

## Testing Process

Version No.: 1.4

User of this document is responsible to use only the current version available on BIMPL QMS Server. This document is meant for restricted internal circulation. No part of this document shall be reproduced, stored in retrieval system or transmitted in any form or by any means – recording, photocopying, electronic and mechanical without prior permission of BIMPL Management.

**Base Information Management (P) Ltd.**

**Mumbai.**

**Table of Contents**

Introduction	4
Scope	4
Process Practitioner	4
1 Objective	4
2 Entry	4
3 Work Flow	5
4. Task	6
5 Guidelines	8
6 Checklists / Standards	10
7 Associated Templates	10
8 Control Mechanism	10
9 Verification	10
10 Tailoring	10
11 Exit	11
12 References	11
13 Glossary	11

Controlled Copy

**Document Release History :** At QMS level ♦QMS Changes♦ document will be maintained (version wise)  
At the Project level the release history will be maintained within the document..

Controlled Copy

### Introduction

- The process helps in validating the end product against the customer requirement. This process specifies activities to be followed for unit, Integration and system testing

### Scope

- This process is applicable to all development projects and change requirements against CCF-RF

### Process Practitioner

PM /PL /TL (QC), PM /PL /TL (Development),Functional (P/L),QC team.

### 1 Objective

The objective of Validation / Testing is to attain the following :

- Determining whether the system meets requirements (implied / documented) or not
- Building customer's confidence in BIMPL's process
- To detect defect and ensure that the system is thoroughly tested before the release

