Imagine all the possibilities that may occur.
   - Consider all the possible scenario.
   - Imagine beyond the documentation and test script.

Tester Qualities Continues...

5. **Communication** skills

- As a tester, you need to interact with your team to gather information related to the product.
- A person with good communication skills can interact with and understand the product requirements.

6. Test for quality over quantity

- Focus on finding sensible bugs.
- Finding quality bugs is more important than Finding a large number of useless bugs.

Tester Qualities Continues...

7. **Learn to prioritize**

- Focus on Finding higher priority bugs that may severely affect the system.
- Prioritizing gives a clear view of which bugs are to be fixed first.

8. Learn from your own mistakes – and from others too

- Learning from mistakes is the best way to improve your skills.
- Learning from others mistake save your time.

Software Development Life Cycle (SDLC)

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality softwares. The SDLC aims to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software development process.
- The software development life cycle (SDLC) is a framework defining tasks
- performed at each step in the software development process.

Software Development Life Cycle (SDLC)

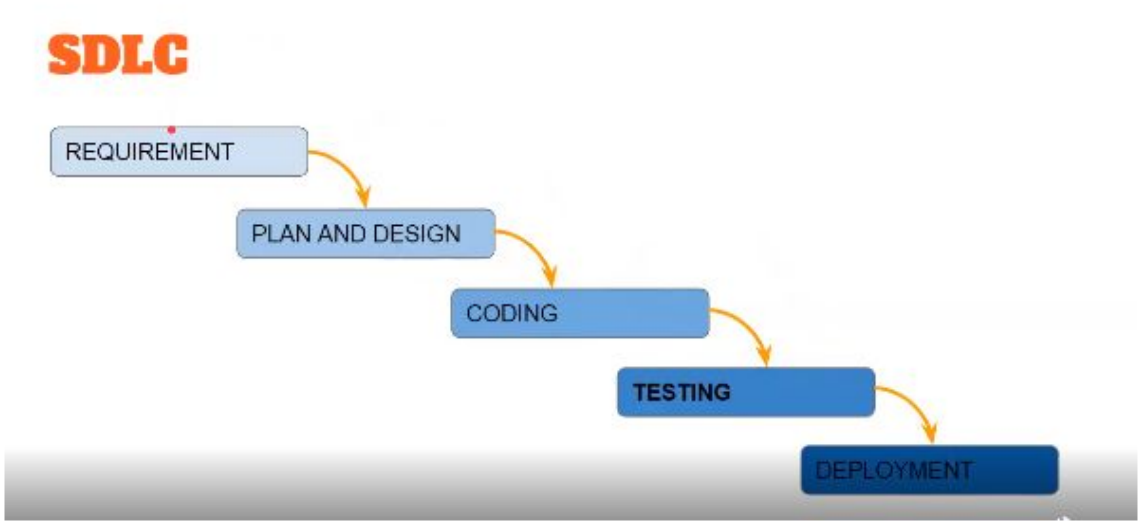
Continues

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

Software Development Life Cycle (SDLC)

Continues



Software Development Life Cycle (SDLC) Continues

A typical Software Development life cycle consists of the following stages:

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Software Development Life Cycle (SDLC) Continues

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through .SRS. . Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Software Development Life Cycle (SDLC) Continues

Stage 3: Designing the product architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity , budget and time constraints , the best design approach is selected for the product.

