**Document Management System**

Test Plan

**Table of contents**

[**Introduction**](#_heading=h.28h4qwu) **4**

[**Scope of Testing**](#_heading=h.nmf14n) **4**

[Preconditions and outer dependencies](#_heading=h.37m2jsg) 5

[Limitations](#_heading=h.1mrcu09) 5

[Suspension criteria and resumption requirements](#_heading=h.46r0co2) 5

[Release/ Acceptance Criteria](#_heading=h.2lwamvv) 6

[**Test Items**](#_heading=h.111kx3o) **6**

[Features to Be Tested](#_heading=h.3l18frh) 6

[Features Not to Be Tested](#_heading=h.206ipza) 8

[**Testing Approach**](#_heading=h.4k668n3) **8**

[Prioritization](#_heading=h.2zbgiuw) 9

[Entry/Exit Criteria](#_heading=h.1egqt2p) 9

[Item Pass/Fail Criteria](#_heading=h.3ygebqi) 11

[Classification of Defects](#_heading=h.2dlolyb) 12

[Resumption and Suspension Criteria](#_heading=h.sqyw64) 13

[**Testing Levels**](#_heading=h.3cqmetx) **13**

[Module Testing](#_heading=h.1rvwp1q) 13

[Test Responsibilities](#_heading=h.4bvk7pj) 14

[Test Methods](#_heading=h.2r0uhxc) 14

[System Testing](#_heading=h.1664s55) 14

[Test Responsibilities](#_heading=h.3q5sasy) 14

[Test Methods](#_heading=h.25b2l0r) 15

[Entry and Exit Criteria](#_heading=h.kgcv8k) 15

[System Integration Testing](#_heading=h.34g0dwd) 16

[Acceptance Testing](#_heading=h.1jlao46) 17

[Test Responsibilities](#_heading=h.43ky6rz) 17

[Other Testing Level Types](#_heading=h.2iq8gzs) 17

[Smoke Testing](#_heading=h.2xcytpi) 17

[Regression Testing](#_heading=h.xvir7l) 17

[PROD Sanity Check](#_heading=h.3hv69ve) 18

[**Test Schedule, Organization and Responsibilities**](#_heading=h.1x0gk37) **18**

[Testing Schedule](#_heading=h.4h042r0) 18

[Test Project Organization & Resources](#_heading=h.2w5ecyt) 18

[Tools and Access Rights Required](#_heading=h.1baon6m) 18

[Training Needs](#_heading=h.3vac5uf) 18

[**Tool for Test Design and Execution**](#_heading=h.2afmg28) **18**

[**Test Environments**](#_heading=h.pkwqa1) **19**

[Development Environment](#_heading=h.39kk8xu) 19

[Actions Required for Development Environment](#_heading=h.1opuj5n) 19

[Test Environment](#_heading=h.48pi1tg) 19

[Actions Required for Test Environment](#_heading=h.2nusc19) 19

[Production Environment](#_heading=h.1302m92) 19

[Actions Required for Production Environment](#_heading=h.3mzq4wv) 19

[**Test Metrics**](#_heading=h.2250f4o) **19**

[Schedule for Test Metrics collection](#_heading=h.haapch) 19

[Metrics Types](#_heading=h.319y80a) 19

[**Risks**](#_heading=h.1gf8i83) **19**

[**Deliverables**](#_heading=h.40ew0vw) **20**

[Test Items](#_heading=h.2fk6b3p) 20

[Test Documents](#_heading=h.upglbi) 20

[**Change history (document)**](#_heading=h.3ep43zb) **20**

**Document Management System (DMS1)**

# Introduction

The purpose of this document is to define the test approach for the project. In this document the project context, test approach, testing levels, test environments, metrics to be collected and initial risks from testing point of view will be defined.

The goal of this document is to explain the test methodology, scope, approach, resources of all testing activities of the project Document Management System.

Proactive testing approach will be applied during this project.

* **Proactive -** An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
* **Reactive -** An approach in which the testing is not started until after design and coding are completed.

The overall strategy of this testing is manual, black box testing in combination with automated testing (regression, smoke and sanity checks).

Testing levels to be used:

1. Module Testing
2. System Integration Testing
3. System Testing
4. Acceptance Testing
5. Other (regression, smoke, sanity check).

Test environment: System Testing will be conducted on the DMS1 Web application.

This document is created for the Project Team, Steering Group, Project Management and any other Reference Groups.

