Test Plan

Responsible team behind document: Validation Team

Important users of this document (Recipient): Validation Team

Short content summary: In this document, the Validation Team has specified its process routines, the main tools and methods used and, as specifically as possible, a time plan.

Version & Date: 1.0 2019-10-17
Inspected by Quality Coordinator: Emmelie Jeppsson
Date: 2019-10-18
Inspected by recipient: Isabelle Johansson
Date: 2019-10-18

Version & Date: 2.0 2019-12-09
Inspected by Quality Coordinator: Emmelie Jeppsson
Date: 2019-12-09

Author(s): Test Leader: Eric Jonsson, Tester: Jonatan Harling

Revision History

Version	Date	Author(s)	Reviewed by	Changes added
1.0	2019-10-14	Eric Jonsson	Emmelie Jeppsson, Isabelle Johansson	Creation of document.
2.0	2019-12-09	Eric Jonsson, Jonatan Harling	Emmelie Jeppsson	Restructure of the document, removing “Test Routines” and added that content to “Types of Testing Implemented” **Changed:** 5 **Added:** 6.2

1. Test goals

The purpose of dedicating a part of the company to testing is mainly to ensure a high quality of the final product. The validation team will throughout the development of the product examine whether the requirements set by the analyst team and signed by the customer are fulfilled or not. In order to ensure high quality, it is also necessary to fulfill the true needs of the customer, which sometimes includes not only verifying requirements but also validating the experience of the product.

2. Validation Team

The Validation Team consists of a test leader, two testers and a quality coordinator. The test leader is leading the group and is responsible for the tests that will be created and executed. The two testers assist the test leader in creating and running tests. The quality coordinator is responsible for evaluating the current quality of the product.

3. Tools

3.1 Selenium

Throughout the developing of the product, Selenium will be used, which is a suite of tools made for web-based applications. Specifically, the Selenium WebDriver run in Java is used. Selenium WebDriver is a tool to communicate with the browser and in that way being able to control it in order to test the requirements. It is a DOM-based tool, where elements can be identified by, for example, its ID. Selenium WebDriver makes it possible to, for example, click on buttons, type in fields or get values from labels.

3.2 TestNG

As Selenium is lacking the ability to create reports representing test results, TestNG is used as a complementary tool to provide that.

3.3 Travis CI

Travis CI is a continuous integration tool used to create tests for software projects hosted at GitHub. As it supports continuous integration, updates of the product can automatically be tested.

4. Types of testing implemented

To ensure a good quality of the product, various types of test is being implemented. Starting from testing the smallest units in the *unit testing*, followed by testing the units working together as modules in the *integration testing*, then the modules together working as a whole system in the *system testing* and finally letting company members as well as end-users test the product in the *acceptance testing*.

4.1 Automated Unit testing

Testing of the simplest units, i.e. methods, programs, etc., is done by the development team through *Travis CI.* This means that whenever an update of the product is made and pushed to the repository at GitHub, a test is automatically run. This is done for the most essential functions of the web application

4.2 Integration testing

Integration testing is where several units are combined and tested as a group. The integration testing will be done by the validation team during the so-called *Test Sessions* and is a process done throughout the whole development of the product. *Black-box testing,* i.e. testing modules (groups of units) without inspecting the code, will be implemented. The integration tests are based on the SRS requirements and are therefore a good indicator of the product's current status.

During the Test Sessions, most of the functional and a few of the non-functional requirements are tested with programs created with the Selenium WebDriver API. All these tests can be run simultaneously and the results are presented automatically with TestNG. The tests run automatically but have to be triggered manually. For the requirements not suitable for Selenium tests, manual tests are implemented and run.

The test results are then manually added in the *Requirements Traceability Matrix.* When the result of a test is *Failed*, a defect is also added to the RTM. In cases of the Selenium tests, the defect contains the output from the test result and a more descriptive explanation. The defects are also marked with a severity going from *Low* to *High.* How this is determined can be found in the document Process Routines. All defects are then added to the backlog as a GitHub issue.

As the requirements, including the ones that already have passed a test, are tested continuously, Test Sessions implement a type of regression testing.
The main purpose of Integration Testing is to verify that the requirements from the SRS are met.

4.3 System testing

System testing is the next level of testing and is performed to see how the system as a whole is working. System testing is carried out towards the end of the project. It is performed during the last test sessions alongside the integration testing. It works as a complement to integration testing as it doesn't test specific requirements but rather worked as a way to find bugs and issues and to give an impression of the application as a whole. To do this the Validation Team follows a set of instructions similar to the ones followed in the acceptance testing. The idea is that the System Testing is more about *validating* than *verifying.*

After a test session including system testing, the results from the integration testing and issues and thoughts from the system testing are added to a document. This document is then sent to the Development Team. Apart from sending the document, a session to orally present what was found is scheduled with a representative from the Development Team.

4.4 Acceptance testing

Acceptance testing is the last level of testing and is where the product is evaluated to see if it is meeting the requirements and the expectations set up before launching the product. It can be divided into the sub-groups *Internal acceptance testing* and *External acceptance testing.*

4.4.1 Internal acceptance testing (alpha testing)

During the Internal acceptance testing or alpha testing, the product is tested by the company itself. It's important that the tests are done by members not involved in the actual development as too much insight into the code and the product will affect the test. Therefore, in MedCom's case, it is done by members outside the UX, development and validation team.

The idea is to gather a group of around 5 members of the company, to perform a task on the product while having the tested members thinking out loud. Afterward, the members are asked to answer a group of questions. The results of the answers as well as the thoughts thought out loud during the performance are documented and summarized and are presented to the development team.

