Volpis Test Strategy

1. **Purpose**

The purpose of this Test Strategy is to define the overall approach that will be taken by the Test Team when delivering testing services to all of the projects within the business.

The document helps to clarify the testing activities, roles and responsibilities, processes and practices to be used across successive projects.

Where a project's testing needs deviate from what this Test Strategy covers, the exceptions will be detailed in the Test Plan.

2. Approach

All testing tasks will be conducted in line with the Software Test Life Cycle (STLC) and in support of the Software Development Life Cycle (SDLC). The documents used within the SDLC will be completed by the Test Team and the project participants responsible for providing information and deliverables to the Test Team.

3. Testing process flow

Static testing (Design, Documentation) -> “Defined scope” (feature) testing -> Integration testing -> “Ad-hoc” testing -> Non-Functional Testing (Security, Performance, Localization, Compatibility) -> Regression testing -> UAT testing -> Release testing

4. **Testing Types**

- **Static testing** is a software testing method that examines a program along with any associated documents but does not require the program to be executed. The purpose of static testing is to reduce cost in the early stages of development based on the amount of rework needed to fix any errors
- **"Defined scope" (feature) testing** is the process of making changes to a software system to add one or more new features or to make modifications to existing features. The purpose is to make sure that your software meets the business requirements and that all major functions work properly
- **Integration testing** is defined as a type of testing where software modules are integrated logically and tested as a group. The purpose of integration testing is to expose faults in the interaction between integrated units
- **“Ad-hoc“ testing** means performing random testing without any plan. The main purpose of “Ad-hoc“ testing is to identify any errors that would usually go unnoticed during formal testing methods
- **Security testing** is a type of software testing that focuses on identifying vulnerabilities, weaknesses, and potential risks in a system's security mechanisms. The primary goal of security testing is to ensure that the software or application being tested can withstand potential security threats and protect sensitive data.
- **Performance testing** is a type of software testing that evaluates the performance and responsiveness of a system under various workload conditions. It involves measuring and assessing how well a software application performs in terms of speed, scalability, stability, and resource usage.
- **Compatibility testing** is a type of software testing that verifies whether a system or application can function correctly and efficiently across different environments, configurations, platforms, and devices. It ensures that the software remains compatible with various operating systems, web browsers, hardware devices, and software dependencies.
- **Regression testing** means to run series of tests every time new code is submitted. The purpose of regression testing is to confirm that a recent program or code change has not adversely affected existing features

5. Testing Tools

Testing tools are software applications or frameworks that assist in automating, managing, and enhancing the testing process. These tools are designed to improve efficiency, accuracy, and effectiveness in various aspects of software testing

We use such **testing tools** as:

- TestRail
- Postman
- Xcode
- Android studio
- Charles
- OWASP Zap
- Jmeter
- Proxyman

6. Team actions

Before starting a project, there are several important actions that a team should take to ensure a smooth and successful project execution. Here are some key steps:

- **The Team Roles and Responsibilities:** Assemble a team with the necessary skills and expertise to accomplish the project goals. Consider each team member's availability, roles, and responsibilities. Ensure there is clear communication about expectations and commitments.
- **The environments:** Defining the environments before a project starts is essential for smooth execution and to ensure that the team has the necessary infrastructure and resources in place.
- **Device matrix:** Defining the device matrix before a project starts involves determining the devices, OS versions and platforms that will be supported or targeted for the project

7. Defining risks

Defining project risks involves identifying and documenting potential events or circumstances that could have a negative impact on the successful completion of a project. It is an essential step in project management to proactively identify and address potential obstacles and uncertainties.

This is a **Risk Template** table:

Cause	Description	Required action	Impact

- The **Cause** is a column in which we define the possible action that can cause a risk
- In the **Description** column we put a detailed description of a risk
- In the **Required action** column is the action that we should take to solve the issue
- The **Impact** column is to specify the impact on the deadline if the risk occurs

8. Test approach

- **Analysis & Planning Phase Entry Criteria**

For all projects the following criteria need to be met before the Test Items are accepted into the Analysis & Planning Phase:

- □ The project scope item list is defined and completed
- □ Documentation defining the scope items is approved

- ☐ All changes in the documents are estimated and approved by BA, PM, and customer
- ☐ Sanity and Unit tests are written, reviewed, and approved by PM, and customer (if needed on the project) (not valid for now)

- **Analysis & Planning Phase Exit Criteria**

For the Analysis & Planning phase to be completed and allow items to move into the Test Phase the following criteria need to be achieved:

- ☐ Test Cases are written, reviewed, and approved by Team Lead
- ☐ Walkthrough and sign-off completed for the Test documents
- ☐ Defined Test Estimate has been published and agreed by PM
- ☐ Positive test scenarios are written, prioritized and approved by BA, Team Lead
- ☐ Sanity and Unit tests are completed (if needed on the project) (not valid for now)

- **Test Phase Entry Criteria**

Before Test Items are made available for the Test Team to test it's expected that:

- ☐ All test tools are available and test infrastructure, and test environments are available for testing
- ☐ All Test Items are development complete (needs to be specified)
- ☐ Sanity and Unit tests have been completed successfully to demonstrate readiness for test (if needed on the project) (not valid for now)

- **Test Phase Exit Criteria**

For the Test Items to exit testing the following conditions will have to be met:

- ☐ The *Test Summary Report* is completed (add hyperlink to the Test Summary Report template)
- ☐ All planned testing activities has been completed
- ☐ All high priority bugs have been fixed, retested and passed
- ☐ No defects are left in an open unresolved status

9. The "Change Request" Management

The Build Manager will ensure that once testing begins no changes or modifications are made to the code used to create the build of the product under test. The Build Manager will inform the Test Team against which version testing will begin and confirm the location within the build is to be taken from.

If changes or modifications are necessary through bug resolution or for any other reason the Build Manager will inform the Test Team prior to the changes being made.

10. Suspension Criteria & Resumption Requirements

Testing of Test Items will be suspended if:

	Suspension criteria	**Resumption requirement**
1	A Severity 1 issue is logged and requires fixing before further testing can take place (a Blocking Issue)	The issue will need to be fixed before the Test Item is returned to the Test Team for testing
2	Significant differences exist between observed behaviour of the Test Item and that shown in Test Scenario, Test Case or as expected from the previous version of the technology	Development, the Test Team and PM must come to a conclusion on resolving the issue and agreeing a definition of the expected behaviour

3	A Test Item sent for testing fails more than 20% of Developer Unit Tests • (if needed on the project) (not valid for now)	The Test Item must be fixed or Unit Tests refactored if out of date and then demonstrated to pass with <20% failure

11. Test Deliverables

The following artifacts will be produced during the testing phase:

- **Test Plan**

Used to prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

- **Test Breakdown**

Which includes the Test Scenarios, their priority and related number of Test Cases along with the defined estimates for time to write and execute the Test Cases.

- **Test Cases**

Detail the pre-conditions, test steps and expected and actual outcome of the tests. There will be positive and negative test cases.

- **Periodic progress and metric update reports**
- **Bug Reporting**
- **Test report**
- **Known issues report** is a document that provides a consolidated list of identified issues or bugs in a software application, system, or product. It serves as a reference for stakeholders to be aware of the known problems and their status.

Name	iOS	Android
Test Runs URL		
Jira issues all (filter)		
Known issues	N/A	URL