Testing Strategy

- Scope of Testing
    - In Scope
    - Out Of Scope
- Testing Requirements
- Acceptance Criteria
- Testing Types
- Testing Types - Implementation Details
- Test Phases
    - 1. Test Design and Test Implementation
    - 2. In-Iteration Testing
        - 2.1 Individual Module Testing
    - 2.2 Regression Testing
    - 2.3 Performance testing
        - The applicability of Web, Mobile App and performance testing
    - 2.4 Mobile App Sanity Testing
    - 2.5 Sanity Testing on the Integrations Environment
    - 3. UAT
- Test Automation
- Test Environments

Scope of Testing

The goal of current project is the iterative update of the Spring Boot version for the application modules and libraries.

In Scope

Application modules are in scope of testing.

Out Of Scope

Application libraries to be updated are out of scope. User interface testing of the modules (Web and Mobile app versions) is out of scope and is performed by separate team.

Testing Requirements

The main requirement for testing is to ensure that Spring Boot upgrades not lead to the functional issues and performance degradation.

Acceptance Criteria

No special acceptance criteria are specified for the project. Modules can be released from testing if the appropriate testing activities (individual module testing, regression testing and performance testing) are passed successfully.

Testing Types

- exploratory testing;
- functional manual testing for each module;
- regression testing;
- sanity testing;
- integration testing;
- performance testing.

Performance testing is not applicable for all modules (please see Applicability table).

Testing Types - Implementation Details

Exploratory testing should perform on the first stage of the project. The goal of exploratory testing is to get familiar with the functionality and to implement test cases for functional manual testing for each module (both Web and Mobile App versions).

Since manual functional testing is organized mainly as end-to-end testing (back-end is tested through UI) the Use Cases testing is used as main test design technique. Test cases should be reviewed and approved by customer.

Regression tests list contains almost all test cases used for individual module testing.

Sanity testing for UAT can be consider as smoke testing, test cases cover only main functionality. Test cases should be reviewed and approved by customer.

Integration and performance tests are automated and provided by customer.

Test Phases

The following phases are applicable for the project:

1. Test design and test implementation.
2. In-Iteration testing.
3. UAT.

1. Test Design and Test Implementation

Unlike the ones below this phase is not a part of each release activities. This phase can be presented at the beginning of the project and at some iterations (where test cases should be updated to cover new features released by customer).

At the beginning of the project the test cases for each module and sanity test cases should be created. In order to understand system and prepare test cases the exploratory testing should be performed.

Customer intensively works on WFM features and has a releases once a month. In case when critical feature is added new test cases should be added to cover new feature. So, this phase and corresponding testing activities should be added to iteration.

Test cases can be found here.

2. In-Iteration Testing

In-Iteration testing contains Individual module testing, regression testing, sanity testing and performance testing activities.

2.1 Individual Module Testing

Individual module testing is performed during first three weeks of the iteration by test team. Testing is performed on development branch. Testing activities include manual functional testing for each module (Web version only) is performed.

To track testing activities ticket for module implementation is used, no separate ticket should be created.

Testing activity can be start when:

- implementation of the particular module is finished;
- unit tests and automated integration tests are performed be Dev team, all tests are passed;
- implemented module code is deployed on test environment;
- tests environment contains actual version of front-end for module (version currently deployed on Prod).

If criteria above are met ticket is moved to testing (the status of ticket is changed to 'In Progress-3').

Testing activities can be finished when:

- all test cases are passed;
- no issues (except of known) exist.

If criteria above are met ticket is moved to 'acceptance' (the status of ticket is changed to 'In Progress-4') and assigned to BA.

2.2 Regression Testing

Regression testing is performed at last week of the sprint by test team. Testing activities include manual regression testing for Web version only for modules to be released.

To track testing activities separate ticket should be created for the release. Ticket should be linked to the appropriate tickets for modules to be released.

Testing activity can be start when:

- testing for all individual modules going to be released is finished and appropriate tickets are passed to customer;

- for modules to be released changes made during the sprint are synced with the latest master (as a latest master the master branch with related customer's release changes is considered);
- test environment is updated with the latest master for all modules, not only modules to be released.

If criteria above are met ticket is moved to testing (the status of ticket is changed to 'In Progress-3').