For Project Summary, Background, Scope – Refer to Project Plan

# Scope of Testing

As per agreed software requirement specification, the scope of testing will be as follows:

Functional requirements should be covered 100% with test cases. Test cases should be prioritized.

Before release: regression testing should be done, all bugs registered in JIRA, all planned test cases executed.

System Testing: 70% 1st priority test cases were successful.

User acceptance testing: executed and 100% planned test cases executed(to be clarified)

Final testing report prepared.

Test items to be covered:

Possibility to create, submit, accept or reject documents. Creation of groups, users, document types.

The main test levels to be used are: Module Testing, System Integration Testing, System Testing, Acceptance Testing and other (regression, smoke, sanity check).

Non functional requirements are currently out of scope, as well as DataBase Testing and any other external interfaces.

## Preconditions and outer dependencies

Prerequisites:

1. Requirements are understood
2. Any unclear specifications have been clarify with PO.

External preconditions that must be available at given time in order to be able to start a specific work for testing the project:

1. Test plan created.
2. Test cases created.

## Limitations

Not applicable.

## Suspension criteria and resumption requirements

*If there are* ***40%*** *of test cases failed, you should* ***suspend*** *testing until the development team fixes all the failed cases. Once testing is resumed, test cases should be executed again.*

The test project manager has the right to suspend all or part of the test if any of the following occurs:

* *One “fatal fault”*
* *More than three “serious faults” occurring in the same function. This will be regarded as equivalent to one “fatal fault”*

*Project management will be informed immediately and will determine the appropriate action.*

## Release/ Acceptance Criteria

This document is created for giving recommendations whether the software items should or should not go live by the test manager or other directive.

The below items should be completed in order to approve the release exit criteria:

* Regression testing done, all bugs registered in JIRA.
* All planned test cases are executed.
* System Testing: 70% 1st priority test cases were successful.
* UAT: executed and 100% planned test cases executed.
* Final Testing Report prepared.

All the software items will be installed to Production environment.

Sanity Check in Production environment will be performed before release.

All functional requirements will be tested. Non functional requirements will not be tested.

Requirements are clear and currently not changing.

Test Team will create test cases to cover all functional requirements and make sure that all features will be tested thoroughly.

# Test Items

Software to be tested in this project: Data management system (DMS) web application.

Interfaces to be tested:

* GUI of DMS

## Features to Be Tested

All the features of Document Management System, which were defined in software requirement specifications under functional requirements are to be tested.

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Applicable Roles** | **Description** |
| Login | Admin  User | Users and administrators can login and work with DMS |
| User creation | Admin | Admin can create a new user account |
| Change password | Admin | Admin can change the password of an existing user |
| User group creation | Admin | Admin can create a new user group and assign users to it. A user can be assigned to multiple groups. A group can have multiple users assigned. |
| Document type creation | Admin | Admin can create a new document type and assign groups that can create and/or accept/reject such doc type. Group can have multiple document types assigned, allowing group members to create and/or accept/reject multiple doc types. |
| Document creation | User | A user, assigned to a group (group member) can create a document of a specific type, if he/she belongs to at least one group that has the permission rights to create documents of that type. |
| Document acception /rejection | User | A group member can accept/reject a document of a specific type, if he/she belongs to at least one group that has the permission rights to accept/reject documents of that type. |
| Document Status | User | Documents will have different status, depending on state in approval process: created, submitted, accepted, rejected. |
| User view | User | Users can view their created documents (not yet submitted). User can view their submitted documents. |
| Group view | User | Members of a user group that accepts documents can see documents submitted for approval of a specific type. |
| Document view | User | Documents can be be reviewed, accepted and rejected. Document has required fields: Unique id, author, type, title, summary, submission date(if submitted), acceptance date(if accepted), rejection date(if rejected), document acceptant(if document is accepted or rejected), reason of rejection(if rejected), attached files(one or more .pdf type files). |
| Document Download | User | A user can download an archive file containing all the documents (with all statuses) where that user is the creator. |
| Statistics | User | Members of a user group that accepts documents can see documents submission, acceptance and rejection statistics, as well as statistics on users with most submissions. |

Test Cases will include: Test ID, Priority, Title, Preconditions, Test Steps, Expected Results, Actual Results, Status, Reference, Automated or not, Comments.

## Features Not to Be Tested

Non-functional testing such as stress, performance or logical database currently will not be tested (out of scope) as well as Database testing and any other external interfaces.

There are no common functionalities with other projects.