Software Development Life Cycle (SDLC)

Continues

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java, and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Software Development Life Cycle (SDLC)

Continues

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Software Development Life Cycle (SDLC)

Continues

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations. business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing). Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

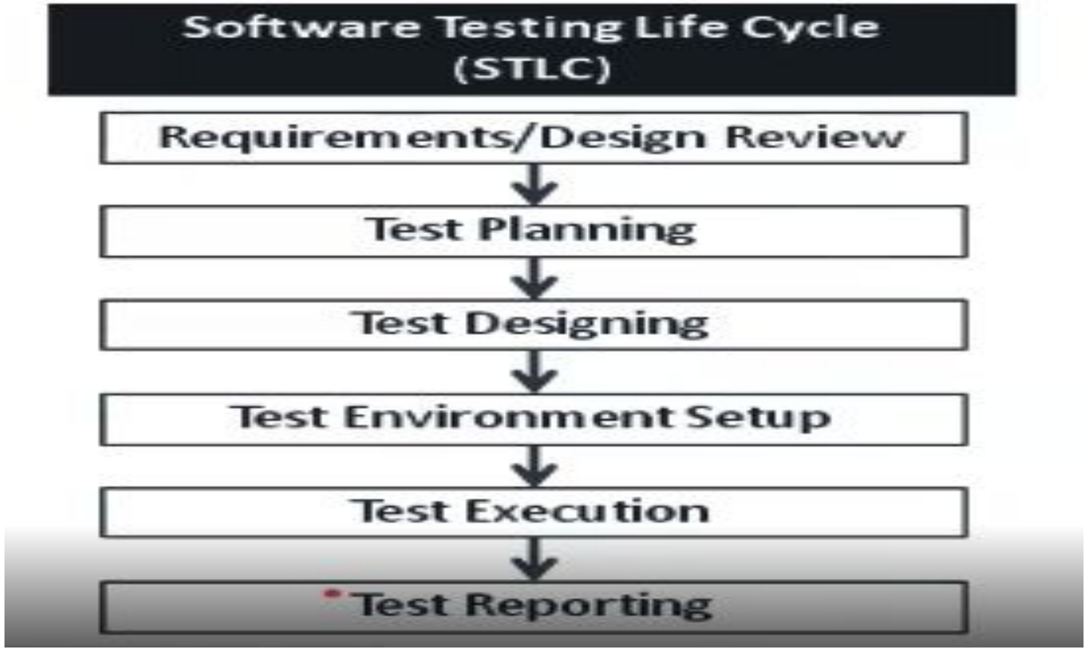
Software Testing Life Cycle (STLC)

Software Testing is not a just a single activity. It consists of series of activities carried out methodologically to help certify your software product. These activities (stages) constitute the Software Testing Life Cycle (STLC).
Software Testing Life Cycle (STLC) defines the steps/stages/phases in testing of software.

However, there is no fixed standard of STLC in the world and it basically varies as per the following:

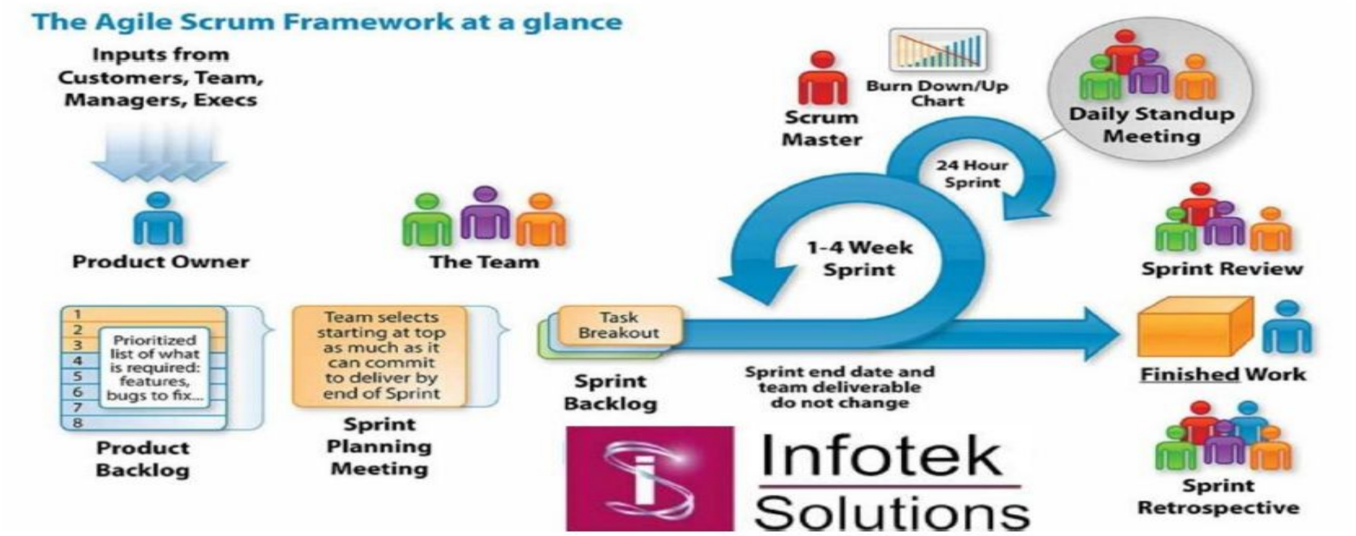
- Software Development Life Cycle
- Whims of the Management