Testing activities can be finished when:

- tests are passed;
- no issues preventing go decision exist.

If criteria above are met ticket is moved to 'acceptance' (the status of ticket is changed to 'In Progress-4') and assigned to BA.

2.3 Performance testing

Performance testing is performed at last week of the sprint by Dev team. Testing activities include automated performance testing.

Testing activity can be start when:

- testing for individual modules going to be deployed is passed and appropriate tickets are accepted and closed by customer;
- for modules to be released changes made during the sprint are synced with the latest master (as a latest master the master branch with related customer's release changes is considered);
- test environment is updated with the latest master for all modules, not only modules to be released.

Testing activities can be finished when:

- performance tests (where applicable) are passed.

The applicability of Web, Mobile App and performance testing

Module	Web	Mobile App	Performance
Module 1	YES	YES	YES
Module 2	YES	YES	YES
Module 3	YES	YES	NO
Module 4	YES	NO	YES
Module 5	YES	NO	NO

2.4 Mobile App Sanity Testing

Mobile app sanity testing is performed at last week of the iteration by test team.

To track testing activities separate tickets should be created for each module to be released for which mobile app testing is applicable. Ticket should be linked to the appropriate ticket for modules to be released.

Testing activity can be start when:

- testing for all individual modules going to be released is finished and appropriate tickets are passed to customer;
- for modules to be released changes made during the sprint are synced with the latest master (as a latest master the master branch with related customer's release changes is considered);
- test environment is updated with the latest master for all modules, not only modules to be released.

If criteria above are met ticket is moved to testing (the status of ticket is changed to 'In Progress-3').

Testing activities can be finished when:

- tests are passed;
- no issues preventing go decision exist.

If criteria above are met ticket is moved to 'acceptance' (the status of ticket is changed to 'In Progress-4') and assigned to BA.

2.5 Sanity Testing on the Integrations Environment

Sanity testing on the Integrations environment is performed after regression testing and mobile app sanity testing are finished (normally one day before the release date). Testing is performed by test team.

To track testing activities separate ticket should be created for the release.

Testing activity can be start when:

- regression testing is finished for all modules to be released;
- for modules to be released release branch is merged into master branch;
- Integrations environment is updated with the latest master of modules to be released.

If criteria above are met ticket is moved to testing (the status of ticket is changed to 'In Progress-3').

Testing activities can be finished when:

- all test cases are passed.

If criteria above are met ticket is moved to 'acceptance' (the status of ticket is changed to 'In Progress-4') and assigned to BA.

3. UAT

UAT is performed after sanity testing on Integrations environment is finished. UAT is performed by test team using.

During this phase sanity testing is performed.

To track testing activities separate ticket should be created for the release.

If criteria above are met ticket is moved to testing (the status of ticket is changed to 'In Progress-3').

Testing activity can be start when:

- testing on Integrations environment is finished;
- UAT environment is updated with the latest master of modules to be released.

Testing activities can be finished when:

- all test cases are passed.

If criteria above are met ticket is moved to 'acceptance' (the status of ticket is changed to 'In Progress-4') and assigned to BA.

Test activities schedule for team releases can be found here.

Test Automation

Unit, integration and performance tests are automated. Tests are provided by customer.

Unit and integration tests are executed by Dev team before functionality is moved to QA. Performance tests are executed at the regression week - last week of the iteration.

Test Environments

The following environments are set up on the project:

- Dev environment;
- Test environment;
- Integrations environment;
- UAT environment.

Dev environment is used by Dev team, but in some cases can be used for testing (in case testing environment is not available). Deployment is performed by our team when it's needed; no notification of the customer's team is needed.

Testing environment is used by QA for manual testing of modules and regression testing (Web version). Deployment is performed by our team when it's needed; no notification of customer's team is needed.

Mobile App regression testing is performed on separate environment shared with customer's team. Deployment can performed by our team when it's needed; notification of customer's team is needed.

Integrations environment is used for short sanity testing of the modules to be released. Deployment is performed by our team when it's needed; no notification of customer's team is needed.

UAT environment is used for short sanity testing of the modules to be released. Release artefact are deployed by customer's DevOps engineer based on the release schedule.

Information on the environments can be found here.