Regression tests will be executed upon update of an old component.

# Testing Approach

The scope of testing includes Verification and Validation of the system under test.

Verification means that it is made sure that all specified requirements are fulfilled. This is mainly done through reviews, inspections and testing.

Validation is the confirmation that the requirements for specific use have been fulfilled

The material used as a base for creating test cases are requirements and technical architecture for the more technical testing levels as Module Testing. Use cases are used for more business like testing as Acceptance Testing and the testing levels in between use a combination of these.

Testing should be done from three different points of view. These are technical testing, integration testing and requirements fulfilment. The technical testing should verify that system functionalities are working internally, integration testing that functionalities are working together and requirements fulfilment that all items created successfully meet the requirements.

<What kind of aspects should be taken to testing e.g. is there need for technical and functional testing separately?>

<What are the main objectives with testing in this project? For example “the main point in this project is to successfully meet the release exit criteria”>

## Prioritization

<Items under test should be prioritized to verify that the more business critical items and/or more difficult to implement items are tested the most carefully. Here is the list you may use to help you with test item prioritization>

<Implemented functionalities

Common new interfaces

Basic use cases or end to end scenarios

Error scenarios and special cases

Old functionalities (Note! Regression Testing should still be carried out separately)

Old interfaces

Abnormal usage of the system >

<Note that it is highly recommended to have maximum ~20% of the items in the highest priority level in the first phase. During the project limit the amount to maximum ~30%>.

## Entry/Exit Criteria

Entry criteria define the level where we should be before the actual testing can be started. Exit criteria describe the level where we should be after the testing is completed.

<**Entry criteria** for each testing level are:

Module Testing:

Documentation is prepared

Module Development environment is prepared

Module is developed

System Testing:

Functionality is working for current sprint

Documentation and requirements are approved

UAT:

Environment is prepared for UAT

Functional and Non Funtional requirements are tested

Environment in question is up and running

Environment in question is stable

Environment in question has been fully allocated for testing

Software items have been implemented and installed to the correct environment

All needed configurations have been completed

Required test data has been imported or created

Base documents are approved and no further changes are expected

Test cases are created based on correct documents

Test cases are reviewed and approved

Earlier testing level exit criteria has been fulfilled >

<**Exit criteria** for each testing level include:

Module Testing:

Module is covered with unit tests (%???)

Unit tests do not fail

System Testing:

Test Cases are documented

Bugs are fixed(%???)

UAT:

PO approves existing functionality.

Test Cases related to testing level are agreed to cover enough of the requirements

All urgent and high severity defects related are fixed and closed

All Test Cases related to testing level in question are executed

All entry and exit criteria are fulfilled>

<Add any additional entry criteria to list, i.e. when the testing could be started e.g. test cases are approved, test environment needs to be up and running, code needs to be compiled, access rights need to be granted>

<List of exit criteria, i.e. when the testing can be considered as completed, e.g. test cases executed, all high severity defects are closed>

## Item Pass/Fail Criteria

<Item passes if it is proved that:

The material used as the base for testing is approved

The test case is correctly based on the correct test base, i.e. Use case or requirement

The test case is reviewed and approved

All test cases related to item are successfully run

The coverage of the test cases is seen and approved to cover enough

The item is seen as functional from integrated parts as well>

<List here any additional criteria for item to pass, e.g. should testware and test case be approved, should integration be working>

Item fails if there are:

<Misunderstandings in the test base material that cause test cases to fail from some point of view

Code is not correct

Human error may cause item to fail in testing phase

Item can not be tested because of test environment issues

If there are any urgent or high severity defects open that are related to some other related issue

If the item is not created at all

Item is not functional because of some other integrated system>

<List here any additional criteria for item to fail, e.g. item not functional, human errors, misunderstandings and integration errors>

## Classification of Defects

The table below shows the classification of the defect during the implementation project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Severity** | **Description** | **Impact to testing** | **Actions** | **Analysis & Fix schedule** |
| **Urgent Defect** | Defect causes that important functionalities are not achievable, that data disappears, performance deteriorates or parts of the system crash | showstopper for testing | Example: Needs actions immediately | Example: Defect analysis should be started immediately, fix delivery time ASAP |
| **High Defect** | Defect means that a functionality does not work according to specifications or does not work at all | Essential function not functional | Example: Needs actions soon | Example: Defect analysis should be started latest in one to two days, fix delivery in one week |
| **Medium Defect** | With Medium severity defect it is meant that some part of the system does not work normally but needs some special actions in order to get all the normal functionalities | Some part of function is not working | Example: Can be accepted, actions to be decided in the Steering group | Example: Defect analysis should be started in one week, fix delivery in two weeks |
| **Low Defect** | With a Low severity defect it is meant a defect that is mainly cosmetic and does not have any effect on functionality | Cosmetic error, e.g. wrong error message | Example: Can be accepted | Example: Defect analysis should be started in one week if there is time and higher severity defects are fixed, fix delivery in one month |

