**MANUAL TESTING**

**By**

**Ramanjaneyulu…**

**CONTENTS**

1. **Introduction**

* **What is Testing?**
* **Why do we need Testing?**
* **Basics of Testing**
* **Ways of Testing**
* **Testing Methods (or) Testing Techniques**

1. **Company Hierarchy**

* **Bidding the Project**
* **Kickoff Meeting**

1. **SDLC (Software Development Life Cycle) Process**

* **Requirement Phase**
* **HLD (High Level Designing)**
* **LLD (Low Level Designing)**
* **Testing Phase**
* **Release and Maintenance Phase**

1. **Types of Testing**
2. **Levels of Testing**
3. **SDLC Models (Software Development Life Cycle Models)**

* **Waterfall Model**
* **V-Model**
* **Iterative Incremental Model**
* **Agile Model**

1. **STLC (Software Testing Life Cycle)**

* **Test Initiation**
* **Test Planning**
* **Test Designing (Test cases writing)**
* **Test Execution**
* **Defects Tracking**
* **Reporting**

1. **Defect Life Cycle**
2. **Traceability Matrix**
3. **Test Metrics**
4. **Roles and Responsibilities of a Tester**

**1.Introduction**

* **What is Testing?**
* Testing is the process of Verification and Validation, where we can find the bugs in the developed software and make sure that the bugs should be fixed, provide a quality and hence client or customer satisfaction.
* Testing is the process of executing Test cases with the intent of finding bugs in software.
* Exercising or evaluating the software by manual or automated means to verify that it satisfies the specified requirements, or to identify differences between expected and actual results
* Testing is the measurement of Software Quality.
* **Why do we need Testing? Or Why do we do Testing?**
* To find bugs
* To provide defect free product to the Client or end-users.
* Quality Assurance to the Client
* Client or Customer satisfaction
* To reduce costly errors
* Cut down costs

**Note:** Defect and Bug both are same, As the terminology may varies from one company to another company and one client to another client.

* **Basics of Testing**

**Project:**

A Project is something (Software, System, Application, etc.,) that is developed specific to one client or customer requirements, And that can be used by that customer himself.

**Example:** Banking Application, Insurance appn, etc.,

**Product:**

A Product is something (Software, System, Application, etc.,) that is developed based on company specifications by gathering the requirements from multiple customers or clients who may use that product, And that can be used by multiple customers.

**Example:** Google, Facebook, Gmail, Yahoo, etc.,

**Quality Concepts:**

**Quality:**

* Justification of all requirements of a customer in a project/Product.
* Meeting the customer requirements.
* Meet customer expectations (cost to use, speed in process or performance, security)
* Time to release.

**Verification:**

* Analyzing the quality by reviewing the documents. (The documents may be test cases document, Metrics documents, etc.,)
* Are we building the product right?
* Checks whether the product is built as per the specified requirement and design specification.

**Validation:**

* Analyzing the quality by executing the test cases against the client requirements.
* Are we building the right product?
* It determines whether the software is fit for use and satisfy the business need.

**Quality Control:**

*Quality control involves the series of inspections, reviews and tests that are used throughout the software process to ensure each work product meets the requirements.*

*A key concepts in Quality control is that all work products have defined, measurable specifications to which we can compare the output of each process.*

**Quality Assurance:**

*Quality A***s***surance consist of a set of auditing (inspecting/look over) and reporting functions that assess the effectiveness and completeness of Quality control activities*

*The main goal of QA is to provide data to management about product quality.*

**Quality Analyst:**

**Quality Center:**

It’s a Test Management tool from HP, Where we manage the testing process like storing the User requirements, Manual test cases, Automation Scripts, Results, Defects, etc., The primary intension of Quality Center is to Track the Defects.

**Defect or Bug:**

* Deviation from the client requirements
* Deviation between actual and expected results.
* **Ways of Testing**

There are Two ways of Testing

1. Manual Testing
2. Automation Testing
3. **Manual Testing:**

Testing the application manually with human efforts with the intent of finding errors is nothing but Manual Testing.

1. **Automation Testing:**

Testing the application by using some tools is nothing but automation testing

* **Testing Methods (or) Testing Techniques**

There are 3 kinds of testing techniques.

1. Black box Testing
2. White box Testing
3. Grey box Testing
4. **Black box Testing:**

If user performs testing on only the functional part of an application without having the knowledge of the structural part, then that method of testing is known as Black box Testing.

