|  |
| --- |
|  |
| **QEdge Technologies**  **Testing Process**  **Version 1.0**  **Dated 01-August-2015** |
|  |
|  |

Revision History

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sl. No | Date | Version # | Section / Page # changed | Details of changes made | Author | Reviewed by | Approved by |
|  | 6-12-15 | 1.0 |  | Initial Release | Santhosh Kumar | Mr.k.k | Mr.K.K |

Confidentiality Notice

This document contains proprietary information of QEdge Technologies.  No part of this document may be reproduced, stored, copied, or transmitted in any form or by any means now known or hereinafter invented, electronic, digital, mechanical, photocopying, scanning, recording or by any information storage or retrieval system, without the express consent of QEdge Technologies Pvt Lmt.

TABLE OF CONTENTS

[1. INTRODUCTION 4](#_Toc114995387)

[1.1 PURPOSE 4](#_Toc114995388)

[1.2 BACKGROUND 4](#_Toc114995389)

[1.3 ENTRY CRITERIA 4](#_Toc114995390)

[1.4 SCOPE 5](#_Toc114995391)

[1.5 TAILORING GUIDELINES 5](#_Toc114995392)

[1.6 DOCUMENT OVERVIEW 5](#_Toc114995393)

[1.7 REFERENCES 5](#_Toc114995394)

[2. Plan Test 7](#_Toc114995395)

[2.1 Purpose: 7](#_Toc114995396)

[2.2 Roles and Responsibilities: 7](#_Toc114995397)

[2.3 Entry Criteria. 7](#_Toc114995398)

[2.4 Control. 7](#_Toc114995399)

[2.5 Process Activities. 8](#_Toc114995400)

[3. Design Test 11](#_Toc114995401)

[3.1 Purpose: 11](#_Toc114995402)

[3.2 Roles and Responsibilities: 11](#_Toc114995403)

[3.3 Entry Criteria. 11](#_Toc114995404)

[3.4 Control. 11](#_Toc114995405)

[3.5 Process Activities. 11](#_Toc114995406)

[4. Implement Test 13](#_Toc114995407)

[4.1 Purpose: 13](#_Toc114995408)

[4.2 Roles and Responsibilities: 13](#_Toc114995409)

[4.3 Entry Criteria. 13](#_Toc114995410)

[4.4 Control. 13](#_Toc114995411)

[4.5 Process Activities. 13](#_Toc114995412)

[5. REVIEW test 15](#_Toc114995413)

[5.1 Purpose: 15](#_Toc114995414)

[5.2 Roles and Responsibilities: 15](#_Toc114995415)

[5.3 Entry Criteria. 15](#_Toc114995416)

[5.4 Control. 15](#_Toc114995417)

[5.5 Process Activities. 15](#_Toc114995418)

[6. Execute Test 16](#_Toc114995419)

[6.1 Purpose 16](#_Toc114995420)

[6.2 Roles and Responsibilities: 17](#_Toc114995421)

[6.3 Entry Criteria. 17](#_Toc114995422)

[6.4 Control. 17](#_Toc114995423)

[6.5 Process Activities. 18](#_Toc114995424)

[7. Report and triage errors/Bugs 19](#_Toc114995425)

[7.1 Purpose: 19](#_Toc114995426)

[7.2 Roles and Responsibilities: 19](#_Toc114995427)

[7.3 Entry Criteria. 20](#_Toc114995428)

[7.4 Control. 20](#_Toc114995429)

[7.5 Process Activities. 20](#_Toc114995430)

[8. TERMS, DEFINITIONS and ACRONYMS 22](#_Toc114995431)

[8.1 Terms and Definitions 22](#_Toc114995432)

Testing Process

# INTRODUCTION

## PURPOSE

The mission of QEdge Technologies is to ensure that software systems and products are delivered to the customer in the best possible working condition, using established standard procedures to measure defects, determine their root causes, and take action to prevent their future insertion.

The purpose of this document is to assist QEDGE TECHNOLOGIES projects in establishing a Software Testing function. This Testing process describes seven activities. They are:

1. Plan Test
2. Design Test
3. Implement Test
4. Evaluate Developed Test
5. Execute Test
6. Report and Evaluate Results
7. Quality Measures

Figure 1 is an overview of the Testing process activities.

