

Test Approach By:

Team Bugvengers

* **Problem:**
* **Date**: 25th May 2022
* **Document Status**: Done
* **Author**: Prashant, Jaspreet, Rahul and Azim
* **Document Number**: Bugvengers v2.0

[CONFIDENTIAL]

Table of Contents

[1. Introduction 3](#_Toc103790748)

[1.1. Purpose 3](#_Toc103790749)

[2. Testing Targets 3](#_Toc103790750)

[2.1. In Scope 3](#_Toc103790751)

[2.2. Out of Scope 4](#_Toc103790752)

[3. Testing Approach 4](#_Toc103790753)

[3.1. Testing Phases and Activities 4](#_Toc103790754)

[3.1.1. Overview 4](#_Toc103790755)

[3.2. Testing Types 5](#_Toc103790756)

[3.2.1. Risk Based Testing 5](#_Toc103790757)

[3.2.2. Regression Testing 5](#_Toc103790758)

[3.2.3. Smoke / Sanity Testing 5](#_Toc103790759)

[3.2.4. GUI Testing 5](#_Toc103790760)

[3.2.5. Performance Testing 5](#_Toc103790761)

[3.2.6. Usability Testing 5](#_Toc103790762)

[3.2.7. Internationalization and Localization Testing 5](#_Toc103790763)

[3.2.8. Static Testing 5](#_Toc103790764)

[4. Test environments 6](#_Toc103790765)

[5. Defect Management 6](#_Toc103790766)

[5.1. Defect life cycle 6](#_Toc103790767)

[5.1.1. Defect reporting 6](#_Toc103790768)

[5.2. Defects Monitoring and Controlling 7](#_Toc103790769)

[6. Testing KPI’s and Testing deliverables 7](#_Toc103790770)

[6.1. Test Strategy 7](#_Toc103790771)

[6.2. Test Plan 7](#_Toc103790772)

[6.3. Test Cases 7](#_Toc103790773)

[6.4. Traceability Matrix 7](#_Toc103790774)

[6.5. Test Report 7](#_Toc103790775)

[7. Testing assumptions and risks 8](#_Toc103790776)

# 

# Introduction

The approach of creating this document for ABC Bank is to write high level testing strategy

Different types of testing to cater the problem statement

Showing our testing KPI’s if we are using scrum model of delivery

## Purpose

This document is produced for use by ABC Banks IT team and assumes proper understanding of the project scope and testing deliverables

## Reviewers and their responsibility:

|  |  |  |
| --- | --- | --- |
| Name | Role Responsibility | |
| Prashant Chauhan | Team Member | Automation & Test Management |
| Jaspreet Kaur | Team Member | Test Automation & Documentation |
| Rahul Mathur | Team Member | Performance Testing |
| Mohd. Azim | Team Member | API Testing |
|  |  |  |

## Customer Company

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature |
| ABC Bank | Customer |  |
| ABC Bank IT Team | Support Team |  |

# 

# Testing Targets

Provide a high-level overview of the main functionalities and business requirements that will have to be covered during testing. Keep the focus on both functional, and non-functional requirements.

## In Scope

E2E ~ Functional and Non-Functional Testing

|  |  |
| --- | --- |
| Entry Criteria | • Requirements to be verified are approved |
| • An integration builds of the application that implements new/changed requirements and/or bug fixes passes unit tests and static analysis and has code review issues resolved |
| • Specification of the target deployment environment (hard-ware, software – OS, databases, browsers, cooperating systems) is approved |
| • The environment that facilitates the tests is set up |
| Exit Criteria | • Tests that cover all new/changed requirements pass |
| • Regression tests pass |
| • All defects are addressed: re-tested and closed or postponed in a defect tracking system |
| • Test reports are available and reviewed |
| Regression Test Approach | At the Design, Development and Continuous Integration stage, a set of regression tests is executed in a regular way following an integration build, e.g., every night, as defined in Test Plan |
| At the Acceptance Testing stage: |
| • A broader set of functional test cases that cover all possibly impacted areas is selected via analysis of new and changed requirements. This regression test set is executed at the Acceptance Testing stage |
| • All tests that cover new and changed requirements are re-executed |
| • Performance tests are executed |
| Test Automation Approach | • A subset of functional tests should be automated for regression testing run after an integration build |
| • Additional test cases could be automated for data-driven testing |
| • Test tools usage for performance tests |
| • Test tools usage for testing web services |
| Test Methods | Black-box test methods are used, such as boundary values tests, positive/negative tests, state-based testing, operational profiles, etc. |
| Accountable | Quality Assurance Lead |
| Responsible | Testers |

## Out of Scope

* No explicit request or approval from the client side
* Features in scope only for future versions
* Commercial off-the-shelf components or products
* Platforms limitation

# Testing Approach

Test approach will be based on tests prioritization and derived mainly from the requirements risk assessment, considering the risk of failure, importance, internationalization and impact.   
Behaviour-driven development often stems from TDD and ATDD. In behaviour-driven development, the purpose of development needs to be tied to a business outcome. So, you’ll have a user story but the user story needs to answer why (in business terms) this feature is being developed. And in BDD, tests are included in user stories as scenarios or specifications.

BDD is also used for acceptance tests. It verifies that the product functions are necessary for the desired business outcome.  
History of the product and its functionality would be helpful for the testing consideration.

Test cases / scenarios should be available on time, should be relevant while testing. Performance testing is also a major part of testing technique performed in order to determine the system behavior regarding time, throughput or volume under a specific workload. Usability testing also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage like Visibility of system status, User control and freedom, Consistency and standards and Error prevention.   
Location specific things can be checked: just for the different languages in application.

Test execution timeframe should be short and time-boxed

## Testing Phases and Activities

Testing should be continuing based on new developments and test activity phases:

* Create new layout, look & feel of the website
* Make website more user friendly and enhance user experience
* Redevelop/redesign all front-end web pages of website

### Overview

The testing process applied on project will consist of the following main phases and activities:

* Test Initiation
* Requirements Analysis
* Test Planning
* Test Analysis and Design
* Test Execution
* Test Reporting
* Test Closure

## Testing Types

### Risk Based Testing

Identify, analyze and evaluate risks related to the software requirements under test (requirements risk assessment)

All software requirements are sorted based on the identified levels of risks

Testing scope might be limited to include only requirements with a certain level of risk

Test design and execution for requirements under test is prioritized, starting the test coverage from the highest risk software requirements

### Regression Testing

The regression test scenarios are created from existing functional/non-functional tests that were agreed and already have been executed prior to this period, History of the product and its functionality is available. Historical test cases / scenarios are available

### Smoke / Sanity Testing

It can be performed at any project phase, Tests are usually scripted, should be relevant, avoiding thoroughly or in-depth analysis and Test execution timeframe should be short, time-boxed

### GUI Testing

Design elements like links, colours, fonts, font sizes, fields etc. are displayed as specified

GUI user actions are performed against specifications

### Performance Testing

Performance testing is a non-functional testing technique performed in order to determine the system behavior regarding time, throughput or volume under a specific workload

### Usability Testing

Perform a heuristic evaluation of the interface based on usability principles, like:

* Visibility of system status
* User control and freedom
* Consistency and standards
* Error prevention
* Recognition rather than recall
* Flexibility and efficiency of use
* Aesthetic and minimalist design

### Internationalization and Localization Testing

The specific characters for each language are used for testing

All the differences are known and checked with the culture and language expert

Checking localization resource files could be involved

Location specific things can be checked: install/uninstall process, UI, functionality related to currency, time and date format, legal aspects requirements, documentation, multimedia, support fonts and chars, text rendering

### Static Testing

Review – Typically used to find and eliminate errors or ambiguities in documents such as Requirements, Design, Test Cases, User Manuals, etc. Performed by Business Analysts and Test Engineers.

Static analysis or Code review – The code written by Developers is analyzed for structural defects that may lead to functional defects. Performed by Developers, manually or with specialized tools.

# Test environments

The environments strategy identified four areas of testing environments.

* **Development environment** – This is the environment where we can sit with developers and fix the issues asap (Unrestricted)
* **Test environment –** All the QA environments where specific testers are doing the testing (Unrestricted)
* **Staging environment –** This is pre-production environment, and this is the most stable environment from all the environment's used within the service delivered (Have some restriction)
* **Production environment –** Represents the live environment. The installation/operation/performance qualifications are conducted here after the deployment or update of the product (Restricted)

# Defect Management

All relevant information about the defects during its life cycle is tracked and stored using the defect management tool.

## Defect life cycle

At a high level look the defect life cycle will include the following phases:

* Defect reporting
* Defect evaluation
* Defect fixing
* Defect validation
* Defect closing

### Defect reporting

Some of the fields are required while creating a defect:

* Issue Type
* Summary
* Saverity
* Priority
* Assignee
* Reporter
* Attachments
* Environment
* Description
* Status

## Defects Monitoring and Controlling

Project custom metrics can be created using the project’s test management tool. These metrics are good indicators of the quality of the product under test. Based on these, the testing priorities can change, new testing activities can be planned.

# Testing KPI’s and Testing deliverables

## Test Strategy

The Test Strategy document is used by the test team to guide how the testing will be managed for the project.

## Test Plan

The Test Plan document is used by the test team to guide a testing cycle execution. It describes the timeline, requirements or test cases in scope to be executed, test environment related activities, roles and responsibilities.

The Test Plan can be created in digital format, using an appropriate Test Case Management tool. In this case, all relevant information that might not be included – like test environments, roles and responsibilities – will be stated in the Test Strategy, accordingly.

## Test Cases

The test case is a set of preconditions, steps, test data and expected results, developed for a given functionality in order to verify its compliance with a requirement or to mitigate a risk.

They are defined using the appropriate methods, reviewed by a different person than the author, maintained during project life cycle, run and re-run if needed with results recorded, and stored.

## Traceability Matrix

The Traceability Matrix is used to trace and correlate the business or other requirements to their implementation, testing or completion. Usually it is a two-dimensional table, which correlates two entities (e.g., requirements and test cases).

The table allows tracing back and forth the links of one entity to the other, thus enabling the determination of coverage achieved and the assessment of impact of proposed changes. At any time, this should provide the project requirements status in terms of their level of completion.

## Test Report

The Test Report summarizes, for a specific time-period or a given testing cycle, the results of testing, including testing team related details, the defect status, a list of deviations from the test plan, test metrics, quality recommendations. All Test Report details are agreed with the stakeholders at the beginning of the project.

The Test Report can be created in digital format, by using a Test Case Management tool, and exported if needed in the agreed format. All relevant information that might not be included – like defect status, quality recommendation – will be stated in other deliverables, accordingly.

# Metrics

Software testing metrics are a way to measure and monitor test activities. If you measure the correct metrics in a right way and transparently, they will guide you to understand the team progress towards certain goals and show the team’s successes and deficiencies.

The whole team approach also critical on the metrics that you will measure and report, so it is very important to popper introduce metrics.

Test execution reporting should contain below information:

* Overall number of test cases (by status)
* Test execution trend
* Number of defects and defect severity distribution
* Defect resolution time
* Defect open vs defect closed

For more complex projects more sophisticated metrics for reporting might be created.

* Percentage of escaped defects
* Percentage of rejected defects
* Percentage of duplicate defects
* Critical/high severity defects index

# Testing assumptions and risks

* Lack of skills
* Lack of resources
* Services failed from third party
* Shift in project major milestones
* Communication issue