**Management**

It is a core team responsible for managing the entire s/w development process

1. Project management
2. Product management
3. Requirement management
4. Change management
5. Configuration management
6. Risk management

**1) Project management**

They consist of several operations for achieving a goal .project management process fall in five groups.

1. Initiating
2. Planning
3. Executing
4. Monitoring and Controlling
5. Closing

**2) Product management**

Managing the entire lifecycle of a product. The team members are development team, QA team and marketing team.

Product life cycle

* Requirement

Marketing or development team analyze the document form of user requirements

* Product specification

Describes the features of product

* Functional specification

Functionality of a product is described

* Coding

Requirements are converted into machine readable format

* Alpha testing

Testing the product in developers side

* Beta testing

Testing the product in customers side

* General release

Product is delivered to customer

**3) Requirement management**

Requirements management is the process of documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. It is a continuous process throughout a project.

***RTM (Requirement traceability matrix)***

* The Requirements Traceability Matrix provides a roadmap to the requirements by organizing them in to a table

3Types

* ***FORWARD TRACEABILITY***

It makes sure that each requirement is applied to the product and that each requirement is tested thoroughly.

* ***BACKWARD TRACEABILITY***

 The purpose behind this type of traceability is to verify that we are not expanding the scope of the project by adding code, design elements, test or other work that is not specified in the requirements.

**4) Change Management**

Changes are needed in any phase new requirements are added to a project for modifying existing requirements.

Changes happens in

* Plans the s/w requirements
* By bug fixes
* By fixing customer complaints

Change is required by the customer to the CCB(change control board).

They will verify the reports, If it is valid then accept records otherwise reject it.

**Scope creep**

It refers to increasing in size of project when the requirements are change

The way to prevent scope creep is to gather requirements completely before the s/m

**Version control**

Problem is mainly occurred for testers. When the version is being tested is not the same version as that with development team has completed.

**RISK**

Risk is the potential loss or unfavorable condition of an organization which involve

unauthorized modification loss of information, unavailability of components etc.

Risk is measured by using Risk Analysis

Risk analysis is performed during initial stages of test planning

Four steps of Risk analysis

1. Form the risk analysis team
2. Identify the risk
3. Estimate the magnitude
4. Select the testing priority

Risk Management

Managing and controlling the occurrence of risk in a s/w project

*Risk reduction Method*

Tools and methods are used to prevent the occurrence of risk

*Contingency planning*

Methods and techniques are used to prevent the occurrence of risk.

3 types of risk

*1) Primary testing Risk*

* Not enough training for testers
* Lack of tools
* Lack of testers(workers
* Lack of customer involvement
* Not enough budget

*2) Primary s/w risk*

* New technology
* New environment
* New development process

3*) Premature release risk*

* The requirement were implemented incorrectly
* The test plan has not been completed
* The s/w released into production contain defect the testing may not identify the defect

**BUILDING OF TEST ENVIRONMENT**

***Test bed***

It is an environment containing the s/m instruments/m tools and other supporting elements need to conduct a test.

***Use case***

It provides the format for capturing the technical requirements applicable to a s/w release. A use case is a written description of how users will perform tasks on your website.  It outlines, from a user’s point of view, a system’s behavior as it responds to a request.

**What Use Cases Include**

Who is using the website

What the user want to do

The user's goal

The steps the user takes to accomplish a particular task

How the website should respond to an action

***Difference between use case and test case***

Use case and test case both uses in the software industry interchangeably and it has its different meanings. One collects the requirements and another analyzes it.