## BACKGROUND

The KPAs from the CMM at levels 2 and 3 were also reviewed to extract any requirements that relate to testing. The test process specification was drafted based on the processes in place.

## ENTRY CRITERIA

There are three entrance criteria for following this process are:

1. Program and project management commitment to Hybrid and the QEDGE TECHNOLOGIES’s Processes.
2. Quality Factors have been identified.
3. A Developed Domain Model and the management assigns a Technical Leader to whom a knowledge transfer will be done on the domain.

QEDGE TECHNOLOGIES Policy is our written organizational policy for implementing SQA to provide management with appropriate visibility into the process being used by the software project and of the products being built. The SQA Policy is based on the SQA KPA of the SEI -CMM. Program and project management must commit to this policy and ensure it is followed.

Quality Factors are characteristics, which a software product exhibits that reflect the degree of acceptability of the product to the user. Identifying quality factors is a software engineering activity that involves identifying and building quality factors in the software program, which can be measured and evaluated.

## SCOPE

This process can be applied to a specific project within QEDGE TECHNOLOGIES or an outside organization.

## TAILORING GUIDELINES

This document features at the current practices at QEDGE TECHNOLOGIES and the areas of Software Testing that are applicable to the current projects. Projects that find it necessary to provide more in-depth details to their Testing processes may add additional requirements or modify existing ones to carry out the activities.

## DOCUMENT OVERVIEW

Section 1, Introduction, provides the purpose, background, entry criteria to this process, scope, tailoring guidelines, and references.

The following Sections takes up each activity at a time to details its requirements, responsibility, Scope and process activities.

## REFERENCES

The following documents were used to develop this process document:

1. Capability Maturity Model for Software, Version 1.1, February 1993.
2. Rational Unified Process

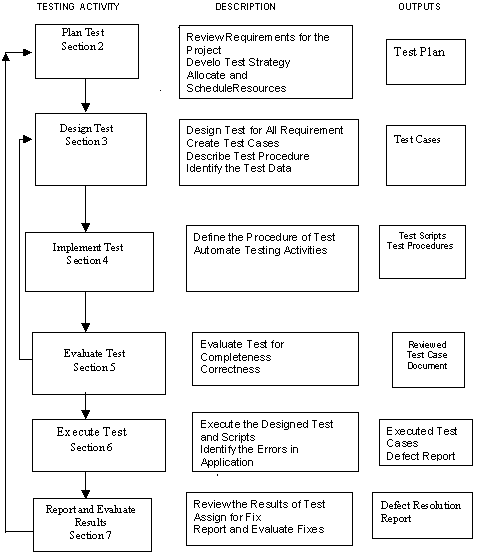


Figure 1. Testing Process Overview

# Plan Test

## Purpose:

To identify and describe the testing that will be implemented and executed. This is accomplished by generating a test plan, which contains the requirements for test and test strategies. A single test plan may be developed, describing all the different types of tests to be implemented and executed, or one test plan per type of test may be developed. At QEDGE TECHNOLOGIES we will develop a singe Test Plan for all the Tests to be performed.

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Quality Management  Software Quality Manager (SQM) | Assign Quality Lead (QL) with specific Technical responsibilities.  Provide technical data and requirements to Quality Lead.  Provide resources for QL.  Review Test Plan activity status and aid in problem resolution.  Approve Test Plan |
| Quality Lead (QL) | Identify Requirement For Test  Assess Risk.  Identify Resources  Create Schedule  Manage Test Plan Activity.  Generate and distribute Test Plan. |
| Test Engineer (SQA) | Develop Test Strategy |

## Entry Criteria.

Quality Lead (QL) is assigned upon project initiation.

Initial High- Level Requirements are documented

## Control.

Controls for this activity are listed below.

a. Requirements determine the scope and depth of Testing.

RUP and Test Process provide the basis for developing detailed instructions required to complete Planning tasks.

## Process Activities.

### Identify Requirements for Test

Identifying the requirements for test is the start of the test planning activity. The requirements for test identify what is being tested, and the scope and purpose of the test effort. Requirements for test are also used to determine the overall test effort (for scheduling, test design, and so on) and are used as the basis for test coverage.

