Merc Test Plan

VERSION 2

1. Introduction

Product Vision

Empower and enable VIs to do their jobs more efficiently and effectively

Business Goals

- Build single platform for VIs
- Better UX (modern)
- Scale to new business opportunities easier (VRS -> VRS + VRI)
- Quick and easy set up

Project Goals

- Increase customer satisfaction (compliments/complains):
  - Shared text
  - Return to 2.25 teaming percentage (0.8% now)
    - Improve UX to decrease amount of clicks
    - Company benefits because VI is more efficient -> company makes $

2. Roles and Responsibilities

Position	Team member
Project Manager	Yuliia T.
Developer Team Lead	Vitaliy B.
Developer	Kyrylo Z.
Developer	Vitalii K.
Developer	Dmytro P.
Developer	Varduhi V.
Developer	Oleksandr C.
General QA Engineer	Oleksii H.
General QA Engineer	Bogdan S.
System Engineer	Arsen Y.

3. Testing levels and types

Testing Levels

A Test Level is a group of test activities that are organized and managed together. A Test Level is linked to the responsibilities in a project or maintenance.

The Test Levels are:

- Integration Testing
- System Testing
  - Functional
    - Smoke
    - Regression
  - Non-functional
    - Specification review
    - Design review
    - Cross-device
    - Performance
    - Security
    - Usability
    - Localisation
- Automation Testing
- Acceptance Testing

Unit Testing

Unit tests are automated tests that test individual software components. Tests are subject to code review along with the actual implementation to ensure that the tests are applicable and useful.

Created, run, and executed by developers at the coding phase and after each code change.

Integration Testing

Testing performed to:

- Expose defects in the interfaces and in the interactions between integrated units or systems;
- Testing performed to expose defects in the interfaces and interaction between software components;
- Testing the integration of systems and packages

System Testing

Testing an integrated system to verify that it meets specified requirements.

System testing is concerned with the behaviour of the whole system. Its objective is to ensure that the system is correctly behaving as defined in the requirements.

The following Test Types should be performed:

- Functional testing: Testing based on an analysis of the specification of the functionality of a component or solution (testing "what" the solution does)
  - Smoke testing: Testing of a subset of all defined or planned test cases that cover the main functionality of a component or system in order to ensure that the most crucial functions of a solution work (there is no need to test in detail). It is recommended to automate the smoke tests as much as possible.
  - Regression Testing: Testing of a previously tested solution, the following modification to ensure that defects have not occurred or are uncovered in unchanged areas of the solution. It is performed when the software or its environment is changed.The regression test consists of previously defined and used test cases.The size of the regression test set is based on the product risk of the solution.
- Non-functional testing: Testing of the attributes of a component or system that do not relate to functionality (testing "how well" the solution works). Non-functional testing includes but is not limited to: Deployment, Performance, Security, Privacy, Usability, Maintainability, Reliability, Portability, Installability testing, etc.
  - User stories, specification review
  - Design, Prototype review
  - Cross-device Testing
    - Cross-device testing is done to ensure that the solution behaves correctly on different devices or when using different browsers on the same device
    - Performance Testing
      - Load (Internal tools)
      - Stress (Internal tools)
    - Security Testing
      - Header middleware etc.
      - OWASP Top 10 Vulnerabilities.
      - Clickjacking tool.

Automation testing

Auto-tests will be created during feature testing at every sprint (with some delay depending on manual testing time) and will be updated due to product and business logic changes (both website and admin panel).

Automation testing will be provided for the regression testing after each code change and when any task delivers to the QA department for testing.

Automation testing Framework:

- TypeScript + PlayWright
- GitActions

- Playw right reporter

Type	Name
Brow ser	Chrome
Applications	nTouch
	Zoom
	Teams

4. Testing levels and schedule

Test level	Test type	Environment	Schedule	Responsible
Unit Tests	Unit	Dev Stage Preproduction	Automated. At the coding phase and after each code change.	Developers
Integration Tests	Integration	Dev Stage Preproduction Production	Automated.	Developers
System Tests	Smoke	Production	Manual after deploy	General QA
	Functional feature Testing	Stage Preproduction	Manual after each module/feature is completed	General QA
			Automation (schedule or manual launch if needed)	General QA
	Regression	Preproduction	Before release	General QA
	Specification review		At early development stage, after US is confirmed by BA and customer	General QA
	Cross-device	Production	Before release	General QA
Performance		Preproduction	Before release	General QA
Security		Preproduction	Before release	Developers
Usability		Preproduction	Before release	General QA