* Usually the Test Engineers will perform the BBT.

1. **White box Testing:**

If user performs testing on only the structural part of an application, then that method of testing is known as White box Testing.

* Usually Developers will perform the WBT.

1. **Grey box Testing:**

If one performs testing on both the functional part as well as structural part of an application, then that method of testing is known as Grey box Testing.

* The Test Engineers who has the structural knowledge will usually perform it.

**Company Hierarchy**

CEO (Chief Executive Officer)🡪Delivery Head🡪Business Analyst🡪Engagement Manager🡪both Development Manger and QA Manager🡪Test Lead🡪Test Engineer

* **Bidding the Project:**

It’s defined as request or proposal, estimation and Signoff.

* **Kickoff Meeting:**

It’s an initial meeting arranged in an organization soon after the project is signed off in order to discuss the overview of the project and also to select a Project Manager for that project.

Usually Project Managers, Technical Managers, Quality Managers and high level management will be involved in this meeting. Sometimes development leads and test leads may also involve in this meeting.

* Apart from this meeting any other startup meeting is also known as Kickoff meeting.

**SDLC (Software Development Life Cycle) Process**

It contains 5 phases.

1. Requirement Phase
2. High Level Designing (HLD)
3. Low Level Designing (LLD)
4. Testing Phase
5. Delivery and Maintenance Phase
6. **Requirement Phase:**

*Task:* Interacting with the customer and gathering the requirements.

*Roles:* ***B.A* →** Business Analysis

***E.M* →** Engagement Manager

*Process:* First of all B.A. will take an appointment from the customer, collect the template from the company, go to the customer on appointed date and collects the requirements using the template. He will get the raw or high level requirements from the client and those high level requirements will be converted to low level requirements and BA will prepare the FRS (Functional Requirement Specifications) document.

FRS document terminology may varies from one client to another client or one project to another project as below mentioned terms.

SRS (System Requirement Specifications)

BRS (Business Requirement Specifications)

BDD (Business Design Document)

CRS (Customer Requirement Specifications)

BD (Business Document)

URS (User Requirement Specifications)

Engagement manager is responsible for the following.

1. He deals with the excess cost of the project.
2. He is responsible for the prototype demonstration
3. **High Level Designing (HLD)**

*Task: 1)* Feasibility study

*2)* Tentative (Temporary) Planning

*3)* Technology Selection

*4)* Requirement Analysis

*Roles:* ***S.A* →** System Analyst

***P.M* →** Project Manager

***T.M* →** Team Manager (or) technical Manager

1. Feasibility study: is a detailed study on the requirements in order to check whether all the requirements are possible or not.
2. Tentative Planning: The resource as well as the time is temporarily planned here in this section.
3. Technology Selection: The technologies that are required to accomplish the project successfully are analyzed and selected here in this section.
4. Resource and Requirement Planning: All the requirements like human resources are software or hardware. i.e., require to accomplish the project successfully will be analyze and listed out here in this section and finally document is ready.
5. **Low Level Designing (LLD)**

Task: Designing code

Roles: Developer

In this phase, the CA (Chief Architect) will divide the project into modules and sub modules. The developers will start developing the code based on the Pseudo code.

1. **Testing Phase**

*Task:* Testing

*Role:* Test Engineers

*Process:* (input is FRS & Build)

1. A test engineer will review the requirements document.
2. While reviewing if at all he get any doubts he will prepare a ‘Review Report’.
3. The review report is sent to the author of the requirements documents for clarification.
4. After understanding all the requirements very clearly the test engineer will take the test case template and write the test cases.

***Text box:*** once the build is released, the test engineer will execute the test cases.

1. After the execution there is a chance of identifying the defects.
2. If at all the defects identify, the test engineer will isolate them in defects profile.
3. The defect profile sends to the developing team for fixes.
4. Once the next build is release the test engineer will execute the test cases in order to ensure that the product is defect free.
5. This process continues till the product is defect free.

The proof of the test phase is Quality Product.

1. **Delivery and Maintenance Phase**

*Test:* 1) Delivery 2) Maintenance

***Delivery:*** The Test engineers (or) senior test engineer will install the software in the client’s environment with the help of the guidelines given in the development document by the development team.

***Maintenance phase:*** whenever the problem arises that problem is considered to the task and based on the problem the corresponding role is appointed and the role will define the process according to the problem and solve it.