The following is performed to identify requirements for test:

#### *Review all materials:*

The requirements for test may be identified from many sources, therefore it is important, that, as the first step, all the materials available for the application / system to be developed should be reviewed. The most common sources of requirements for test include Business Design Document, Use Case Document, CR Analysis Report, Technical design Document, existing requirement lists, use-case models, use-case realizations, supplemental specifications, design requirements, business cases, interviews with end-users, and review of existing systems.

#### *Indicate the requirements for test:*

Independent of the source of the requirement for test, there must be some form of identification that a requirement is going to be the target of a test. This results in the generation of a list of requirements for test. This list may be based upon an existing list, or newly generated.

The output of this step is a report (the list) identifying those requirements that will be the target of test.

This identification of test items is done by the Quality Lead in consultation with PM, and Test Engineers,

### Assess Risk

#### *Identify and justify an operational profile factor for test.*

Most application has functions that are used often and others that are infrequently used. Therefore, to acceptably test an application, one must ensure not only are the highest risk requirements for test tested, but also those that are frequently used (as these often have the highest end-user visibility).

Identify an operational profile factor for each requirement for test and a statement justifying why a specific factor value was identified. This is accomplished by reviewing the business case(s) or by conducting interviews with end-users and their managers. Another method is to observe the end-users as they interact with the system or use software monitors / recorders to record end-user interaction with the system (for analysis).

#### *Identify and justify a test priority factor.*

Upon identifying and justifying the test risk and operational profile for each requirement for test, a test priority factor should be identified and justified. The test priority factor identifies the relative importance of the test requirement, and the order or sequence in which it will be tested.

Using the risk factors, operational profiles, contractual obligations, other constraints, or a combination of all of these identifies the test priority factor. It is important to consider all these factors when identifying the test priority to ensure that the testing is appropriate and focused.

### Develop Test Strategy

#### *Purpose*

Identifies and communicates the test techniques and tools

Identifies and communicates the evaluation methods for determining product quality and test completion

#### *Identify and describe the approach to test*

The approach to test is a statement (or statements) describing how the testing will be implemented. This should state or refer to what will be tested, the major actions taken while testing, and how the results will be verified. The statements should provide enough information to the reader so they can understand what will be tested

#### *Identify the criteria for test*

The criteria for test are objective statements indicating the value(s) used to determine / identify when testing is complete, and the quality of the application-under-test. The test criteria may be a series of statements or a reference to another document (such as a process guide or test standards).

#### *Identify any special considerations for test*

Any special considerations for testing or dependencies should be listed, such as those shown below:

Test databases are to be restored by Operations resources

Legacy system synchronization must be made available (or simulated)

See Guidelines: Test Plan for additional information on developing test strategies.

### Identify Resources

Once it's been identified what's being tested and how, there is the need to identify who will do the testing and what is needed to support the test activities. Identifying resource requirements includes determining what resources are needed, including the following:

#### *Human resources (number of persons and skills)*

##### Human resources are to be identified for the testing team after estimating the volume of work to be done in the required time frame. Skill set of the resources should be in line with the Project needs.

#### *Test environment (includes hardware and software)*

##### A separate environment has to be created for the testing team, which resembles the production environment. All hardware and software resources must be identified for test execution phase.

#### *Tools*

##### There will be request from the Project Manager or the Client asking for the automation testing. Quality Lead will analyse the need of the project to have an automation tool / testing. If required, then tool to be used for the project will be identified.

#### *Data*

##### Quality Lead will analyze the requirements with the testing team and if there is need for generating Data before testing, it must be identified

### Create Schedule

Test estimation should consider partitioning the effort differently within each phase of the testing lifecycle as the weight (of effort) for some types of vary during the lifecycle.

Testing effort needs to include time for regression test.

#### *Estimate test effort*

Testing effort will be estimated in terms of Resource Planning Task break up tables or Work Break up Schedules

#### *Generate test schedule*

Testing Schedule will be planned in Task break up tables or Work Break up Schedules

#### *Generate Test Plan*

To organize and communicate to Project Stakeholders and testing team about the activities that planned during the testing of the project.

### 2.5.1.16 *Approve Test Plan*

The Test Plan generated after all the process activities will be thoroughly evaluated and approved by the management. This will be a reference for further testing activities.

# Design Test

## Purpose:

To identify a set of verifiable test cases for each build.

To identify test procedures that show how the test cases will be realized.