Software Testing Life Cycle (STLC) Diagram



Agile Scrum Methodology Concept

The Agile process is really quite simple, so let's concentrate on Scrum methodology here which is the most popular. As the graphic shows the process starts with the product owner creating a prioritized product backlog.


The Agile Scrum Framework at a glance
Inputs from Customers, Team, Managers, Execs
Product Owner
The Team
Scrum Master
Burn Down/Up Chart
Daily Standup Meeting
24 Hour Sprint
1-4 Week Sprint
Sprint Review
Prioritized list of what is required: features, bugs to fix...
1
2
3
4
5
6
7
8
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective
Infotek Solutions

Agile Scrum Methodology Concept Continues...

The prioritized backlog is actually one of the most understated feature of this process too many companies start a development project by trying to outline every single thing that has to go into this application or software but they fail to realize that a large percentage of the features may never be used may never work correctly or may just waste valuable time and dollars. A backlog is comprised of stories, in the story the product owner describes who will be conducting the operation what they will be doing and why. This gives great insight and clarity to the team. By creation a backlog the product owners essentially making a lot of the dreams and wishes in story form, then the product owner organizes the backlog in prior order with the most important items at the top and the least important at the bottom. Now this list and priority can change after each sprint or iteration. The goal here is to focus on what brings value to the business and not on what we would like to have.

Agile Scrum Methodology Concept Continues...

The process is all driven by the prioritized product backlog we just described. The product backlog is the input necessary for the sprint planning meeting.

Sprint is a term used in the scrum framework to describe the basic unit of development & testing, about time if we talk here so in most cases it will be from one week to one month in length for sprint. A project or release is usually made up of several or many sprints. Getting back to the sprint planning meeting the purpose of the meeting is for the team to commit to the amount of work they can complete during the sprint the work is roughly sized using one of several agile estimation techniques allowing the team to determine the proper amount of work for the sprint.

Agile Scrum Methodology Concept Continues...

An important point is here is the team isn’t just doing this as a guessing game, it is much more disciplined than that. A team has a velocity or amount of work per sprint which over time they know to be accurate.
Good teams will never commit to more than their normal velocity they also recognize the importance of the priorities in the product backlog and will always work on the highest priority items which will fit in the sprint. This gives the customer their highest priority items and therefore their highest value as early as possible.

Agile Scrum Methodology Concept Continues...

Once the sprint planning meeting is completed the team starts the actual work each day during the sprint the team gets together for a short meeting lasting 15 minutes or less called the daily stand-up meeting or daily scrum meeting. During this meeting team members tell each other that what they have completed since the last meeting, what they intend to complete before the next meeting and whether that have any impediments which may cause them to become blocked.

The team uses this time to determine how best to share information to help each other in order to complete their sprint commitment.

This short meeting exposes risks and knowledge to be shared which in turn help the team be more effective.

Agile Scrum Methodology Concept Continues...