And if at all any frequent change request in the application, then the application goes to maintenance state. That time few of the developers and test engineers will work on the changes on request of the client.

* **Types of Testing**

1. **Smoke Testing or Build Acceptance Testing or Build Verification Testing:**

* It’s a type of testing in which one will perform overall testing on the released build in order to confirm whether the build is proper or not for performing detailed test.

Usually the user or tester will check the following during test.

1. Whether the major functionalities of an application available or not.
2. Whether the user can navigate all the pages of an application or not.
3. Whether the build is testable or not.

* This is the initial level testing in which we test on the build to check whether the build is enough stable to continue for further testing. Here we will check the major functionalities and navigation points.

1. **Sanity Testing:**

It’s a type of testing in which one will perform detailed testing on the build in order to check the consistency of the objects and its actions.

1. **Retesting:**

Retesting is 2 types.

1. Testing the same functionality or already tested functionality with different kinds of data.
2. Whenever we raise the defect and defect is fixed we will retest the same defect to make sure whether the defect is really fixed or not.

* Here we execute only the steps which are mentioned in the defect.

1. **Regression Testing:**

Once the code change is happened due to defect fix or change request or new functionality request from the client, and we have to make sure that the existing functionality did not effect. i.e., or else we have to make sure that the code change did not impact on the existing functionality.

1. **Alpha Testing:**

It’s a type of User Acceptance Testing in which the testing is performed in our Organization before delivery to the client in presence of user/client.

*Advantages:* If at all any defects are identified then there is a chance of fixing them immediately.

1. **Beta Testing:**

It’s a type of User Acceptance Testing in which the testing is performed at the client’s place either by the third party testers or end user.

*Disadvantages:* If at all any defects are identified during β - Testing then there is no chance of fixing them immediately.

1. ***Static Testing:*** If at all one perform testing on an application or application related factors without doing any action on the application or whenever it is not been executed then that type of testing is known as static testing Ex: GUI Testing & Document Testing, Code Reviews.
2. ***Dynamic Testing:*** If at all one perform testing on an application by doing some action on the application or whenever it is been executed then that type of testing is known as Dynamic Testing.
3. **Installation Testing:**

It is a type of testing in which one will install the application into the environment by following the guidelines given in the deployment document by the development team. If at all the installation is successful then he will come to a conclusion that the guidelines are perfect otherwise the guidelines are not perfect.

1. **Compatibility Testing:**

One will install the application into the different environments to make sure whether the application is compatible with all the environments or not.

1. ***Monkey testing (or) Gorilla (or) Chimpanji Testing:*** It is a type of testing in which one will perform abnormal actions on the application intentionally (wantedly) in order to check the stability of the application.
2. ***Exploratory Testing:*** It is a type of testing in which the domain experts will perform testing on an application without any requirements document supports paralelly by knowing the functionality.
3. ***Usability Testing:*** It is a type of testing in which one will perform testing on the application in order to check whether it is user friendly or not.
4. ***End-to-End Testing:*** It is a type of testing in which one will perform testing on a complete transaction or end to end scenarios of an application.
5. ***Port Testing:*** It is a type of testing in which one will perform testing on the application after deploying it in to the clients environment in order to check whether it is compatible with that environment or not.
6. ***Security Testing:***  It’s a type of testing in which one will check the application is secured or not.

Test Engineer will concentrate on the following areas.

***a)*** Authentication ***b)*** Direct URL Testing ***c)*** Fire Wall Leakage Testing

1. **Authentication:** This is a type of security testing where in the test engineer will check whether the application is allowing only the authorized persons or not.
2. ***Direct URL Testing:*** It is a type of security testing where in the test engineer will try to access the page without logging in by directly giving the URL (Uniform Resource Locator) or by copying and pasting the URL in the another browser.
3. ***Fire Wall Leakage Testing:*** It is a types of testing in which one will logging to an application as one level of user and try to access the pages belonging to another level.
4. **Adhoc Testing:** Testing the application based on our knowledge.

This can be performed in two situations.

1. When we don’t have enough time to execute our all test cases.
2. When we have more free time after executing all our test cases.

**Levels of Testing**

There are five levels of testing.

***1.*** Unit level Testing ***2.*** Module level Testing ***3.*** Integration Testing

***4.*** System level testing ***5.*** User Acceptance Testing (UAT)

***1. Unite level Testing:*** If one performs on the unit then it is known as unit level testing. It is a white box testing and usually done by the developers or white box testers.