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Software Quality Manager  (SQM) | Provide Resources for Designing. |
| Quality Lead | Review Application workflow.  Review Designed Tests and Manage Test Design Activity |
| Test Engineer (SQA) | Identify and Describe Test Cases  Identify Test Case Data |

## Entry Criteria.

An approved Test Plan

Test Deliverables are identified.

Requirements are base lined.

Test Environment is defined and established.

## Control.

Controls for this activity are listed below.

* Requirements determine the scope and depth of Testing.
* The developed application architecture and Test Requirements specifies the Tests to be performed. Initially only functional testing is accommodated.
* RUP and Test Plan provide the basis for developing detailed instructions required to complete Testing tasks.

## Process Activities.

### Analyse application workflows

The purpose of this step is to identify and describe the actions and / or steps of the actor when interacting with the system. These test procedure descriptions are then used to identify and describe the test cases necessary to test the application.

Notes: These early test procedure descriptions should be high-level, that is, the actions should be described as generic as possible without specific references to actual components or objects.

For each use case or requirement,

* Review the use case flow of events, or
* Walk through and describe the actions / steps the actor takes when interacting with the system

### Identify and describe test cases

The purpose of this step is to establish what test cases are appropriate for the testing of each requirement for test.

The Test Case will have the following important attributes

1. Test Description – explaining the actions do be done on the application
2. Expected Result – expecting the desired output

The Test Cases for a particular module will be documented in a Single document, the document will be base lined after the first round of design is complete for a specific requirement baseline for the particular module. This document will evolve throughout the development lifecycle to accommodate change requirement or changes in the application. This document will be under Version Control and any changes made have to be documented.

#### Naming Conventions

To be followed as per base lined Process

##### Name of document:

To be followed as per base lined Process

##### Test Case Number:

To be followed as per base lined Process

##### Version:

To be followed as per base lined Process

### Identify test case data

Using the Test Cases created above, review the test cases and identify the actual values that support the test cases. Data for three purposes will be identified during this step:

Data values used as input

Data values for the expected results

Data needed to support the test case, but is neither used as input nor output for a specific test case.

# Implement Test

## Purpose:

To create or generate reusable test scripts using automation tools

To maintain traceability of the test implementation artifacts back to the associated test cases and use cases or requirements for test using automation tools

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Software Quality Manager  (SQM) | Provide Resources for Automation  Training of Test Roles on Automation Tools |
| Quality Lead (QL) | Review Application workflow.  Manage Test Implementation Activities |
| Test Engineer (SQA) | Record, Generate or Program Test Scripts. |

## Entry Criteria.

Test Plan is drafted and released to all mentioned in the distribution list.

Test Deliverables are identified.

Requirements are baselined.

Test Environment is defined and established.

## Control.

Controls for this activity are listed below.

* Requirements determine the scope and depth of Testing.
* The developed application architecture and Test Requirements specifies the Tests to be performed initially only functional testing is accommodated.
* RUP and Test Plan provide the basis for developing detailed instructions required to complete Testing tasks.

## Process Activities.

### *Record, generate, or program (enhance) test scripts*

#### Create, generate, or acquire test scripts

For each structured test procedure in the test model at least one test script is created or generated.

The following steps are performed to create, generate, or acquire test scripts:

Review existing test scripts for potential use

Set-up the test environment (including all hardware, software, tools, data, and application build)

Initialize the environment (to ensure the environment is in the proper state or condition for the test)

Create or acquire the test scripts:

Record / capture: for each structured test procedure, execute the test procedure to create a new test script by following the steps / actions identified in the structured test procedure and using the appropriate recording techniques (to maximize reuse and minimize maintenance)

Modifying existing scripts: edit the existing manually, or delete the non-required instructions and re-record the new instructions using the recording description above

Programming: for each structured test procedure, generate the instructions using the appropriate programming techniques

To automatically generate test scripts, refer to your specific test script generation tool.

Continue to create, generate, or acquire test scripts until the desired / required test scripts have been created

Modify the test scripts as necessary (as defined in the test model)

#### Create / maintain external data sets

To create external data sets, the following steps are performed:

Review the test model, test cases, and structured test procedures

Create the data sets using the appropriate tools and methods

Modify the test scripts to utilize the data sets

Test / debug test scripts

Upon the completion of creating, generating, or acquiring test scripts, they should be tested / debugged to ensure the test scripts implement the tests appropriately and execute properly. This step should be performed using the same version of the software build used to create / acquire the test scripts.