Up to this point we haven’t mentioned to the role on the graphic called Scrum Master. The person filling this special role is responsible for helping ensure the success of the project. They remove impediments for the team help the team make responsible decisions and basically support the team in any way possible.

This role touches all areas of the organization while representing the team and is usually vital for team success. During the daily meeting this person note any potential impediments and immediately begin working to make sure the impediment won’t be a problem. At the end of the sprint usually on the last day two other meetings are held, the first is the sprint review or sprint demo.

Agile Scrum Methodology Concept Continues...

The sprint review or sprint demo is important for many reasons. First of all it shows the product owner and client what the team accomplished, they already knew what we were going to work on. By now they can see the results firsthand they can also provide feedback based on what they see. By sprint review or sprint demo we can mitigate the risk and deliver exactly what the client wants.
The final piece of the process is a sprint retrospective. Like the sprint review the sprint retrospective has one main purpose in the case of the sprint review the main purpose is to improve the product while the sprint retrospective main purpose is to improve the team in the process this core principle of continuous improvement is one of the things which makes scrum completely different from other development frameworks.

Agile Scrum Methodology Concept Continues...

The sprint retrospective empowers teams to be successful by allowing them to work on improvement each sprint. These improvements are not driven by managers but by the team taking pride in their own work. However once the sprint retrospective is completed the team cycles back to the beginning of a new sprint where they begin with sprint planning.

A very important part of the process is the finished work. Finished work means it is a piece of the product which has been completed and is functional in other words every sprint the team will finish work which has some value and is truly completed. It allows working software to be demonstrated and if enough working software has been completed it may allow for an early product release. As we can see with the help of scrum framework by little planning and flexibility greatly increases the odds of a successful project.

Test Plan

Test plan is document detailing approach to test a system (machine or software). It is basis for testing any software or machine. Test plan describes important information about test project, resource that will be used during testing efforts. It identifies features to be tested, test design technique, entry and exit criteria and testing environment, dependencies of tester and risks, as well as Schedule.

Purpose of the Test Plan Document

- Specify the approach that Testing will use to test the product, and the deliverables (extract from the Test Approach).
- Break the product down into distinct areas and identify features of the product that are to be tested.
- Specify the procedures to be used for testing sign-off and product release.
- Indicate the tools used to test the product.
- List the resource and scheduling plans.
- Indicate the contact persons responsible for various areas of the project.
- Identify risks and contingency plans that may impact the testing of the product.
- Specify bug management procedures for the project.
- Specify criteria for acceptance of development drops to testing.

Types of Test Plan

- Master Test Plan : A single high level Test plan for a testing system that unifies all other type of test plans
- Testing level : Test plans for each level of Testings:
    - Unit Testing
    - Integration Test Plan
    - System Test Plan
    - Acceptance Test Plan

- Testing Type : Testing Plans for major type of testing like performance Test plan, security Test plan.

Contents of a test plan

- **Introduction**

This is the first section of the test plan and should contain information about what are the things/information we are presenting in this (test plan) document. In short we can say a brief description/purpose about the test execution plan of that specific Project.

- **Approach**

This section of the test plan will describe what type of test approach we are adopting for execution of the project. Is it manual testing or automation testing? with suitable reasons.

Contents of a test plan Continues...

- **Schedules**

Here mention the details of the project schedules , such as start date and end date .In between if we have beta or alpha release mention schedule(s) of that also, apart from these, give build wise start date and end dates. If test cases are divided into module wise you can give the schedules as per that also.

(Note) It is advised to include buffer time in the schedules, so that unexpected events can be handled in time.

Contents of a test plan Continues...

- **Resources**

Give the Hardware, software(s) requirements and the man power needed for the Project execution.

(Note1) Retain spare systems (systems apart from the test environment) and at least one backup person, so that unexpected events can be handled.

(Note 2) Do consider the budget allocation for that project, while requesting for the hardware & software(s) and manpower.

Contents of a test plan Continues...