***2. Module level testing:*** if one performs testing on a module then that is known as module level testing. It is a black box testing and usually done by the test engineer.

***3. Integration testing:*** Usually the developers will integrate the modules once they are developed with the help of interfaces (linking program). It is theirs responsibility to test whether the interfaces are working properly or not. As it is done by the developers it is known as white box integration testing.

Usually developers integrate the modules in four approaches.

1. ***Top down approach:*** in this approach the parent modules developed first and then integrated to the child modules.

M1

M2

M3

M4

1. ***Bottomup approach:*** in this approach the child modules developed first and the integrated to the parent modules.

M1

M2

M3

M4

1. ***Hybrid or Sandwitch approach:*** this approach is a mixture of top down approach & bottom up approach.

M1

M2

M3

M4

1. ***Big bang approach:*** once all the modules are ready then at a time they will be integrated this approach is known as big band approach.

***Stub:*** while integrating the modules in top down approach if at all any mandatory (compulsory) module is missing then that module is replaced with a temporary program known as Stub.

M1

M2

M3

***Driver:*** while integrating the modules in bottom up approach if at all any mandatory module is missing then that module is replaced with a temporary program known as ‘Driver’

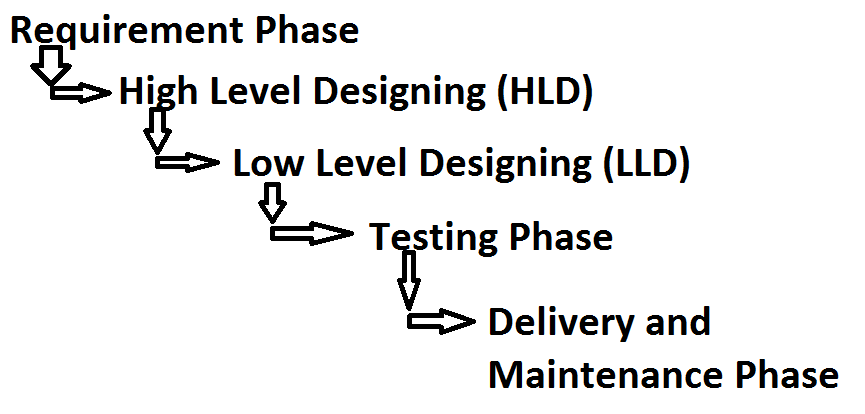
***4. System level Testing:*** If one performs testing on the application after deploying it into the environment then it is known as system testing. It is a black box testing and usually done by the test engineers. At this level of testing usually they perform different types of testing like system integration testing (black box integration testing), Load testing, stress testing, performance testing, etc.…

***5. User Acceptance Testing (UAT):*** If at all the same system testing is done in the presence of user then it is known as user acceptance testing. It is a black box testing and usually done by the test engineer.

**SDLC Models (Software Development Life Cycle Models)**

* **Waterfall Model:**

When the customer requirements are clear and complete will follow the Waterfall Model. Here the process would be done phase by phase.



**Advantages:**

* It’s a simple model.
* Project monitoring and maintenance is very easy.

**Drawbacks:**

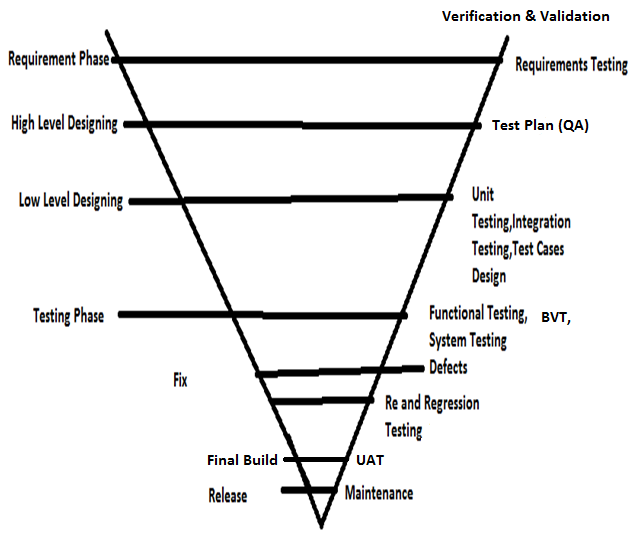
* Cannot accept the new changes or requirements in the middle of the process.
* **Prototype Model:**

When the customer requirements are unclear and confusing will follow the Prototype Model.