The following steps are performed to test / debug test scripts:

Set-up the test environment (if necessary)

Re-initialize the environment

Execute the test scripts

Evaluate Results

Determine appropriate next action:

Results as expected / desired: no actions necessary

Unexpected results: determine cause of problem and resolve

# REVIEW test

## Purpose:

Test coverage is the measurement of testing completeness, and is based on the coverage of testing, expressed either by the coverage of test requirements and test cases, or the coverage of executed code.

The purpose of review test is to generate and deliver the test review summary. This is accomplished by reviewing the test results, identifying and logging change requests, and calculating the key measures of test. The test review summary presents the test results and key measures of test in an organized format and is used to assess the quality of the target-of-test and the quality of the test process.

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Software Quality Manager  (SQM) | Review Test Review Activity Status  Decide Coverage Measures |
| Quality Lead (QL) | Submit Test for Review |
| Project Manager | Review Test  Submit Change Request for Tests |

## Entry Criteria.

Test Design is complete.

## Control.

Controls for this activity are listed below.

* The developed application architecture and Test Requirements specifies the Tests to be performed initially only functional testing is accommodated.
* RUP and Test Plan provide the basis for developing detailed instructions required to complete review tasks.

## Process Activities.

### Submit Tests for Review

On completion of the Test Design Activity for each phase, the Designed Tests will be submitted to the designated SQA of the particular project.

### Decide Coverage Measures

#### Requirements-based test coverage

Requirement Traceability Matrix document is prepared and it helps in making sure that the requirements are not missed out. This would be done where the BDD is available with proper documentation and numbering done. For BDDs, which are incomplete, RTMs will not be prepared.

### Analyse Test Case and Submit Change Requests for Test Design

After the Test Design or the Implementation activities, the artifacts of test were reviewed to ensure that the designed test was not influenced by non-target-of-test factors.

In this activity, the test Cases are analysed to identify the details regarding the differences between the expected tests from the field data and the actual test results. Differences indicate potential defects in the target-of-test and should be entered into a tracking system as Test Summary, and the changes, which are required, are incorporated into the test case document.

After all the Designed Tests are reviewed the Test Case Document is approved for execution.

# Execute Test

## ****Purpose****

Tests will be executed at different pre defined Level and phases within the software build:

Unit Level Testing (not applicable for TEs)

Module Level Testing

Integration Level Testing (not applicable TEs)

System Level Testing

Regression Level Testing

Execution will span throughout the SDLC.

The Entry and the Exit Criteria for the Level of Testing are classified under the different identified Levels of testing.

### *Unit Test Stage:*

The developers do unit level of testing. Here the focus in on the unit of programme structure, local data structure, boundary conditions, independent paths, program level field and functional checks.

### *Module Test Stage:*

The Test engineers do Module level testing. Here the test focuses on the functionality of the module at user level. The test includes Interfaces, boundary conditions, field level validation, field level check, user interface checks in terms of usability.

### *Integration Test Stage:*

The purpose of the Integration Test Stage is to ensure that the assembly of the system’s components collaborates as intended, as well as that the increment has the right behaviour.

The developers do the integration level of testing. They checks for Data Dependency and Data Transfer Check

### *System Test Stage:*

The purpose of the System Test Stage is to ensure that the complete system functions as intended. The system integrator compiles and links the system in increments. Each increment needs to go through testing of the functionality that has been added, as well as all tests the previous builds went through (regression tests).

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Software Quality Manager  (SQM) | Review Test Execution Activity Status |
| Quality Lead (QL) | Track Test Execution |
| Test Engineer (SQA) | Execute Test Cases  Evaluate Execution of Test  Verify Test Results |

## Entry Criteria.

### *Entry To Unit Test Stage:*

* Clean Compile of the Unit Under Test
* [Unit Test Cases](..\Templates\MSoffice\Testing\Unit%20Level%20Test%20Case.doc) Ready

### *Entry To Module Test Stage:*

* Complete testable Module availability
* Module Testing related Test Cases Ready

### *Entry to Integration Test Stage:*

* Unit Test Complete with all Defects Resolved

### *System Test Stage:*

* Approved Test Plan
* Approved Test Cases
* Approved RTM
* Unit Test Cases execution report should be available to the testing team (Optional)
* Approved Internal Software Release Notes (SRN)

## Control.

Controls for this activity are listed below.