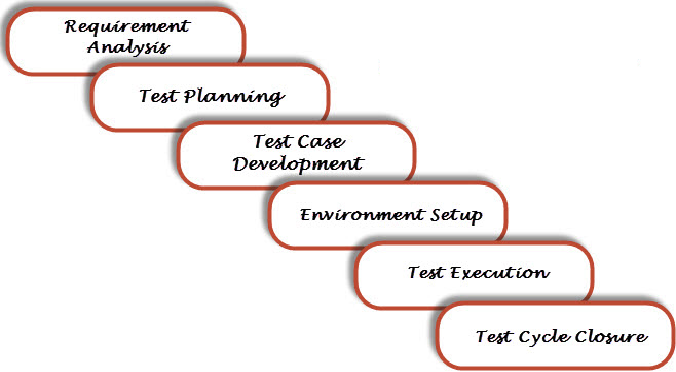
|  |  |
| --- | --- |
| [Use Case](https://en.wikipedia.org/wiki/Use_case) | [Test Case](http://en.wikipedia.org/wiki/Test_case) |
| Use Case: In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human or other external system. | Test Case: A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. |
| Use case describes us the overview of the software functions to reach the final destination. | It demonstrates the test progress and the interaction between the software and the user. |
| The root of the use case is the requirement of the Business. | The root of the test case is the framework of the Use case. |
| It mostly interacts with the user. | It interacts with the results. |
| Use case can be completed in one time. | Test case required repeated process of the testing. |
| It is applicable for business purpose, where it shows the framework of the software and the growth of the process | Test case is obviously for Test Engineers, they understand it well. |
| There is need of documents and research for Use case. | Test case depends on the test scripts, each test scripts complete one step. |
| Use case results are not verified. | Test case results are verified, it checks if the result that shows in Use cases is functioning properly or not. |
|  |  |
| Usually Use cases are described by the Analyst from business; they collect the requirement by research and in the report. | Test case only described or designed by the specialists. They can be software analyst, QA team, however mostly by the Test Engineer. |

**Prerequisites of test planning**

* Test objectives
* Acceptance criteria
* Assumption
* People issues
* Constraints

**STLC (software testing life cycle)**

STLC is the testing process which is executed in systematic and planned manner.



***Requirement Analysis***

In this step Quality Assurance (QA) team understands the requirement in terms of what we will testing & figure out the testable requirements.

|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| Following documents should be available:  – Requirements Specification.  – Application architectural  Along with above documents Acceptance criteria should be well defined. | Prepare the list of questions or queries and get resolved from Business Analyst, System Architecture, Client, Technical Manager/Lead etc.  Make out the list for what all Types of Tests performed like Functional, Security, and Performance etc.  Define the testing focus and priorities.  List down the Test environment details where testing activities will be carried out.  Checkout the Automation feasibility if required & prepare the Automation feasibility report. | List of questions with all answers to be resolved from business i.e. testable requirements  Automation feasibility report (if applicable) |

***Test Planning***

Roadmap for the entire testing activity.

|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| Requirements Documents (Updated version of unclear or missing requirement).  Automation feasibility report. | Define Objective & scope of the project.  List down the testing types involved in the STLC.  Test effort estimation and resource planning.  Selection of testing tool if required.  Define the testing process overview.  Define the test environment required for entire project.  Prepare the test schedules.  Define the control procedures.  Determining roles and responsibilities.  List down the testing deliverable.  Define the entry criteria, suspension criteria, resumption criteria and exit criteria.  Define the risk involved if any. | [Test Plan](http://www.softwaretestingclass.com/test-plan-template/) or Test strategy document.  Testing [Effort estimation](http://www.softwaretestingclass.com/software-estimation-techniques/)document. |

**Contents in test plan**

**1. Introduction**

**2. Objective**

**3. Scope**

Functions to be tested and functions not to be tested.