* **V- Model:**

V- stands for Verification and Validation.

* It is a recognized model by organizations
* This model defines the mapping in between multiple stages of development and multiple stages of testing ( or The time gap between the development and testing meets at one point that’s why it’s called as V- Model.)



**Advantages:**

* As both verification and validation are done and also the test management process is focused. The outcome will be a quality product.

**Drawbacks:**

* Time consuming model.
* Costly model.
* **Iterative Incremental Model:**

The project will be splitted into different different releases and requirements. Each requirement will be developed in one release and will give to the testing team. Once the release 1 is done development team will work on next release once it is done, they will integrate the new release with old release and will give to the testing team. Like this process will be continues till the end of the releases. Finally the development team will integrate the releases.

**Advantages:**

* Quality

**Dis Advantages:**

* It will takes more time to implement0
* Uses only for the large and long projects.
* Costly model.
* More testing efforts are required.

**Test Strategy:**

Test strategy document explains about the approach of testing that we follow throughout the project. i.e., it explains the testing methodologies that we are going to perform.

Test Strategy document contains:

1. Smoke Testing
2. Functional Testing
3. Usability Testing
4. Performance Testing
5. Security Testing
6. User Acceptance Testing (UAT)

**STLC (Software Testing Life Cycle)**

STLC contains 5 phases.

1. Test Initiation
2. Test Planning
3. Test Design
4. Test Execution and
5. Reporting
6. **Test Initiation:**

BA (Business Analyst) gets the requirements from client and he/she will review the requirements and they prepare the Test Strategy document.

1. **Test Planning:**

Once the Strategy document is ready and requirements are freezed, then TL (Test Lead/Team Lead) will prepare the test plan.

In Test Planning TL will define modules to be tested and not to be tested and approach of testing and the roles and responsibilities of each and every individual member in the team and schedule of testing. And also he express risks and mitigations.

**Test Plan Template:**

1. Modules to be tested
2. Modules not to be tested
3. Approach of testing
4. Roles and Responsibilities
5. Schedule
6. Entry Criteria
7. Exit Criteria
8. Suspension Criteria
9. Resumption Criteria
10. Risks and Mitigations

The above template is a general test plan template used in the most of the organizations. Apart from this Test Plan template every company will be having their own template.

For Example find the below template which is used in one of the MNC company.

## Contents of a Test Plan

1. Purpose
2. Scope
3. Test Approach
4. Entry Criteria
5. Resources
6. Tasks / Responsibilities
7. Exit Criteria
8. Schedules / Milestones
9. Hardware / Software Requirements
10. Risks & Mitigation Plans
11. Tools to be used
12. Deliverables
13. References
    1. Procedures
    2. Templates
    3. Standards/Guidelines
14. Annexure
15. Sign-Off

## Contents (in detail)

**Purpose**

This section should contain the purpose of preparing the test plan

**Scope**

This section should talk about the areas of the application which are to be tested by the QA team and specify those areas which are definitely out of scope (screens, database, mainframe processes etc).

**Test Approach**

This would contain details on how the testing is to be performed and whether any specific strategy is to be followed (including configuration management).

**Entry Criteria**

This section explains the various steps to be performed before the start of a test (i.e.) pre-requisites. For example: Timely environment set up, starting the web server / app server, successful implementation of the latest build etc.

**Resources**

This section should list out the people who would be involved in the project and their designation etc.

**Tasks / Responsibilities**

This section talks about the tasks to be performed and the responsibilities assigned to the various members in the project.

**Exit criteria**

Contains tasks like bringing down the system / server, restoring system to pre-test environment, database refresh etc.

**Schedules / Milestones**

This sections deals with the final delivery date and the various milestone dates to be met in the course of the project.

**Hardware / Software Requirements**

This section would contain the details of PC’s / servers required (with the configuration) to install the application or perform the testing; specific software that needs to be installed on the systems to get the application running or to connect to the database; connectivity related issues etc.

**Risks & Mitigation Plans**

This section should list out all the possible risks that can arise during the testing and the mitigation plans that the QA team plans to implement incase the risk actually turns into a reality.

**Tools to be used**

This would list out the testing tools or utilities (if any) that are to be used in the project (e.g.) WinRunner, Test Director, PCOM, WinSQL.