* The developed application architecture and Test Requirements specifies the Tests to be performed initially only functional testing is accommodated.
* RUP and Test Plan provide the basis for developing detailed instructions required to complete Evaluation tasks.

## Process Activities.

### *Execute Test Procedures/Cases*

To execute the tests, the following steps should be followed:

* Set-up the test environment to ensure that all the needed components (hardware, software, tools, data, etc.) have been implemented and are in the test environment.
* Execute the test procedures.

Note: executing the test procedures will vary dependent upon whether testing is automated or manual.

* Automated testing: The test scripts created during the Implement Test activity are executed.
* Manual execution: The structured test Cases developed during the Design Test activity are used to manually execute test.

### *Evaluate Execution of Test*

The execution of testing ends or terminates in one of two conditions:

* Normal: all the test Cases (or scripts) execute as intended and to completion.

If testing terminates normally, then continue with Verify Test Results

* Abnormal or premature: the test procedures (or scripts) did not execute completely or as intended. When testing ends abnormally, the test results may be unreliable. The cause of the abnormal / premature termination needs to be identified, corrected, and the tests re-executed before any additional test activities are performed.

If testing terminates abnormally, continue with Recover From Halted Tests.

### *Verify Test Results*

Upon the completion of testing, the test results should be reviewed to ensure that the test results are reliable and reported failures, warnings, or unexpected results was not caused by external influences (to the target-of-test), such as improper set-up or data.

The most common failures reported when test procedures and test scripts execute completely, and their corrective actions are given below:

* Test verification failures - this occurs when the actual result and the expected result do not match. Verify that the verification method(s) used focus only on the essential items and / or properties and modify if necessary.
* Unexpected GUI windows - this occurs for several reasons. The most common is when a GUI window other than the expected one is active or the number of displayed GUI windows is greater than expected. Ensure that the test environment has been set-up and initialized as intended for proper test execution.
* Missing GUI windows - this failure is noted when a GUI window is expected to be available (but not necessarily active) and is not. Ensure that the test environment has been set-up and initialized as intended for proper test execution. Verify that the actual missing windows are / were removed from the target-of-test.

### *Recover From Halted Tests*

There are two major types of halted tests:

* Fatal errors - the system fails (network failures, hardware crashes, etc.)
* Test Script Command Failures - specific to automated testing, this is when a test script cannot execute a command (or line of code).

Both types of abnormal termination to testing may exhibit the same symptoms:

* Many unexpected actions, windows, or events occur while the test script is executing
* Test environment appears unresponsive or in an undesirable state (such as hung or crashed).

To recover from halted tests, do the following:

* Determine the actual cause of the problem
* Correct the problem
* Re-set-up test environment
* Re-initialize test environment
* Re-execute tests

# Report and triage errors/DEFECTS (or)Bugs

## Purpose:

Defects or Bugs are the ultimate work product of the testing process. Dealing with defects is a necessary part of using or developing software. Dealing with them in the wrong way can waste days and weeks of your time and can delay defect repair by days.

Herein, we consider "bug" a synonym of "defect".

## Roles and Responsibilities:

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Software Quality Manager  (SQM) | Review Defects |
| Quality Lead (QL) | Triage/Evaluate Bugs |
| Project Manager (PM) | Triage/Evaluate Bugs  Assign Defects for Fixes |
| Test Engineer (SQA) | Submit/Report Defects  Close Defects |
| Developer (DEV) | Debug Application and Fix – Resolve Bugs |

## Entry Criteria.

All Design Tests are executed for the Test Cycle.

## Control.

Controls for this activity are listed below.

* The developed application architecture and Test Requirements specifies the Tests to be performed initially only functional testing is accommodated.
* RUP and Test Plan provide the basis for developing detailed instructions required to complete Evaluation tasks.

## Process Activities.

### *Report Defects*

As the test execution is taking place, If a defect is found, it is entered in the defect tracking tool, PVCS tracker/BugZilla, which is available to the whole project team. Maintaining the defects and its status is the responsibility of the testing team

Defects can have the screenshots attached for more clarity on the defects.

