

Table of Contents

1. INTRODUCTION

- 1.1. Purpose
- 1.2. Project Overview
- 1.3. Audience

2. TEST STRATEGY

- 2.1. Test Objectives
- 2.2. Test Assumptions
- 2.3. Test Principles
- 2.4. Data Approach
- 2.5. Scope and Levels of Testing
 - 2.5.1. Exploratory
 - 2.5.2. Functional Test
 - 2.5.3. User Acceptance Test (UAT)
- 2.6. Test Effort Estimate

3. EXECUTION STRATEGY

- 3.1. Entry and Exit Criteria
- 3.2. Test Cycles
- 3.3. Validation and Defect Management
- 3.4. Test Metrics
- 3.5. Defect Tracking and Reporting

4. TEST MANAGEMENT PROCESS

- 4.1. Test Management Tool
- 4.2. Test Design Process
- 4.3. Test Execution Process
- 4.4. Test Risks and Mitigation Factors
- 4.5. Communications Plan and Team Roster
- 4.6. Role Expectations
 - 4.6.1. Project Management
 - 4.6.2. Test Planning (Test Lead)
 - 4.6.3. Test Team
 - 4.6.4. Test Lead
 - 4.6.5. Development Team

5. TEST ENVIRONMENT

1. INTRODUCTION

1.1.Purpose

This test plan describes the testing approach and overall framework that will drive the performance testing of the Santander – OnBoarding project. The document introduces:

- **Test Strategy:** rules the test will be based on, including the givens of the project (e.g.:start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- **Execution Strategy:** describes how the tests will be performed and process to identify and report defects, and to fix an implement fixes.
- **Test Management:** process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster).

1.2.Project Overview

OnBoarding Project is a set of APIs and resources which have the objective of providing new users a way of access to the on-boarding process into the company, checking their identity via NID verification, credit information, and validating user data as emails or phones.

1.3. Audience

- Project team members perform the specified tasks of this document, and provide input and recommendations on this document.
- Project Manager plans the testing activities in the overall project schedule, reviews the document, track the performance of the test according to the specified tasks, approves the document and is accountable for the results.
- The stakeholders representatives and participants may take part in the UAT test to ensure the business is aligned with the results of the performance tests.
- Technical team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

2. TEST STRATEGY

2.1. Test Objectives

The test **objective** is to verify that the performance of Santander – OnBoarding APIs and resources works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, and prioritize lower severity defects for future fixing.

The final product of the test is twofold:

- A performance production-ready software.
- A set of stable test scripts that can be reused for Performance test execution.
- An integrated launcher project for automated execution of test scripts in order to obtain scheduled HTML reports.

2.2. Test Assumptions

Key Assumptions

- Production like data required and be available in the system prior to starting of performance testing.
- Infrastructure is assumed to be adapted for the system that is going to be tested.

General

- Exploratory testing would be carried out once the build is ready for testing.
- Functional testing is not considered for this estimation.
- All the defects would come along with a snapshot JPEG format.
- The Test Team will be provided with access to the Test environment via VPN connectivity.
- The Test Team assumes all necessary inputs required during test design and execution will be supported by Development/BUSINESS ANALYSTs appropriately.
- Test case design activities will be performed by QA Team.
- Dev Team will be in charge of test environment and preparation activities.
- Dev team will provide defect fix plans based on the defect meetings. The same will be informed to QA Team prior to start of defect fix.
- BUSINESS ANALYST will review and sign-off all test cases prepared by QA Team prior to start of test execution.
- The defects will be tracked through JIRA only. Any defect fixes planned will be shared with QA Team prior to applying the fixes on the test environment.
- Project Manager/BUSINESS ANALYST will review and sign-off all test deliverables.
- The project will provide test planning, test design and test execution support.

- QA Team will manage the testing effort with close coordination with Project PM/BUSINESS ANALYST
- Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
- There is no environment downtime during test due to outages or defect fixes.
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

Performance Testing

- During performance testing, QA Team will have at its disposal the necessary hardware, with the appropriate scalability settings, at the execution time.
- The QA Team will perform Performance testing only on Santander – OnBoarding project.