**4. References**

**5. Testing Process Overview**

**6. Test Strategy**

Testing Types and Tools

**7. Test Environment**

**8. Test Schedule**

**9. Control Procedure**

**10. Roles And Responsibilities**

**11. Deliverable**

**12. Entry Criteria**

**13. Suspension Criteria**

**14. Resumption Criteria**

**15. Exit Criteria**

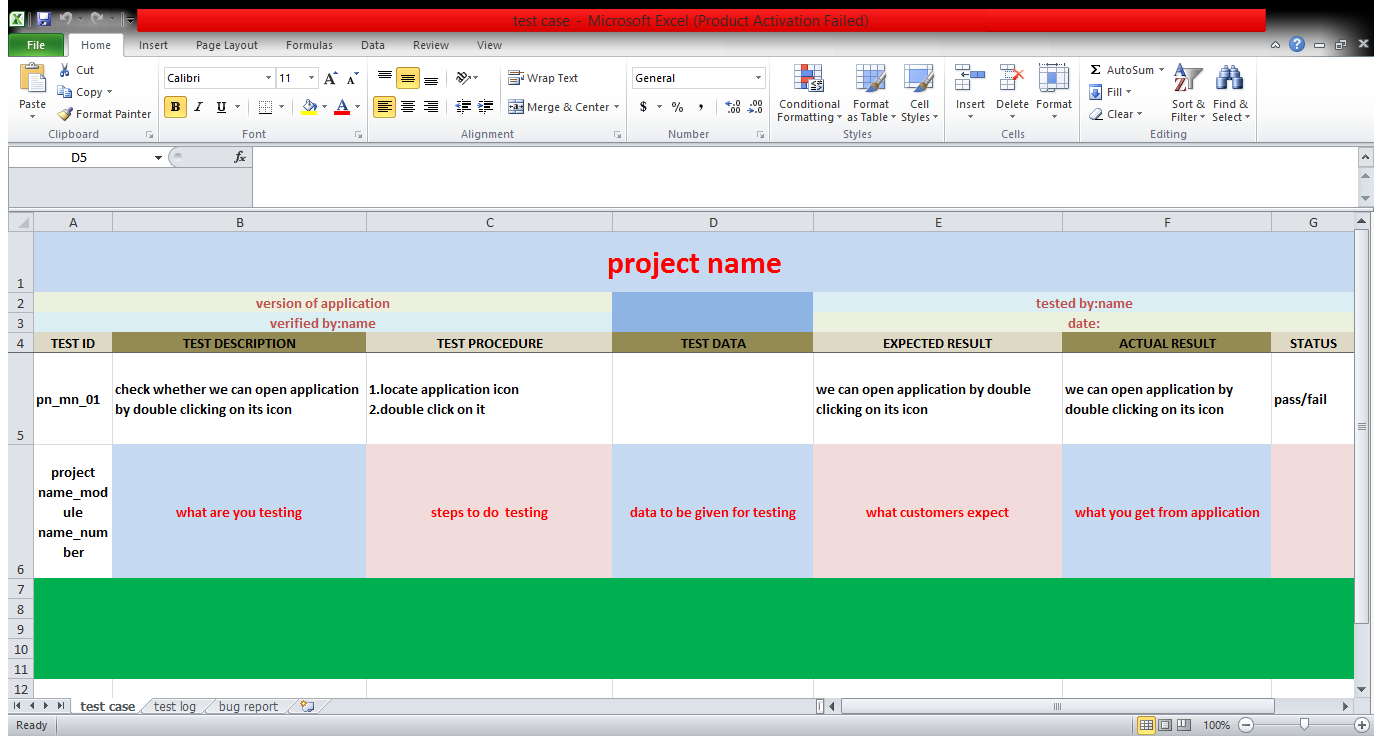
**16. Risk**

**17. Acronyms**

***Test case development***

|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| Requirements Documents (Updated version of unclear or missing requirement).  Automation feasibility report. | Preparation of test cases.  Preparation of test automation scripts (if required).  Re-requisite test data preparation for executing test cases. | Test cases.  Test data.  Test Automation Scripts (if required). |

**TEST CASE FORMAT**



***Test Environment Setup:***

Basically test environment decides on which conditions software is tested. This is independent activity and can be started parallel with Test Case Development. In process of setting up testing environment test team is not involved in it. Based on company to company may be developer or customer creates the testing environment. Meanwhile testing team should prepare the smoke test cases to check the readiness of the test environment setup.

|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| Test Plan is available.  Smoke Test cases are available.  Test data is available. | Analyze the requirements and prepare the list of Software & hardware required to set up test environment.  Setup the test environment.  Once the Test Environment is setup execute the Smoke test cases to check the readiness of the test environment. | Test Environment will be ready with test data.  Result of Smoke Test cases. |

# *Test Execution:*

In this phase testing team start executing test cases based on prepared test planning & prepared test cases in the prior step.