#### Defect\_id

Is a unique ID to be generated by the Tester

#### Submitted by

Is the name of the Tester

#### Test Case ID

Is the unique identifier of the Test Case

#### Phase:

Specifies the Iteration the Defect was submitted (Build#, Module name)

#### Severity:

The severity of the defects could be categorized as follows:

**Fatal:**  Any defects, which are seen as show stoppers. A scenario where the application is no longer in working condition. Examples like “Page not found”, System hangs, etc.

**Major:**  Any defects which are seen as malfunctions. Deviation from the required functionality as specified in the requirement documents.

**Minor:**  Defects where there is no functionality affected. Only cosmetic changes. Look and field defects

**Suggestion and Enhancement:** These are all the points of suggestion for the product improvement but the decision of implementation depends on the decision of Project Management

**Description**: Is the details of the identified defect of the system (optional)

**Origin**: the Tester, Developer, Stakeholders or any external agencies participating in the Beta test can post Defects. Origin is the organization submitting the Defect (optional)

#### Taxonomy of Bugs:

### *Review Defects and Assign for Fix*

Once the Defects are submitted to the Project manager (Development team) will review all the posted defects for the validity. Once the Bug is concluded by a valid one it has to be assigned to a developer this is done by changing the state to Open and subsequently assign.

Any Bug, which is invalid or duplicate, Will be discussed with the project management team and decision taken on the respective defects.

### *Assign Bug Fix Priority*

Any open Bug that is assigned should be prioritized for fixes. This is done by the project Manager in case any customisation of the prioritisation. The following priorities can be assigned to the Bug all the priority labels are self explanatory:

* Critical
* High
* Medium
* Low

### *Fix Bugs and Report Resolution*

The responsibility of resolution of the defects is with the development team. Once the defects are fixed, the status of the defects is changed to “FixedForVrification” in the tracking tool. For each defect, the developer has to enter his/her observation about the defect in the remarks column if at all any.

### *Retest and Close Bugs*

On the receipt of the next SRN, each defect will be tested again which are in the scope of testing and change the status of the defect to “Closed” if the defect is fixed. Else, the defect is returned back to the development team.

If the “Closed” defect is found existing, then the status of the closed defect is changed to “Re-Open”

*Hold/As Per Design/Tester’s Error*

Hold - In case the information is not sufficient about the defect to determine the genuinity of it.

As Per Design - In case the defect related to newly encountered functionality that is not as per BDD and TCD.

Tester’s Error - In case the defect due to the wrong testing and the defect is not really a defect.

# TERMS, DEFINITIONS and ACRONYMS

## Terms and Definitions

Terms fundamental to any discussion of quality assurance include:

**PROCESS**. An organized set of activities performed for a given purpose with appropriate guidelines.

**QUALITY**. The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs.

**QUALITY POLICY**. The overall quality intentions and direction of an organization as regards to quality, as formally expressed by top management.

**QUALITY MANAGEMENT**. The aspect of overall management function that determines and implements a quality policy.

**QUALITY SYSTEM.** The organization structure, responsibilities, procedures, processes and resources for implementing quality management.

**SOFTWARE PROJECT PLAN (SPP).** The SPP describes a developer’s plans for conducting a software development effort, also known as **Project Plan**. The term “software development” is meant to include new development, modification, reuse, reengineering, maintenance, and all other activities resulting in software products. The SPP also provides the acquirer insight into, and a tool for monitoring, the processes to be followed for software development, the methods to be used, the approach to be followed for each activity, and project schedules, organization, and resources.

**SOFTWARE METRICS.** Numerical measures of the products and processes that make up the software project. The careful use of numerical measures can introduce precision and clarity to the process instead of opinion, informality, and open interpretation.

**SOFTWARE QUALITY**. Ability of software product to satisfy specified requirements.

**SOFTWARE QUALITY ASSURANCE.** Planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements, and, set of activities designed to evaluate the process by which software work products are developed and maintained.

**QUALITY ASSURANCE PLAN (QAP).** The QAP describes the plans and activities of the SQA staff.

**DOCUMENT CHANGE REQUEST (DCR)**

|  |  |
| --- | --- |
| Document Title: | Tracking Number: |
| Name: | |
| Contact: | Phone: |
| Mailing Address: | |
| Short Title: | Date: |
| Change Location:  (use section #, figure #, table #, etc.) | |
| Proposed change: | |
| Rational for Change: | |
| Note: For the Software Engineering Process Group (SEPG) to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale. | |