2.3.Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Pre-production environment will emulate the production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4.Data Approach

In performance testing, Santander – OnBoarding will contain pre-loaded test data and which is used for testing activities, like DNI images or associated documents.

2.5.Scope and Levels of Testing

2.5.1. Exploratory

PURPOSE: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

SCOPE: First level performance testing for each APIs and resource.

TESTERS: QA Team.

METHOD: this exploratory testing is carried out in the applications without any test scripts and documentation.

TIMING: at the beginning of each cycle.

2.5.2. Funcional Test

PURPOSE: This test plan is not about functional testing, but some minor functional testing will be necessary when obtaining the executed test scripts. These functional tests will check the application functionalities, and is carried out by feeding the input and validates the performance metrics/statistics from the application.

2.5.3. Performance Test

PURPOSE: Performance testing will check the proper infrastructure behavior and autoscaling. The performance testing is carried out by feeding the input and validates the performance metrics/statistics from the application.

Scope: The below table details about the scope of Performance (and Funcional) test. The scope is high level due to changes in the requirement.

API	RESOURCE	Pre-loaded data
CBC	Entity Report	-
CBC	Physical Report	-
CCE	Risk Score	-
CSO	List	-
CSO	Pep	-
IDP	Document Type	-
IDP	Process Complete	Documents
ID&V	Documents Type	-
ID&V	Identity Type	-
ID&V	Verifications	Images
NID	Validation	-
STA	Locations	-
STA	Places	-
STA	Postcode	-
STA	Timezone	-

2.5.4. Stress Tests

This test type is about progressive increase of concurrent accesses in order to discover break points into the infrastructure. The below table details about the specifications of the stress tests for each API and resource:

Concurrent Users	Steps	Expected Results
Progressive and unlimited	1º. Set unlimited users. 2º. Send OK request. 4º. Validate success rate.	1º. Total of users executed. 2º. Expected requests limit achieve.

2.5.5. Load Tests

This test type is about executing short duration scenarios with different number of concurrent users, checking the behavior of the microservices, the functionality and the latency of the request in concrete and specific scenarios. The below table details about the specifications of the load tests for each API and resource:

Concurrent Users	Steps	Expected Results
100	1º. Define concurrent users. 2º. Send OK request. 3º. Validate response data. 4º. Validate success rate. 5º. Validate latency. 6º. Extract performance metrics and statistics.	1º. 200 OK received.
500		2º. Correct JSON structure received.
1.000		3º. Expected success rate achieved.
5.000		4º. Expected latency achieved
10.000		
50.000		
100.000		
250.000		

2.5.6. Pike Tests

This test type is about of checking the microservices and infrastructure behavior in case of abnormal usage, with high concurrency scenarios. The test scripts will execute requests from a low concurrency level to a high concurrency level, and vice versa. The below table details about the specifications of the pike tests for each API and resource:

Concurrent Users	Steps	Expected Results
1 → 10.000 → 1	1º. Define concurrent users. 2º. Send OK request. 4º. Validate success rate. 5º. Validate latency and timeout rate.	1º. Total of users executed.
1 → 100.000 → 1		3º. Expected success rate achieved.
1 → 250.000 → 1		4º. Expected latency and timeout rate achieved.

2.5.7. Stability Tests

This test type is about of checking the infrastructure behavior when a stable number of users is executing requests during a long period of time. The below table details about the specifications of the stability tests for each API and resource:

Concurrent Users	Steps	Expected Results
100	1º. Define concurrent users. 2º. Establish time period. 3º. Send OK requests. 4º. Validate success rate. 5º. Validate latency. 6º. Extract performance metrics and statistics.	1º. Total of users executed.
500		3º. Expected success rate achieved.
1.000		
5.000		4º. Expected latency achieved.
10.000		
50.000		
100.000		
250.000		

2.6. Test Effort Estimate

3. EXECUTION STRATEGY

3.1. Entry and Exit Criteria

- The **entry criteria** refer to the desirable conditions in order to start test execution.
- The **exit criteria** are the desirable conditions that need to be met in order proceed with the implementation.
- Entry and exit criteria are flexible benchmarks. If they are not met, the QA Team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final “go-no go” decision.
- Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
- Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

Exit Criteria	QA Team	Technical Team	Notes
100% Test scripts executed.			
95% pass rate of test scripts.			
No open critical and high severity defects.			
95% of medium severity defects have been closed.			
All remaining defects are either cancelled or documented as change requests for a future release			
All expected and actual results are captured and documented with the test script.			
All test metrics collected based on automated reports.			
All defects logged on Zephyr.			
Test Closure Memory completed and signed off.			
Test Environment cleanup completed and a new backup of the environment.			