<Please agree the severity levels and their descriptions with the project team. Note the differences of critical defects between business and technical point of views.>

## Resumption and Suspension Criteria

<This chapter should include information when we have done enough testing. This means that the lowest testing exit criteria should be mentioned. You can refer to Prioritization-chapter and add here for example information which prioritization levels must be tested before a testing level and finally the whole system can be approved>

<Are there any approval criteria given in the contracts? Refer to chapter “Release Criteria”>

<How many defects and which severity can be open to get the product approved?>

<Are there other conditions to receive approval for the system?>

<Is there some rule for the maintenance work or defects that may occur in the production?>

<Are there different approval levels present? For example developers approve the Module Testing, test team in the System Testing and System Integration Testing and customer Acceptance Testing>

<How is one level approved? Are there any special criteria that for example customer should see the results of each and every testing level to approve those separately>

<Which items should receive approval to get the whole system approved? This refers to prioritizations that the most business critical parts should be tested more carefully than the less business critical parts>

<Who approves the product and how?>

<Is there Acceptance Testing for this?>

<Write reference to Acceptance Testing Plan>

# Testing Levels

## Module Testing

<This testing level is the lowest testing level and means that components of code are tested component by component. May be for example structural testing where the program logic, paths through code and the efficiency of the code is tested. The objective of Module Testing is to establish that each component of the software works in isolation and fulfils its component functionality.>

After this testing level the items should be ready for System Testing after some Smoke Testing.

### Test Responsibilities

Module testing will be conducted by the development team informally as part of the development of the system.

### Test Methods

Each developer is responsible for testing his/her own code.

## System Testing

System Testing is a technical testing type to verify that a system meets its specified requirements. Here the behaviour of the whole system is tested. Test cases should be based on requirements and risks.

This testing level should be executed by test team. The actual testing should be completed in the Test environment that is fully allocated for testing at this point.

Between Module Testing and System Testing some Smoke Testing (see [chapter 5.5.1](#_heading=h.2xcytpi)) should be executed to approve the system to be ready for System Testing. The system should be ready for System Integration Testing when the exit criteria for this testing level are fulfilled.

### Test Responsibilities

<Who is responsible for this testing level?>

Test team is responsible for this level of testing.

<Who is responsible for defining test cases?>

Test team is responsible for defining and creating test cases and executing tests.

<What is the base material for test cases?>

DMS software requirement specification

<Who is responsible for this material?>

PO - Giedrius Graževičius

<Who reviews test cases?>

### Test Methods

<What are the methods for this testing level, e.g. test team will create test cases and execute these after approval according to an earlier decided schedule?>

Test team will create test cases and execute these according to test case priority and an earlier decided schedule.

Test Cases will include: Test ID, Priority, Title, Preconditions, Test Steps, Expected Results, Actual Results, Status, Reference, Automated or not, Comments.

<How test cases are created? Meaning that is there any special test technique used or similar>

Test cases will be filled in excel file will all required fields(mentioned above) filled.

<How the testing is done?>

1. Developer marks user story as done(available to test feature)
2. Available to test features with highest priority will be selected from the test case list.
3. Test case is executed following exact test steps.

### Entry and Exit Criteria

<Define entry criteria>

* Defined and Approved Requirements
* Test Plan
* Test Cases and Test Data
* Test Tools
* Testable Code with Appropriate Test Environment

<What has been done before this to be ready to start testing?>

Unclarity about requirements clear out with PO. Test plan, test cases and test data? prepared. Testing tools installed. DMS1 web application available for testing.

<Define exit criteria>

* Ensuring all critical Test Cases are passed
* Achieving complete Functional Coverage
* Identifying and fixing all the high-priority defects
* Fixing all the ‘Show Stopper defects’ or ‘Blockers’ and ensuring that none of the identified Critical/Severity 1 defects are in Open Status
* Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully

<What has to be done to complete this testing level?>

All critical Test cases executed. All Test Cases for functional requirements are created. Defects are registered.

