**Test Strategy Document**

ABC Games

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Reviewer(s)** | **Description** |
| May 25th, 2022 | 0.1 | Lets Get QAed | Lets Get QAed Team | Final Draft |

Table of Contents

Contents

[1. Introduction 4](#_Toc104393856)

[2. Objective 4](#_Toc104393857)

[3. Scope of Testing 4](#_Toc104393858)

[4. Test Approach 4](#_Toc104393859)

[5. KPIs 7](#_Toc104393860)

[5.1 Automation KPIs 7](#_Toc104393861)

[5.2 Manual Testing KPIs 8](#_Toc104393862)

# Introduction

The purpose of this document is to describe in detail the overall test strategy to be used for “ABC Gaming Service Provider”. It describes the scope of test automation, testing approach and test methodology, develop, execute, and maintain the automated test scripts which would reduce the overall testing time of ABC Games Mobile application and ultimately would quicken the launch of the new game.

# Objective

Principle objectives of testing of ABC Games mobile application journeys which would improve test accuracy, enable faster launch and enhance overall team productivity and product quality. Also, this document covers the test approach to be implemented.

# Scope of Testing

* Planned scope :
  + Functional Testing
  + Integration Testing
  + GUI Testing
  + API Testing
  + Automation Testing
  + Performance Testing(memory leaks, network throttling, device hardware factors)
  + Mobile Testing
  + Device Compatibility Testing
  + Security Testing
  + Compliance Testing(Service Regions)
  + Cloud Testing
  + Big Data Testing
  + Usability Testing(vol)
  + Interoperability Testing
  + Regression Testing (Automated and Manual)
  + HA/DR Testing
* Physical testing will be done only on 2 versions of the OS (latest and the one below the latest one)
* Device farms like (AWS/Browserstack/Kobiton, etc.) will be used for testing different resolutions and OS versions other than the 2 mentioned in the above step.

# Test Approach

Following are the testing types that will help us expedite the launch process and also release a high-quality game into the market.

**FUNCTIONAL/SYSTEM TESTING:**

Testing of each module of the app. Validating whether correct data is sent and received from the API Gateway. Checking if all the pages as per the wire-frames take in desired information and same is stored and retrieved at the server end. Validating if the game rules are as per the requirements.

**INTEGRATION TESTING:**

Focus on seamless integration between different modules and making sure the complete functionality behaves as one system. Data transactions with other modules/screens/apps/apis will be tested.

Testing end-to-end through the application and APIs, including positive and negative tests

**FUNCTIONAL AUTOMATION TESTING:**

The objective is to develop an automation testing framework using which the mobile and APIs can be tested. This framework will be extensible so that as new features are developed or new scenarios are identified or products changes, we will be able to extend or change it as required.

A hybrid framework which incorporates the best practices of Test Library, Test Modularity, Data Driven, and Descriptive Programming approaches will be developed which will be used with Rest-Assured and Appium to automate the test scenarios defined in the regression test pack for both UI and API testing. This framework mainly focuses on re-usability of automation code and automated test scripts which reduces the maintenance effort. This translates to a significant savings over the course of time. The automation suite will be integrated with Jenkins/AzureDevOps pipelines as soon build deployed regression suite will be executed with report notified to stakeholders.

Automation Framework will consist of different modules handling Android and iOS natives apps.

**REGRESSION TESTING:**

Regression testing is essential for ensuring software quality of the existing features of the game app. Nagarro will take Risk Based Testing approach for new game that focuses on test cases that are identified as business critical and whose failure had high impact on the system.

**GUI TESTING:**

Validating the game is designed as per the given requirements and wireframes. Testing the look and feel of the game with different mobile device resolutions like mobiles, tablets, ipads, etc.

**SECURITY TESTING:**

* It is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended.
* The security testing is performed to check whether there is any information leakage in the sense by encrypting the application or using wide range of software’s and hardware’s and firewall etc.
* Software security is about making software behave in the presence of a malicious attack.

**Diagram

Description automatically generated with medium confidence**

* The OWASP Top 10 Web Security Risks Testing
* The WASC Threat Classification – Below attacks and weaknesses
* DAST aspects of Mobile Application.

**PERFORMANCE TESTING:**

* Load, Stress and Endurance Testing with min 1000-2000 users concurrently in the game.
* Mobile App profiling for testing application performance on any particular device and resource consumption.
  + Memory profiling, Detecting Leaks
  + Detecting increasing memory issues
  + Inspecting CPU activity with CPU profiler.
  + Inspecting network usage with Network profiler.
  + App launch time.
  + Inspect energy use with Energy Profiler
* Performance baselining, benchmarking and optimization as per NFRs provided by Kotak.
* Continuous Performance monitoring and API response time tracking in each phase of testing as a part of API Testing.