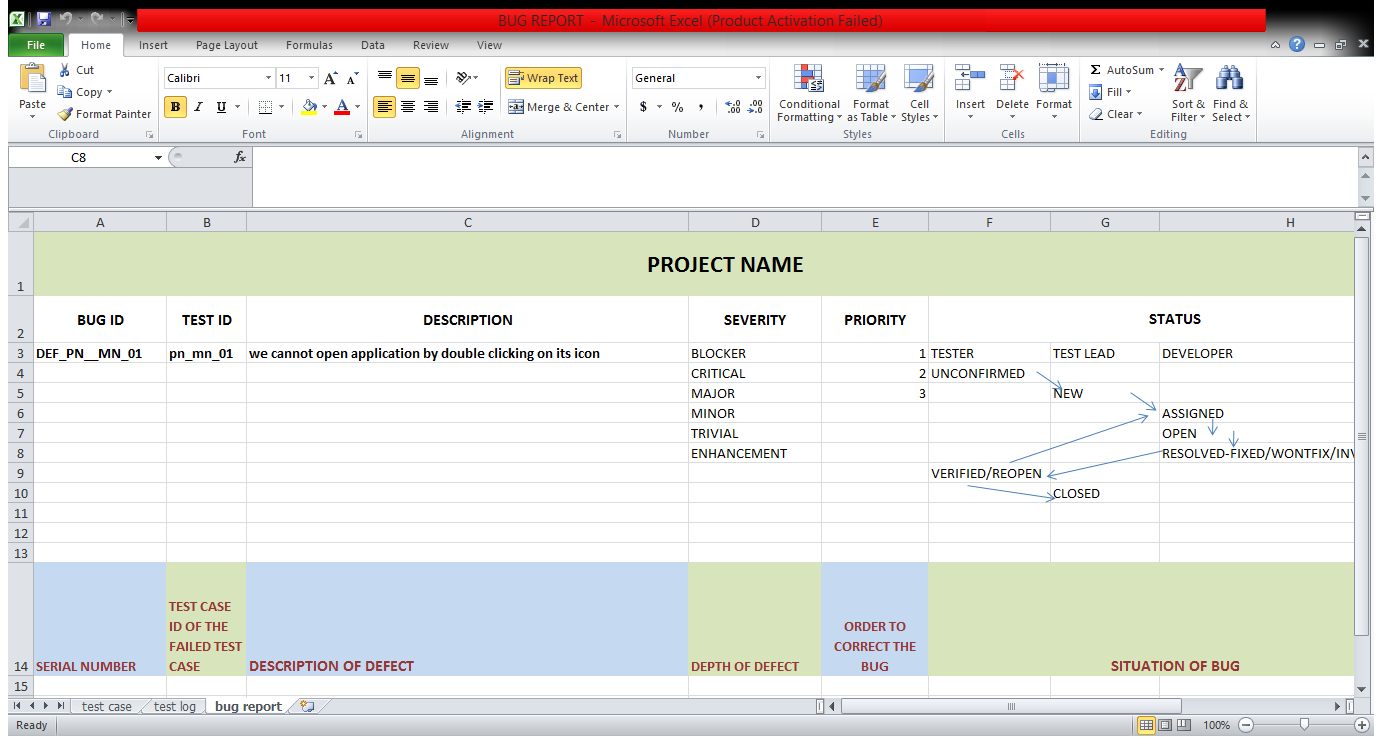
|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| [Test Plan](http://www.softwaretestingclass.com/test-plan-template/)or Test strategy document.  Test cases.  Test data. | Based on test planning execute the test cases.  Mark status of test cases like Passed, Failed, Blocked, Not Run etc.  Assign Bug Id for all Failed and Blocked test cases.  Do Retesting once the defects are fixed.  Track the defects to closure. | Test case execution report.  Defect report. |

***Test Cycle Closure:***

Call out the testing team member meeting & evaluate cycle completion criteria based on Test coverage, Quality, Cost, Time, Critical Business Objectives, and Software. Discuss what all went good, which area needs to be improve & taking the lessons from current STLC as input to upcoming test cycles, which will help to improve bottleneck in the STLC process. Test case & bug report will analyze to find out the defect distribution by type and severity. Once complete the test cycle then test closure report & Test metrics will be prepared.

|  |  |  |
| --- | --- | --- |
| **Entry Criteria** | **Activities** | **Deliverable** |
| Test case execution is completed  Test case Execution report  Defect report | Evaluate cycle completion criteria based on Test coverage, Quality, Cost, Time, Critical Business Objectives, and Software Prepare test metrics based on the above parameters.  Prepare Test closure report .Share best practices for any similar projects in future | Test Closure report  Test metrics |

**DEFECT REPORTING**



**Defect severity**

Severity is defined as the degree of impact a defect has on the development or operation of a component application being tested.

Higher effect on the system functionality will lead to the assignment of higher severity to the bug. Quality Assurance engineer usually determines the severity level of defect

6 types

* + Blocker Blocks development and/or testing work
  + Critical crashes, loss of data, severe memory leak
  + Major major loss of function
  + Minor minor loss of function, or other problem where easy

workaround is present

* + Trivial cosmetic problem like misspelled words or misaligned

text

* + Enhancement Request for enhancement

**DEFECT PRIORITY**

Correcting the order of the bug. Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.1

Defect priority can be categorized into three class

**Low:** The defect is an irritant but repair can be done once the more serious defect has been fixed  
**Medium:** During the normal course of the development activities defect should be resolved. It can wait until a new version is created  
**High:** The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.