### 2 Entry

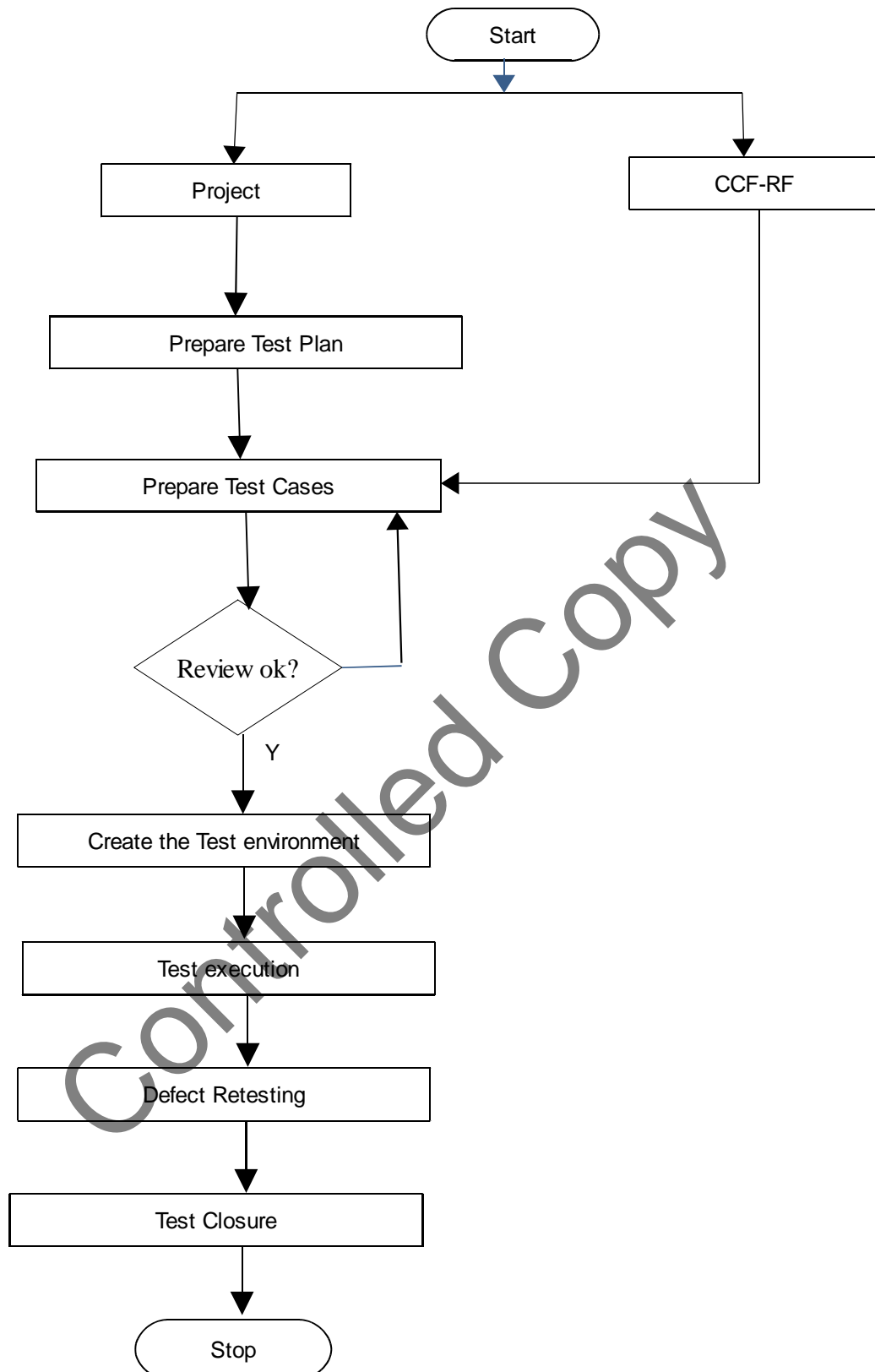
#### Criteria

- Work product is ready for Testing

#### Input

- Work product
- Existing Test Case (If available in Test case repository)
- Request form(RF)
- Technical documents (ERD,SDD,IQ,ID,CFG,DD) if applicable

## 3 Work Flow



## 4. Task

Sub Process	Input Document	Entry Criteria	Output Document	Exit Criteria	Responsibility
<b>4.1 Prepare Test Plan for Project</b>					
4.1.1. Prepare a Test plan	RF, Project Plan, Project schedule,	Project Plan, Project schedule submitted by PM (Development ),	Test Plan	Resource allocated ,Time estimated, Test bed requirement information gather and identified Test case.	PM /PL /TL
4.1.2. Review Test plan	Test Plan	Test Plan is ready	Reviewed Test Plan	Test plan reviewed	PM /PL /TL
<b>4.2 Prepare Test Cases</b>					
4.2.1. Prepare or modify Test Case based on RF (Modify in case of already exist and also increment the test case version) Refer Guidelines 5.1	RF, Impact Document	Requirement identified in RF	Test Case	Test Case prepared	QC Team
4.2.2. Test case review Baseline Test case	Test Case	Test case is Completed	Reviewed Test Case	Test Case Reviewed	PM /PL /TL
4.2.3. Update RTM	Test Case	Test Case is Ready	Updated RTM	RTM Updated	PM /PL /TL
<b>4.3 Create the Test environment</b>					
4.3.1. Establish Hardware Environment (Like Machine Configuration,Network,Other Hardware needed such as printer/scanner etc.  4.3.2. Install Software (like OS,JDK,Application Servers (Jboss / IIS / Jaguar etc.)  4.3.3. Database setup and configuration (Eg. Server Version,Client Version	RF Test Plan	Test Plan is ready		Establishes Hardware and software Environment	PM /PL /TL

4.3.4. Intimate to QC team along with delivery details mention in Internal delivery template (Through mail for performing testing.	RF, Intimation Internal delivery template mail from Developers, UTR	RF, UTR received.		Checked out work products from Repository	Tester
4.3.5. Create work products (jars, web parts, MSI etc) and deployment on test environment.	NA	Work product ready for Testing.		Test environment Ready	Tester
<b>4.4 Test execution</b>					
4.4.1. Execute the testing as per test case and update test results in Test case (Refer Guidelines 5.3)	1.Test case (without actual result) (Newly designed test Case/existing test case from test case repository)	Product or Product component is ready for testing.	1.Test case (with actual result with updation of reference No.of screen shot) 2.Defect log.	Execution of Test case is completed.	Tester
4.4.2.Log identified defects (if any) in the 'Defect log'					
4.4.3.Analysis Test result and Communicate results to the development team					
<b>4.5 Rework (If any) [Defect Retesting]</b>					
4.5.1.In case of defects do the retesting	1.Test case (with actual result) 2.Defect log *(with severity and priority ).	Retest the fixed bug, Product or Product component .	1. Passed Test case. 2. Known Issue document	Reported Bug fixed verified and tested. *Action taken is updated in defect log repository	QC team.
<b>4.6 Test Closure</b>					
4.6.1. Prepare the Test Closure	1. Passed Test case. 2. Known Issue document	Testing is completed.	1.Approved and reviewed Test case 2.User manual (if applicable)	Product or Product component is ready for release.	QC team.