The purpose of the alpha testing to provide results that can be used to fix bugs and issues or to provide proposals for future work. It can also provide useful input regarding the design and user experience.

4.4.2 External acceptance testing (beta testing)

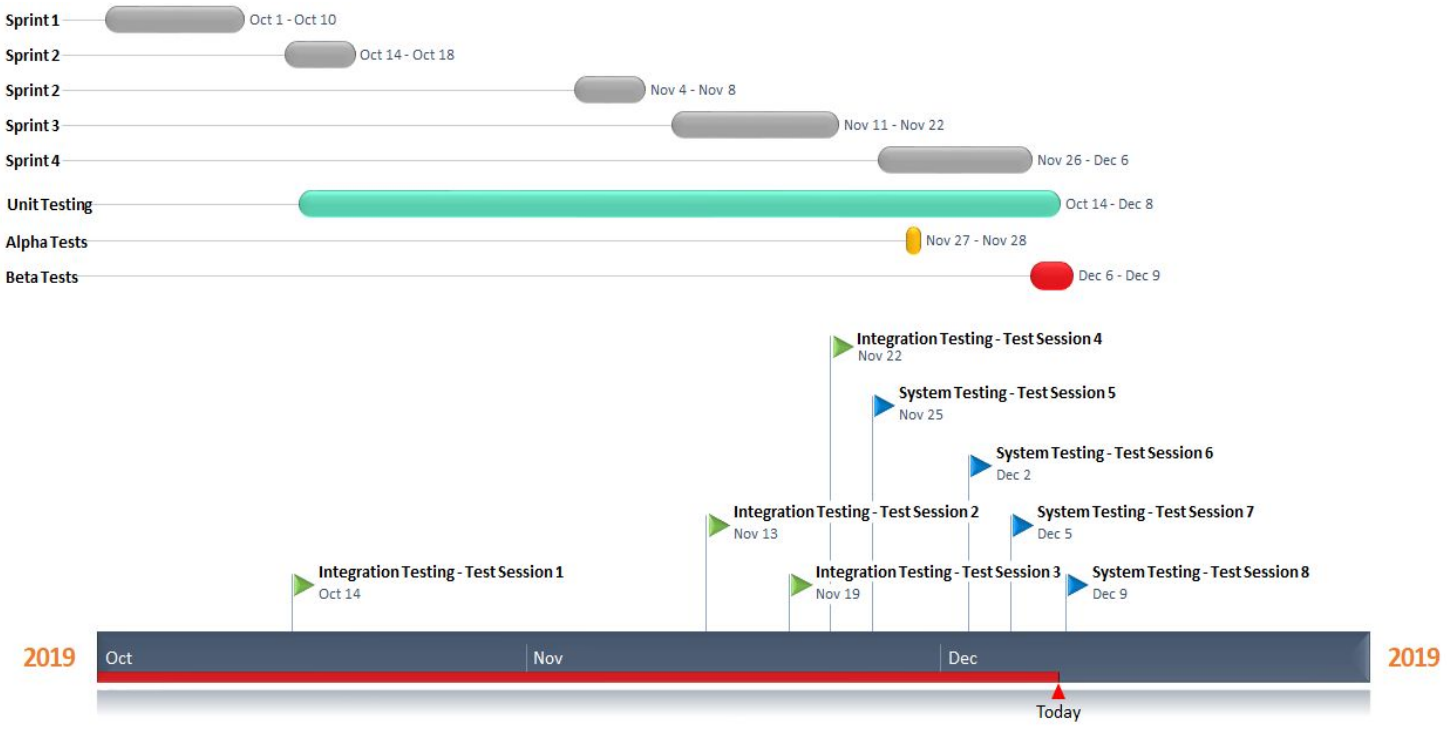
External acceptance testing is when the product is tested by someone outside of the company.

The beta testing consists of gathering end-users to let them test the product. In MedCom's case, these are preferably doctors. As in alpha testing the beta testers will perform a task while thinking out loud and answer questions. They are also asked to rank different aspects of the product, for example in relation to usability. This is done by responding to a SUS-test where the end-users respond to questions by ranking aspects of the product from 1 to 5. The answers can then be used to calculate a score ranging from 1 to 100.

As the beta tests will be performed late in the project, the majority of the results will be stored as proposals for future work. Another purpose is to gather quality metrics that can be used to ensure high quality of the product.

5. Time Plan

To meet the requirement of the SRS (System Requirement Specification), the validation team has developed a plan describing what shall be done and in which order. The time plan (from October and forward) created is as follows:



6. Documentation of the testing

6.1 Requirements Traceability Matrix (RTM)

To keep track of all of the tests, to connect them with the requirement(s) and to provide a comprehensible overall picture of the functionality and quality of the product, a Requirement Traceability Matrix (RTM) is being used. In the RTM, all requirements with its associated test as well as the current status of those tests i.e. *passed* or *failed,* are being presented. In case of a failed test, a defect to describe how or where the test failed, is documented to present to the development team. However, it is important to state that a passed test will not stop being tested. As the validation team is applying *regression testing,* tests must be run anew whenever there is a chance that a part of the product that is modified could affect the test.

Apart from using an RTM for internal reasons, it can also be used externally to present to the customer. The RTM together with the Quality Management Document works as proof of the quality and to ensure that the company has provided the customer with what was agreed upon. If desired, the customer will be provided with the final RTM when delivering the product. In addition to ensuring quality and delivering a product that meets the requirement, providing the customer with the RTM can also facilitate the process of developing the product further in the future.

6.2 Quality Factors

Towards the end of the project various quality factors including understandability, usability and maintainability were produced and documented in order to ensure high quality of the product. The documentation of the quality factors in more detail is found in the Quality Management Document.