- **Environment**

Give a detailed list of environment(s) on which the test bed setup has to be created. Mention each OS & software(s) along with its version(s) and patch(s).

(Note) Any specific test cases those are specific to OS or software(s). Give the total number of such specific test cases in a tabular format (if applicable)

Contents of a test plan Continues...

Test Methodologies
This section of the test plan mentions the details of the type of test methodology that is used to validate the system/application /product. Give a brief description about the Methodology adopted Along with the suitable reasons why we have chosen method with respect to the project/product. Following are most frequently used test methodologies.

- Black box testing methodology
- Gray box testing methodology
- White Box testing methodology

Contents of a test plan Continues...

- **Requirements/features to be tested**

List out the features/requirements of the system/product/application that are to be tested. Give a detailed list module wise. (If available).

(Note) Represent the data in a tabular format with use case id, SRS id and test cases id. This makes an easy read ability.

Contents of a test plan Continues...

- **Requirements/features not to be tested**

List out the features/requirements of the System/product/application that are not to be tested because of various reasons like:

a) Non availability of environment or different software(s).
b) Any specific feature(s) or functionality that is not yet implemented in the Current Build etc..

Contents of a test plan Continues...

- **Assumptions**

List out the possible assumptions for the project like

  - It is assumed that test environment is properly configured and specific to that project only which will not be used for any other purpose.
  - Development test environment will be separate from QA's test environment.
  - Development team releases testable software to testing team on schedule etc.

Contents of a test plan Continues...

Risks
Here mention the possible risks for the project execution. They might be

For example

- Testing the functionality of the product which is integrated with a third Party software.
- Team is not properly trained on the technology on which the application is built and to be tested.
- Execution of many test cases in a shorter period of time due to deadlines etc.

Contents of a test plan Continues...

Test Exit Criteria

Give the information about when to stop testing of a product. List out the conditions at which testing can be stopped. The reasons can be like

- All test cases are executed and bugs were fixed to an extent
- Test cases completed with certain percentage passed
- Project has reached the end date
- Product is stable after certain regression test cycles performed on it.
- Bug rate falls below a certain level etc.

Contents of a test plan Continues...

Release Criteria
When testing is completed and QA deems that the following Items have been satisfactorily met, a recommendation will be made by QA to release the product. A Test and Evaluation report will be submitted reflecting quality Assurance assessment of the product at this time.

- All Severity 1 defects are resolved.
- All must be fixed bugs or high priority bugs are resolved
- All Severity 2 problems must have acceptable work-arounds, which are documented in Release Notes or Read me file, which are available to customers.

Contents of a test plan Continues...

- The main features of the product are fully implemented and function according to requirements.
- All test cases are executed at least once
- Product is stable on specified OS
- User documentation is accurate and fully descriptive of the product. Etc.
  (This is the end of the contents of the test plan)

Test Plan Sample & Test Plan Template

Test Plan Sample:
https://training.qaonlinetraining.com/manual-testing/test-plan/testplan-sample/

Test Plan Template:
https://training.qaonlinetraining.com/manual-testing/test-plan/test-plan-template/

Test Cases

Test case is a series of steps or a set of execution conditions or predicted results with specifying inputs to test correct behavior / functionality, features of application. Aim of test case is to divide software functionality into small unit of function that is testable with specified input and generate result that is measurable.

Test case is feature or function that should be executed with specified input range, given preconditions and generates result against expected result.

How to write a good test case

How write a good test:

- Test case should be accurate and tests what is intend to be test.
- No unnecessary steps should be in test case
- Test case should be reusable. So you can use test case to test different applications
- Test case should be traceable to test requirements
- Test case should be independent. There is no dependency in your test case and you should be able to execute test case in any order without dependency on other test case.
- Test case should be easy, clear and small. So other testers can understand your test case by reading once.
- Ensure that all positive and negative scenario cover in test case.