5. Risk & Issues

1. **The loss access to the environment**
   The problem occurred because of changes in account settings or changing environment preferences.
   How to solve: You have to write a message to "X" slack channel and tag @Steve
2. **PingOne account password expiration**
   The problem occurred if you didn't changed password in time, so you can't login to Merc with PingOne
   How to solve:
   Open your PingOne account and change your password before it expired If it already expired you should write email to ML@so.com
3. **Testing on a same account**
   This problem occurred when you and someone from your team use same test account for testing. In this case occurs critical issues and blockers because Mercury thing that 1 user use 2 or more queues.
   How to solve:
   When you use test accounts make sure that noone else using this account while you testing
   If you observe new critical issues doubleckeck that noone else using your test account while you testing
4. **Endpoint breakage**
   The problem occurs on nTouch, WebRTC, SignTime. QA and developers need all endpoints working to proceed properly testing and developing.
   How to solve:
   Make sure that you enter hold servers data correctly in Property table and URL
   Generator Make sure that you are using correct link/hold servers for each environment
   Make sure that problem is on endpoint side
   If you are sure that endpoint broke write a message to "X" slack channel and tag @Steve

6. Test logistics

Env.	Testing Conducted	Artifacts Tested	Performed By	Schedule	Deploy Procedure
Dev	Unit Tests Integration Tests Smoke feature testing	Source Code	Developers	CI/CD deployment tool	GitHub actions
	Requirements testing	Merc	Manual QA Engineers	Requirements tests before sprint	Manual job run

	Local test Functional test Smoke test E-2-E Autotest Regression test Localisation testing			Local test runs when feature passed code review Functional testing runs after each feature is complete and deployed to env Smoke test runs 1 time per week Autotest runs while regression Regression test runs before deploy Localisation testing runs when new localisation passed code review	TypeScript + PlayWright
QA	Unit Tests Integration Tests Smoke feature testing	Source Code	Developers	CI/CD deployment tool	GitHub actions
	Smoke test E-2-E Autotest Regression test Performance testing	Mer	Manual QA Engineers	Smoke test runs 1 time per 2 weeks Autotest runs while regression Regression test runs before release	Manual job run TypeScript + PlayWright Jenkins -release branch
VRI NA, VRS NA	Hot fix	Source code	Developers	After release	Developers manual job + DevOps (if needed)

		Smoke test	Mercury	Manual QA Engineers	After release	Manual job run Playwrite framework

7. Test criteria – entry and exit criteria

Entry Criteria

Gives the prerequisite items that must be completed before testing can begin. It can vary from project to project, but here are some of the examples that can be generally used to mark the beginning of the testing:

Unit tests

- Code is finished
- 100% unit test coverage with reviews
- Code complexity and analysis are done

Integration tests

- Completion of the unit testing phase
- Priority bugs found during unit testing have been fixed and closed
- Integration plan and test environment to carry out integration testing is ready
- Each module has gone through unit testing before the integration process

System testing

- Unit testing is finished
- Complete or partially testable code is available
- Requirements are defined and approved
- Availability of sufficient and desired test data
- Test cases are developed and ready to execute
- Priority bugs found during previous testing activities have been fixed and closed
- System testing environment is available

Acceptance:

- Successful completion of system testing phase
- Priority bugs found during previous testing activities have been fixed and closed
- Functional and Business requirement has been met
- Acceptance testing environment is ready
- Test cases are available

Exit Criteria

Can be defined for each test level or for the general testing process, right from test planning, specification, and till execution. The exit criteria for terminating the process of testing are:

Unit tests:

- Successful execution of the unit tests
- All the identified bugs have been fixed and closed
- Project code is complete

Integration tests:

- Successful execution of the integration tests
- Satisfactory execution of stress, performance, and load tests
- Priority bugs have been fixed and closed

System tests:

- Successful execution of the system tests
- All specified business and functional requirements have been met
- Priority bugs have been fixed and closed
- System's compatibility with supported hardware and software

Acceptance:

- Successful execution of the user acceptance tests
- Approval from management to stop UAT
- Business requirements got fulfilled
- No critical defects have been left out

Bugs priority types

- Blocker – Defect causes a critical loss of business functionality or a complete loss of service has occurred
- Highest– Defect causes a major impact to business functionality and there is not an interim workaround available
- High – Defect causes a major impact to business functionality
- Medium– Defect causes minor impact to business functionality and there is an interim workaround available
- Low – Defect causes minor impact to business functionality
- Lowest – Defect is cosmetic only and usability is not impacted

8. Testing Environment

The list of things that contributes to the environment set up for the purpose of testing:

- Software product, on which testing needs to be performed
- Sorenson and PingOne account
- Software:
  - nTouch
  - SignTime
  - Conference API collection (Postman)
  - Property Table Editor
  - URL Generator
- Testing environments
  - Old LZ Dev
  - New LZ Dev
  - New LZ QA

9. Test Deliverables

Expand

- Test Cases
- Bug Reports
- Test Run Report (Metrics from TestRail)
- Automation Tests with Playwright and TS
- Load Testing report