**Deliverables**

This section contains the various deliverables that are due to the client at various points of time (i.e.) daily, weekly, start of the project, end of the project etc. These could include Test Plans, Test Procedure, Test Matrices, Status Reports, Test Scripts etc. Templates for all these could also be attached.

**References**

Procedures

Templates (Client Specific or otherwise)

Standards / Guidelines (e.g.) QView

Project related documents (RSD, ADD, FSD etc)

**Annexure**

This could contain embedded documents or links to documents which have been / will be used in the course of testing (e.g.) templates used for reports, test cases etc. Referenced documents can also be attached here.

**Sign-Off**

This should contain the mutual agreement between the client and the QA team with both leads / managers signing off their agreement on the Test Plan.

1. **Test Design:**

Once the Test Plan is ready and testers will understand the requirements and they prepare the test cases based on the requirements.

Test cases can be designed based on the below test design techniques.

1. BVA (Boundary Value Analysis)
2. ECP (Equivalent Class Partition) and
3. Error Guessing
4. **BVA:** BVA is nothing but min-1 and max+1

**Ex:** The UserName field in the Gmail Login functionality should allow minimum 6 characters and maximum 15 characters. In such case we need to write the test cases for min-1 and max+1 characters.

1. **ECP:** Valid data and invalid data (both +ve input data and –ve input data)
2. **Error Guessing:** Analyzing the future occurrence error
3. **Test Execution:**

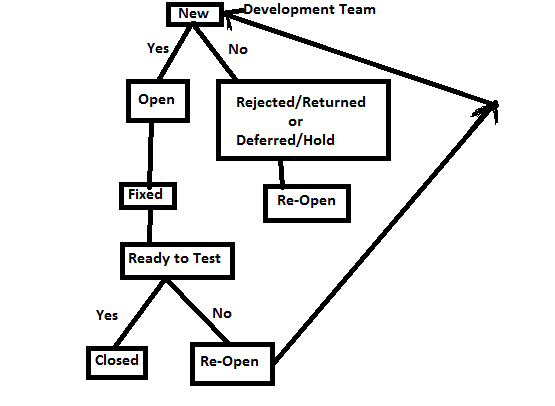
Once the test cases are freezed and initial build is released then testing team will start executing the test cases. If any defects found during execution we raise the defects.

1. **Reporting:**

We report about the testing status by using metrics.

**Defect Life Cycle (Bug Life Cycle)**

Defect Life Cycle explains about the defect age or defect tracking and updating process.



**Severity:** The impactness of the defect on the testing.

Severity will be given by the Test Engineers.

**Priority:** The impactness of the defect on the client business.

Priority will be given by the TL or Developers itself.

**Show Stopper:** Which blocks your testing.

Severity or Priority will be given as below.

Very High: If the total testing blocks

High: If one of the module or functionality blocks your testing

Medium: If one or two test cases blocks and the remaining are working.

Low: Cosmetic defects (Spelling mistakes)

**Traceability Matrix**

Traceability Matrix is nothing but mapping the requirements with the test cases to make sure whether all of our test cases are giving 100% coverage or not.

There are three kinds of Traceability Matrices.

1. RTM (Requirement Traceability Matrix)
2. DTM (Defect Traceability matrix)
3. FTM (Fulltime Traceability Matrix)

**Test Metrics**

Metrics will be used to report the progress of testing and status to judge the quality and schedule time.

Metrics are 2 types

1. Process Metrics
2. Product Metrics

* Process Metrics will be used to report the coverage, status and progress.
* Product Metrics are used to report and analyze the quality.

Different Metrics are shown below.

1. Requirement Coverage (Mapping the requirements with the test cases).
2. Efforts=(no. of test cases created/no of test cases planned)\*100
3. Execution=(no. of test cases executed/no. of test cases planned)\*100
4. % of tc’s passed=(total no. of test cases passed/total no. of test cases executed)\*100
5. % tc’s Failed=(total no. of test cases failed/total no. of test cases executed)\*100
6. % tc’s blocked=( total no. of test cases blocked/total no. of test cases executed)\*100
7. Tester Efficiency=Total no. of defects found by tester/total no. of defects found
8. Bounce Rate=Total no. of defects rejected or cncelled/Total no. of defects found
9. Defect Severity Index=(No. of high severity defects/total no. of defects)\*100