3.2. Test Cycles

- There will be two cycles for performance testing. Each cycle will execute all the scripts.
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
- The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.

3.3.Validation and Defect Management

It is expected that the QA Team execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.

The defects will be tracked through JIRA. The Dev Team will gather information on a daily basis from the automated reports, and request additional details from the Defect Coordinator. The Dev Team will work on fixes.

It is the responsibility of QA Team to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the Dev Team the fix and its implementation, communicate with QA Team when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the Dev Team to review automated reports on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

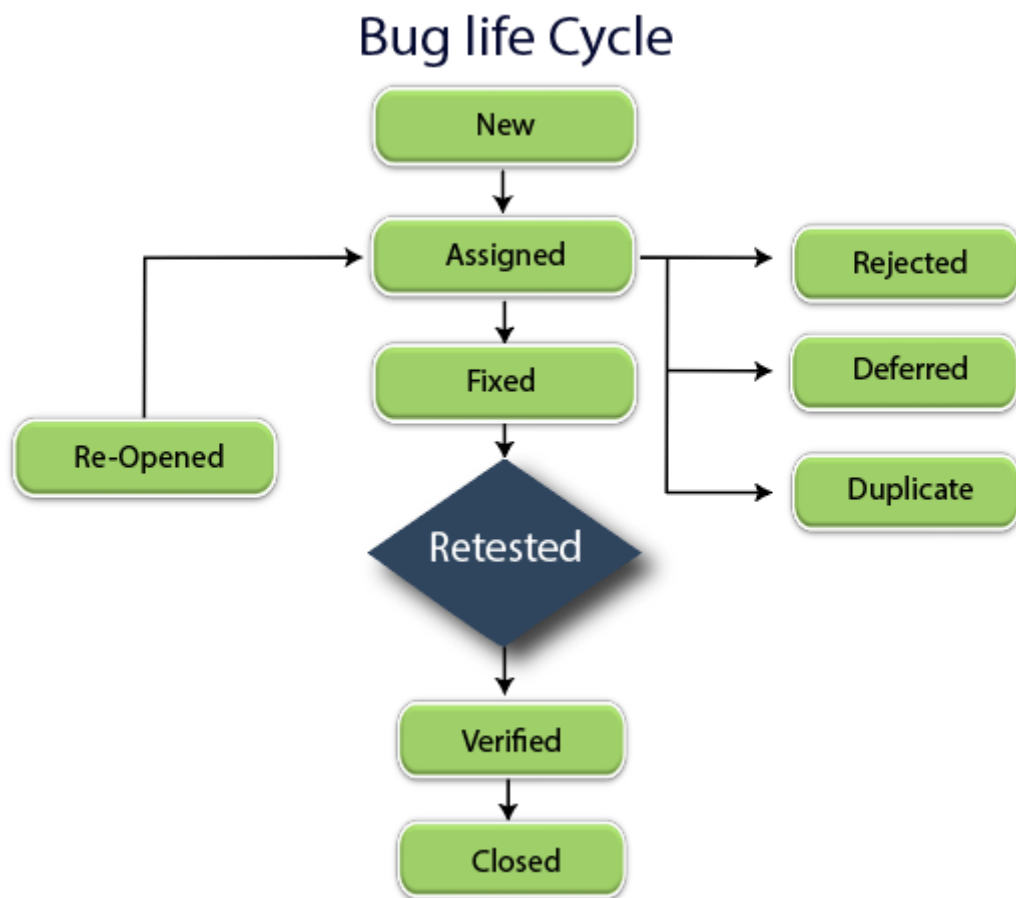
Defects found during the testing will be categorized according to the following categories:

Severity	Impact
Critical	Defect enough to crash the system, file corruption, potential data loss or it causes a infrastructure crash.
High	It causes problems on vital application functionalities.
Medium	Degrades the quality of the system or prevent other areas of the product to be tested.
Low	Minimun impact in product use.

3.4.Test Metrics

3.5. Defect Tracking and Reporting

Following flowchart depicts Defect Tracking Process:



4. TEST MANAGEMENT PROCESS

4.1. Test Management Tool

Zephyr JIRA add-on is the tool used for test management. All testing artifacts such as test cases, test results are updated in the **Zephyr** tool.

- Project specific folder structure will be created in **Confluence**.
- Each resource in the QA Team will be provided with read/write access to add/modify test cases in Zephyr.
- During the test design phase, all test cases are written directly into Zephyr. Any change to the test case will be directly updated in Zephyr.
- Each tester will directly access their respective assigned test cases and update the status of each executed step in Zephyr directly.
- Any defect encountered will be raised in JIRA linking to the particular test case/test step.
- During defect fix testing, defects are re-assigned back to the tester to verify the defect fix.
- The tester verifies the defect fix and updates the status directly in JIRA.
- Various reports can be generated from the automated and integrated launcher project to provide status of test execution. For example, status report of performance test cases executed, passed, failed, performance metrics/statistics, etc.

4.2. Test Design Process

The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered. Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the QA Team. The testers will rework on the review defects and finally obtain approval and sign-off.

During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements. In order to check the infrastructure behavior, after designing the functional phase of the test case, tester will implement the validations of the corresponding KPIs.

Sign-off for the test cases would be communicated through mail by BUSINESS ANALYST. Any subsequent changes to the test case if any will be directly updated in Zephyr.

4.3. Test Execution Process

Once all test cases are approved and the test environment is ready for testing, tester will start an exploratory test of the application to ensure the application is stable for testing. Each tester will have assigned test cases directly in Zephyr.

QA Team will use Zephyr and JIRA for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the project manager as escalation. If any **showstopper** is detected during exploratory testing, it will be escalated to the Dev Team for fixes.

Each tester performs step by step execution and updates the executions status. The tester enters **pass** or **fail** status for each test or step directly in Zephyr. QA Team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.

If any defect is detected, it will be raised as per severity guidelines to JIRA tool detailing steps to simulate along with screenshots if appropriate. Daily test execution status, provided by the integrated and automated reporting, as well as defect status will be reported to all stakeholders.

If there are any defects that are not part of steps but could be outside the test steps, such defects, it will be necessary to capture them in JIRA and map them against the test case level or at the specific step that issue was encountered after confirming with Test Lead.

This process is repeated until all test cases are executed fully with pass/fail status. During the subsequent cycle, any defects fixed applied will be tested and results will be updated in JIRA during the cycle.

4.4. Test Risks and Mitigation Factors

4.5. Role Expectations

The following list defines the involved roles of the management, planning or execution of the test for the project.

Roles	Name	Contact Info
Project Manager	Pablo Mateo Fernández	Email: pablo.mateofernandez@gruposantander.com Phone: +34 695 290 216
Business Analyst	Susana López Morón	Email: susana.lopezmoron@gruposantander.com Phone: +34 635 161 433
Test Leader	Sergio Fernández Vicente	Email: sergio.fernandezvicente@servexternos.gruposantander.com
Development Leader	Jose Luis Cobo Rincón	Email: joselu.cobo@servexternos.gruposantander.com
QA Team	Sergio fernández Vicente	Email: sergio.fernandezvicente@servexternos.gruposantander.com
	Daniel Torres Candil	Email: daniel.torrescandil@servexternos.gruposantander.com
DEV Team	Jose Luis Cobo Rincón	Email: joselu.cobo@servexternos.gruposantander.com
	Rubén Arroyo Rodríguez	Email: ruben.arroyo@servexternos.gruposantander.com

4.5.1. Project Management

4.5.2. Test Planning (Test Lead)

4.5.3. Test Team

4.5.4. Test Lead

4.5.5. Development Team

5. TEST ENVIRONMENT