<What are the results and how those are delivered?>

The results will be shown in the final Test Report.

<What has to be approved to finish with this testing level? Add here any additional information if there are some special approval methods for this testing level>

It should be approved that functional coverage is complete and high-priority defects are closed.

Gal reikia pamineti kokius testavimus darysim?

Functional testing will be covered by test cases:

* Black-box testing ( Happy Path as well as Negative testing approach will be applied)
* Database testing

Non-functional testing will be covered via checklist:

* Security testing
* Performance testing
* Browser Compatibility testing

## System Integration Testing

System Integration Testing is a technical testing type to verify that a system meets its specified requirements from the integration point of view. Here the behaviour integrations to other systems are tested. Test cases should be based on requirements and risks.

This testing level should be executed by test. The actual testing should be completed in the Test environment that is fully allocated for testing at this point.

The system should be ready for Acceptance Testing when the exit criteria for this testing level are fulfilled.

Because the scope of the project is limited to the creation of the document management system, which has no external system integrations (as per software requirements provided), there will be no formal system integration testing. ???

## Acceptance Testing

Acceptance Testing is a formal testing type where the objective is to verify that a system meets its specified requirements from the user point of view. Most often the use cases or similar end to end business scenarios are tested here.

This testing level should be executed by customer and there should be an Acceptance Testing Coordinator available to plan and monitor the testing. Customer should include end users or other authorized entities to determine whether to accept a system or component. The actual testing should be completed in the environment as production like as possible but not in production if the production is already in use.

The system should be ready for production use when the exit criteria for this testing level are fulfilled.

### Test Responsibilities

The business, customer, will test this testing level. This means that usually the end users are the testers as mentioned above. You may refer to Acceptance Test Plan to check for the information from there.

## Other Testing Level Types

We have a few other testing types that should be taken into consideration while planning, designing and executing tests. These testing types are Smoke Testing, Regression Testing and PROD Sanity Check.

### Smoke Testing

Smoke Testing is testing basic operations in the system without planned test cases to show that the system is functional. This should be executed every time new version of code is delivered to an environment to verify that the quality of the code is high enough. Smoke Testing is one of the main entry criteria for System Testing.

In Module Testing level the person doing the installations is responsible for doing Smoke Testing. In the System and System Integration Testing levels, that happen in the Test environment the responsible person or team is the test team. If Acceptance Testing is done in another environment, then the new version in the environment should also be smoke tested before starting the actual testing and that should be carried out by the customer.

### Regression Testing

Regression Testing is testing some certain set of test cases to verify that a system has not been affected by updates as new version or defect fixed. Regression test set should be executed every time new version is installed to a system or related defect fixed.

Difference to Smoke Testing is that here test cases are designed and approved and form a ready-made set of test cases based on normal use cases. The regression test set can be executed as a whole or only a part of the test set may be chosen in special cases where smaller updates are done to the system.

### PROD Sanity Check

PROD Sanity Check means that the Production environment is verified to be operational by some small use cases after installation. It should be verified that all servers are up and running and graphical user interface is operational. Also it should be verified that all basic functionalities work as for example search-functionalities.

This testing type is a responsibility of the person doing the installations. ???

# Test Schedule, Organization and Responsibilities

## Testing Schedule

|  |  |
| --- | --- |
| **Event** | **Date** |
| 1st Sprint review | 2019-02-07 |
| 2nd Sprint review | 2019-02-28 |
| 3rd Sprint review | 2019-03-20 |
| Release | 2019-03-27 |

## Test Project Organization & Resources

## Tools and Access Rights Required

## Training Needs

# Tool for Test Design and Execution

General: browsers (mozilla, chrome, safari), OS: Windows, MacOS, Linux?

Manual Testing: Microsoft Excel (Test Cases, Checklists, Coverage graphs), Microsoft Word (Test Plan), Jira (backlog and sprint, bug registration).

Automation Testing: Selenium WebDriver, Eclipse.

# Test Environments

## Development Environment

### Actions Required for Development Environment

## Test Environment

### Actions Required for Test Environment

## Production Environment

### Actions Required for Production Environment

# Test Metrics

## Schedule for Test Metrics collection

## Metrics Types

# Risks

The following risks have been identified:

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Severity** | **Probability** | **Course of Action** |
| The project schedule is too tight; it's hard to complete this project on time | Medium | 30% | Set **Test Priority** for each of the test activity. |

# