

Test Approach By:

Team Bugvengers

* **Problem: Mobile Game Native App Testing**
* **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](#_Toc104397370)

[1.1. Purpose 3](#_Toc104397371)

[1.2. Roles & Responsibilities: 3](#_Toc104397372)

[1.3. Customer Company 3](#_Toc104397373)

[2. Testing Targets 4](#_Toc104397374)

[2.1. In Scope 4](#_Toc104397375)

[3. Testing Approach 5](#_Toc104397376)

[3.1. Testing Phases and Activities 5](#_Toc104397377)

[3.1.1. Overview 5](#_Toc104397378)

[3.2. Testing Types 6](#_Toc104397379)

[3.2.1. Functional Testing 6](#_Toc104397380)

[3.2.2. Interruption Testing 6](#_Toc104397381)

[3.2.3. Smoke / Sanity Testing 6](#_Toc104397382)

[3.2.4. Memory Leak Testing 6](#_Toc104397383)

[3.2.5. Performance Testing 6](#_Toc104397384)

[3.2.6. Usability Testing 7](#_Toc104397385)

[3.2.7. Localization Testing 7](#_Toc104397386)

[3.2.8. Security Testing 7](#_Toc104397387)

[4. Test environments 7](#_Toc104397388)

[5. Defect Management 8](#_Toc104397389)

[5.1. Defect life cycle 8](#_Toc104397390)

[5.1.1. Defect reporting 8](#_Toc104397391)

[5.2. Defects Monitoring and Controlling 8](#_Toc104397392)

[6. Testing KPI’s and Testing deliverables 8](#_Toc104397393)

[6.1. Test Strategy 8](#_Toc104397394)

[6.2. Test Plan 8](#_Toc104397395)

[6.3. Test Cases 9](#_Toc104397396)

[6.4. Traceability Matrix 9](#_Toc104397397)

[6.5. Test Report 9](#_Toc104397398)

[7. Metrics 9](#_Toc104397399)

[7.1. App Metrices 10](#_Toc104397400)

[8. Testing assumptions and risks 10](#_Toc104397401)

# 

# Introduction

Creating Test Approach for Mobile Gaming company called ABC is planning to release a new game in the market. The game is expected to be a Native mobile app and is expected to support both Android and iOS. There is no corresponding web component (website) for the game, it is supposed to be only a native app.

From device support point of view, the game is supposed to support major iOS and Android devices. ABC is also expecting the performance of the native app be good and it should work fine for user load of close to 1000 concurrent users.

## Purpose

A high-level testing approach is presented in this document. This document outlines how native applications will be tested.  
In this strategy document, we would also like to include our KPIs.

## Roles & Responsibilities:

|  |  |  |
| --- | --- | --- |
| Name | Role Responsibility | |
| Prashant Chauhan | Team Member | Approval for QA test activities and resource allocation |
| Jaspreet Kaur | Team Member | Automation Testing |
| Rahul Mathur | Team Member | Performance Testing |
| Mohd. Azim | Team Member | Manual Testing |
|  |  |  |

## Customer Company

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature |
| ABC Mobile Gaming Company | Customer |  |
|  |  |  |

# 

# 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, 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 |

# 

# Testing Approach

* All stories will be analysed by QA team before starting Test creation tasks.
* Test case creation will be aligned within sprints which includes functional testing.
* Test execution will start once all Sub-tasks related to development are closed and story is moved to QA and unit tested by the development team.
* QA team will again review the story before starting Test Execution and update the test cases if there is any change in Story from the time when test cases were created.
* QA will utilize automation tool to test the extensive native application functionality which includes numerous workflows and permutations of the modifiers.
* QA will utilize browser stack and devices to regression test of each device OS listed above.
* QA Team will follow standard Severity process for Defects.
* Defect will be reported outside the story and linked to each defect will be assigned to Product Owner for prioritization by QA team.
* For every Test cycle such as Regression, Sanity and Smoke completion test status report will be shared across the entire ABC Game development team.
* Daily stand-up report will be provided to the team.
* QA Status Report will be provided to a broader team members/management
* Defect review, Test case review, and the Retrospective meeting should align within the team in every Sprint.
* All Critical/Blocker tickets must be QA complete for the story ticket to closed.
* Test estimation efforts will be provided in Sprint Planning/Estimation.
* All failed test cases in a Sprint will be re-executed in next Sprint once the related to the associated test case is resolved.

## Testing Phases and Activities

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

* Create new layout, look & feel test cases of the Native Mobile App
* Make application more user friendly and enhance user experience by proper testing
* Testing on all front-end application pages of the user perspective

### 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

### Functional Testing

This mobile testing type aims to verify that every function of an app is working exactly as required. It focuses on testing user flows within the app, as well as each feature that comprises each user flow.

* App installs and launches correctly
* Users can sign-up and log into the app easily
* Text boxes, buttons, menus, and icons function accurately
* [Push notifications](https://www.browserstack.com/guide/test-push-notification-on-android-devices) render correctly and appear at the right intervals
* Any transactions or purchases should happen seamlessly

### Interruption Testing

* Incoming or phone call when an application is running
* Incoming message or SMS when an application is running
* Low battery when an application is running
* The device is plugged in or out of charging when an application is running
* Device shutting down when an application is running
* OS upgrade occurring when an application is running
* Loss and restoration of the network while an application is running

### 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

### Memory Leak Testing

A memory leak refers to a situation in which the app fails to return the memory it has acquired for temporary use in order to function. The available memory for the app drains, and the app cannot function. If an app is frequently used or opened, a small memory leak can result in its termination. Memory leaks emerge from programming bugs, so every app needs to be tested for this issue. Memory leak testing is done by running an app on multiple devices.

### Performance Testing

* **Device performance**: Installation and log-in time, battery consumption, memory consumption, etc.
* **Network performance**: Delays, errors, pauses in receiving digital information or rendering network-activated features
* **API/Server performance**: Speed and formation of data transfer from back-end to front-end
* **Recovery capabilities**: Built-in back-up and recovery functions that can save or recover user data in the event of data loss.

### 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

### Localization Testing

[Localization testing](https://www.browserstack.com/guide/localization-testing-on-websites-and-apps" \o "Localization testing" \t "_blank) checks these features to ensure that they are activated and functioning in the right locations. Customers always prefer apps with UI elements aligned with their culture, language, and device accessibility. They expect their experience to be adjusted to their localized needs and preferences.

### Security Testing

App users are becoming increasingly conscious of issues surrounding data security. Online privacy and confidentiality of personal data are major concerns for most netizens – [70%](https://techjury.net/stats-about/app-usage/#gref) report being concerned that their personal information will be shared without permission. In fact, [81% of users](https://techjury.net/stats-about/app-usage/#gref) say they would uninstall an app and switch vendors because of security concerns.

# 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

## App Metrices

* Number of downloads
* Active users
* Session intervals
* Average usage time
* Churn Rates
* Your ranking in the Apple App Store and Google Play Store
* Abandonment rates
* Conversion rates
* Number of crashes
* Speed
* Latency
* Retention rates
* User feedback
* App loading per period

# Testing assumptions and risks

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