Traceability Matrix

The traceability matrix links a business requirement to its corresponding functional requirement right up to the corresponding test cases. If a Test Case fails, traceability helps determine the corresponding functionality easily. It also helps ensure that all requirements are tested.

Additionally, A traceability matrix is not only for a manual testing specific tool, tester can also use for automation projects as well.
It is also not a tool that can be used just by the testing purpose. The development team can use the same to map Business Requirement document / Functional Specification Document requirements to functions /conditions to make sure all the requirements are developed.

Why we use Traceability Matrix

We know that the focus of testing is to test maximum feature, functionalities of application. Traceability matrix establishes a link between user requirements and test cases. In simple word, Traceability matrix is simple document which maps user requirements and with test case ids.

Lets understand more with example:
https://training.qaonlinetraining.com/manual-testing/traceability-matrix/what-is-traceability-matrix/

Traceability Matrix Template

Kindly click on the below link for Traceability Matrix Template:

https://training.qaonlinetraining.com/manual-testing/traceability-matrix/traceability-matrix-template/

Defect

Defect is an error or a bug in the application which is created. Example defect maybe broken link or error in function. A developer or programmer while designing application can make a mistakes or errors. If there are mistakes and errors in application, it means there are flaws in application. These flaws are called Defects.

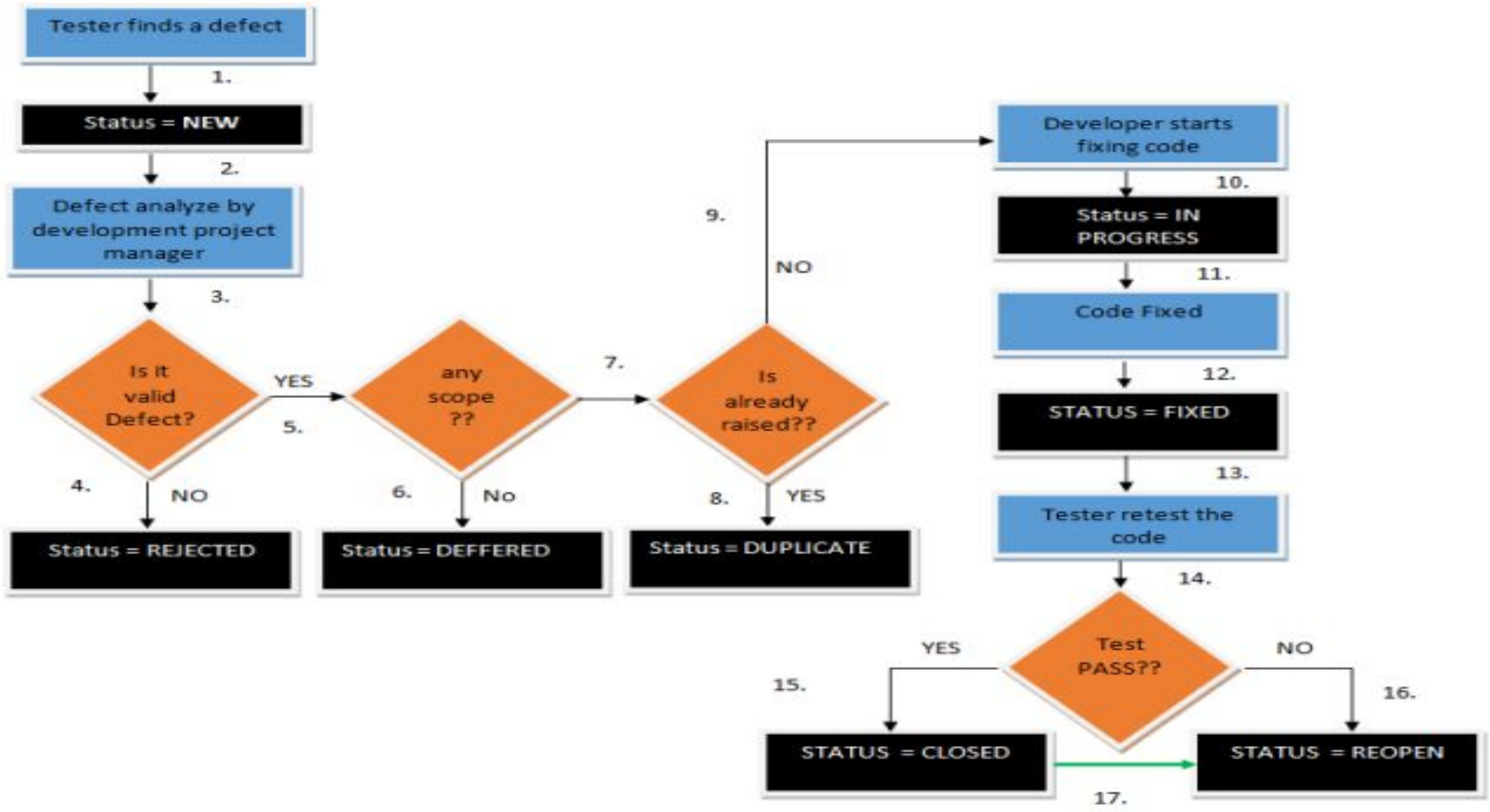
Different organizations have different names to describe this variation, commonly defects are also known as bug, problem, incidents or issues.