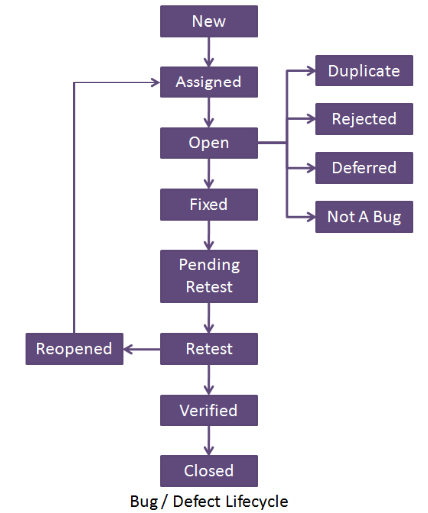
Let see an example of low severity and high priority and vice versa

**A very low severity with a high priority:** A logo error for any shipment website can be of low severity as it not going to affect the functionality of the website but can be of high priority as you don't want any further shipment to proceed with wrong logo.

**A very high severity with a low priority:** Likewise, for flight operating website, defect in reservation functionality may be of high severity but can be a low priority as it can be scheduled to release in a next cycle.

**BUG LIFECYCLE**

**Defect life cycle** is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed. The bug has different states in the Life Cycle.



**Bug life cycle includes following steps or status:**

1. **New:**  When a defect is logged and posted for the first time. Its state is given as new.
2. **Assigned:**  After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It’s state given as assigned.
3. **Open:** At this state the developer has started analyzing and working on the defect fix.
4. **Fixed:** When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’ and the bug is passed to testing team.
5. **Pending retest:**  After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
6. **Retest:**  At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
7. **Verified:** The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.
8. **Reopen:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.
9. **Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.
10. **Duplicate:** If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate**“.**
11. **Rejected:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.
12. **Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are **priority**of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
13. **Not a bug:**  The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of color of some text then it is not a bug but just some change in the looks of the  application.

**Bug tracking Tools**

Mantis,jira,bugzilla etc

**Testing Reports**

Reporting test results is a continous process whenever problems are encountered they should be reported to the decision imported content of a final test report are

* Define the scope
* Present the test results
* Efficient conclusion

***8 interim reports***

The **Interim Test Report** provides an internal status of the current formal **testing** activities. It provides a summary of open defects and associated severity.

**1) Functional testing status**

The report will show the percentage of functions which have been fully tested, not tested, partially tested.

It is represented in graphical format.

**2) Functions working timeline**

Reports show the current status of testing versus actual plan.

**3) Expected versus actual defect uncovered timeline**

Report is to show the number of defects uncovered is above or below the expected number.

**4) Defect detector versus corrected gap**

It will show the number of defects detected v/s number of defects being corrected.

**5) Average age uncorrected defects by time**

It will show the breakdown of gap by no of days it has taken to complete or correct the defect identified.

**6) Defect distribution**

Report is to explain how defects are distributed among the modules or function being tested.

**7) Relative defect distribution**

Purpose is to normalize the defect distribution presented.

**8) Testing action report**

It is designed for the test manager or the development manager .it include all the details about result.

**METRIC**

Software Metrics are used to measure the quality of the project. Simply, Metric is a unit used for describing an attribute. Metric is a scale for measurement.

**Test metrics example:**

* How many defects exist within the module?
* How many test cases are executed per person?
* What is the Test coverage %?

**Test Metrics are used to,**

1. Take the decision for next phase of activities such as, estimate the cost & schedule of future projects.
2. Understand the kind of improvement required to success the project
3. Take a decision on process or technology to be modified etc.

**3 types**

* Process metric
* Product metric
* Software quality metric

**Process metric**

The metric used to measure the characteristics of the method, technique and tools employed in developing implementing and maintaining the s/w s/m.

**Product metric**

Metric used to measure the characteristics of the documentation and code.

**Software quality metric**

Metric contain the software i/p and o/p details

**Test metric life cycle**

***Analysis***

* Identification of metrics
* Define the identified metrics

***Communication***

* Explain the need of metric to other testing team.

***Evaluation***

* Capture and verify the data.

***Report***

* Develop the report with effective conclusion
* Take feedback

***Factors of metrics***

* Reliability

Refers to the consistency of measurement.

* Ease of use

How easy to capture and the use the measurement data

* Timeliness

Refers to whether the data was reported in sufficient time to impact the decision

* Calibration

The movement of a metrics so it becomes more valid.

***How to calculate test metrics***

Percentage of test cases executed= (no. of test cases executed)/ (total no of test cases written)\*100.