4.6.2 Prepare Release note for client and send the deliverables	1.Approved Test case 2. Internal delivery template	Product or Product component is ready for deliver.	1. Release Note 2. Prepare ID/IQ in case of Projects 3. Work products (Jar, Msi, etc) in case of CCF.	Deliverable are ready to Release. Store results to Repository.	QC Team
4.6.3 Release review	1. Deliverable are ready to Release.	Release is ready	Certified Release note	Release sent to client	PM /PL /TL

## 5 Guidelines

### 5.1 Guidelines for preparing Test Cases:

- Write test case before implementation of requirements.
- Test cases are to be written for all requirements.
- Follow standard template for all test cases.
- Format followed (Alignment , Naming convention etc) by test case should be uniform for the entire application.
- Method for test cases identifier, is first 2 digits for module code, next 2 digits for Sub module code(Screen type) and next 2 digits for incremental number. Each part should be separated by dot '.'.
- Test case version no. syntax should contain the major version no., minor version no. Each part of the version should be separated by dot '.'. Major version will be incremented if functionality is added and minor version for minor changes.
- Test case identifier and its version will be maintained at centralized location (at Base office)
- Use very simple English and general words.
- All the scenario/Test cases should be easily understandable , clear and to the point.
- Maintain the Test case in flow so that execution order is maintained and time is saved.
- The prerequisite for the scenario /Test cases should be mentioned.
- Use a mix of both 'positive' and 'negative' tests.
- Provide the test data.(Print screen)
- Test case should be such that any person going to execute it or read it gets clear picture of what need to be done without needing explanation.
- Release history should be maintained for all test case.
- Request acknowledgment mail from client after release is received and deployed.

### 5.2 When Request Form is delivered to QC for testing check for the below mentioned details.

- Technical documents (ERD,SDD,IQ,ID,CFG,DD),



- Repository Path (Release Components to be taken for testing),
- Technical Details ,Functional Impact and Technical impact.
- SQL requirements.

### 5.3 Test execution:

- Start testing according to the Requirement based on the Test Case
- Create the test data for Configuration received by development team.
- Fill the actual result against the Expected result and mention the status of the result (Pass or Fail).
- If bug found then report the same to development team with specific log and print screen documents (i.e., Specific Defect Log).
- Do not write a essay about the problem. Be Specific and to the point. Try to summarize the problem in minimum words yet in effective way.
- Do not combine multiple problems even they seem to be similar. Write different reports for each problem.

### 5.4 Close the Request with the below mentioned documents

- User Manual Document (if applicable)
- Release Document
- IQ and ID Document
- DD Document (if applicable)
- SDD Document (if applicable)
- ERD Document (if applicable)
- Executables
- RTM (if applicable), Test cases and Screen shots
- Known Issue document (if applicable)

### 5.5 Guidelines for Unit Testing, Integration Testing, and System Testing.

- Unit testing is the testing the parts of the program. In unit testing each test case is independent from the others. Using a request form, software tester prepares 'Unit Test Cases' These test cases are approved by TL/PL. Taking this test case as base tester test the application. While testing the functionality if any defects have been found, they are recorded using defect log. The programmer fixes the bugs found and the same is test again. The more emphasis is to be given on functionality.
- In integration testing individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.
- System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) and/or a System Requirement Specification (SRS). Different type of testing that are considered during system testing are compatibility testing, usability testing, security testing, performance testing, regression testing.

### 5.6 Guidelines for Testing Effort estimation.

- **Capers Jones basic formula:**

Number of Test cases = [Number of Function Points] x 1.2

Total Actual Effort, TAE = (Number of Test cases) \* (Percentage of development effort /100)

**This method is done in a case when** a detailed low level design document or requirement document is available (i.e measure of function point is available) & Previous data for development and testing is available.

- **Experience Based:**

1) You already tested similar application in previous project.

2) Inputs are taken from subject Matter experts who know the application (as well as testing) very well.

- **Work Breakdown Structure** - It is created by breaking down the project into small tasks. For e.g. Modules are divided into sub-modules. Sub modules are further divided into functionalities and functionalities are divided in sub-functionalities.

Review all the requirements from Requirement Document to make sure they are added in WBS.

Now you can figure out the number of tasks your team needs to complete. Estimate the duration of each task.

## 6 Checklists / Standards

- Test plan review checklist.
- Test case review checklist
- Release checklist

## 7 Associated Templates

- Test Plan
- Test Case
- Defect log
- Review note

## 8 Control Mechanism

None

## 9 Verification

Item	Activity	Who
Test Plan	Review	PM
Test case	Review	PM

## 10 Tailoring

Refer 'qms/projects/BASQMS001/Working/Releases/<QMS latest version>/Org\_proj\_db/Tailoring Guidelines.'

### 11 Exit

#### CRITERIA

- Work Product is ready to delivered to end User
- All the test cases are executed and all defects are closed.

#### OUTPUT

- Test Plan
- Baseline Test Case
- Tested Product or Component's
- Updated Defect Log

### 12 References

None

### 13 Glossary

qms/projects/BASQMS001/Working/Releases/<latest> version>/Org\_proj\_db/Glossary/'QMS Glossary Ver 1.0.xls'

Controlled Copy