Defect Continues...

When Software tester examine the expected result and actual result during the testing application, if expected result deviate from actual result then results into a defect. These defects occur because of error in logic or in coding which results into failure results.

Note: When a tester finds a bug or defect it’s required to convey the same to the developers. Thus they report bugs with the detail steps and are called as Bug Reports, issue report, problem report, etc.

Defect Life Cycle



Defect Life Cycle Continues...

Check out above graphical representation of Defect Life cycle.

You can see points between states. Let’s discuss these points.

1. Suppose tester finds that actual result and expect result do not match this means a defect occurs. The defect is assigned a status NEW.
2. The defect is assigned to development project manager who will analyze the defect.
3. He will check whether it is a valid defect.

Defect Life Cycle Continues...

4. Consider that in the flight reservation application of QTP tool, the only valid password is mercury. But you test the application for some random password, which causes logon failure and report it as defect. Such defects due to corrupted test data, miss configurations in the test environment, invalid expected results etc are assigned a status REJECTED.
5. If not, next the defect is checked whether it is in scope.
6. Suppose you find a problem with the email functionality of QA Training portal (www.training.qaonlinetraining.com). But it is not part of the current release. Such defects are DEFERRED (postponed)

Defect Life Cycle Continues...

7. Next, manager checks whether a similar defect was raised earlier (it means this defect occurred earlier).
8. If yes defect is assigned a status DUPLICATE
9. If no, the defect is assigned to developer who starts fixing the code.
10. During this stage defect is assigned and developer is working on that then IN-PROGRESS status assigned to Defect.
11. Developer is trying to fix the defect.
12. Once code is fixed. Defect is assigned a status FIXED.

Defect Life Cycle Continues...

13. In next process, tester will re-test the code.
14. Tester will check test is passed or not
15. In case the test case passes then CLOSED status assigned to Defect.
16. But if the test cases fails again the defect is REOPEN and again assigned to the developer.
17. Consider a situation where during the 1st release of QA training portal (www.training.qaonlinetraining.com) defect which occurred during the testing of email functionality that one fixed by developer, and assigned a status closed. During the 2nd upgrade release the same defect (email one) again re-surfaced. In such cases, a closed defect will be REOPEN.

That's all to Defect Life Cycle

Defect Report

Defect Report is a document that identifies and describes a defect detected by a tester. The purpose of a defect report is to state the problem as clearly as possible so that developers can replicate the defect easily and fix it.

Sample of Defect Report:
https://training.qaonlinetraining.com/manual-testing/defect-management/defect-report/