**COMPLIANCE TESTING:**

For mobile game launch Define Your Target Audience and countries to launch. Demographics where to launch and based on that check and test the rules for data privacy and GDPR, etc.

**BIG DATA TESTING:**

* Testing out different advertising strategies.
* Analyzing performance of the application
* getting insight from real time data to make future improvements.
* For real time user data testing the response from app, fallback due to interruptions, or all the data capture the performance of game while playing and how to improve the functionality this can happen only when we know the data and where to improve in app and game.

**INTEROPERABILITY TESTING:**

Testing whether the game still keeps running or gets paused when we switch to other apps or when we get a call.

**DEVICE COMPATIBILITY TESTING:**

Testing whether the game behaves exactly as expected in different OS devices or resolutions or device hardware configurations.

**CLOUD TESTING:**

Testing cloud data storage and cloud functions (if used). Big data testing will need cloud tools for validation and verification of data.

**HA/DR TESTING:**

Testing high availability and Disaster recovery as the game can be played at any time in the day and the user

# KPIs

#### Automation KPIs

* **ROI**

ROI is the effort saved by after implementation of test automation in the project as compared to the effort involved when manually performing the same activity.

Formula for calculation: *ROI = (Manual Effort – Automation Effort)/ Manual Effort \* 100*

Planned calculation frequency: *Monthly*

* **Automation Coverage**

Automation coverage is the percentage of tests till date from the overall planned automation scope.

Formula for calculation: *(Automated Tests / Total Tests) \* 100*

Planned calculation frequency: *Weekly*

* **Automation Progress**

Automation progress is a graphical representation of the work completed versus time. The completed work is on the vertical axis vs time on the horizontal axis.

* **Number of bugs found**

Number of regression bugs found by every automation run

* **Number of Production leak**

Number of production leaks or bugs by automation

* **Number of false negatives**

Number of invalid bugs found by automation

#### Manual Testing KPIs

The frequency could be measured at Sprint level in a Scrum or at Release level in iterative model.

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics Name | Nagarro Standards | Frequency | Definitions |
| Defect Leakage | <= 4 | Release | Client reported defects per 100 days of estimated effort for the unit of work under consideration. |
| Sprint Defect Leakage | <= 12 | Sprint | Client reported defects per 100 story points for the User Stories executed in the Sprint. |
| Defect Density | <= 80 | Release | Total defects detected internally and externally per 100 days of estimated effort for the unit of work under consideration. |
| Internal Defect Density | <= 240 | Sprint | Total defects detected internally per 100 story points for the User Stories executed in the Sprint |
| Defect Removal Efficiency | >= 95% | Release | Number of defects detected through internal reviews and testing as a percentage of total number of defects detected in the unit of work under consideration. |
| Defect Rejection Rate | <=12 | Sprint | (No. of defects rejected/ total no. of defects raised) X 100 |
| Review Effectiveness | >= 30% | Release | Number of defects detected through internal reviews as a percentage of total number of defects detected in the unit of work under consideration. |
| Sprint Completion Ratio | >= 90% | Sprint | Extent of the Sprint Backlog that was completed at the end of a sprint as a percentage of the scope of the sprint that was committed at the start of the sprint. |
| Sprint Stability | >= 90% | Sprint | A measure of the stability of the sprint based on the number of changes that occur during the course of the sprint. |
| Velocity |  | Sprint | A measure of productivity of a Scrum team based on the no. of story points completed in a sprint. |
| Effort Variance | + / - 5% | Sprint | Variance between the actual and estimated effort with respect to estimated effort. |
| Rework | <= 15% | Sprint | Time spent in fixing defects from internal and external reviews and testing as a percentage of total time spent in coding. |
| Capacity Utilization |  | Sprint | Work planned for a Scrum team as a ratio of their available capacity based on the no. of resources, their availability and the duration of the sprint. |
| Test Case execution ratio |  | Sprint | No. of test cases executed / Time spent in test execution |
| Test Case Preparation Productivity |  | Sprint | (No of Test Case)/ (Effort spent for Test Case Preparation) |
| Test Case Efficiency |  | Sprint | (Number of defects detected / Number of test cases run)\* 100 |
| Resource Ageing Ratio |  | Program | Average age of the team members in the project as a ratio of the Project age. |
| Resource Ramp-up Time | <= 4 weeks | Program | Maximum time to transition knowledge to a new team member who is replacing an outgoing team member. |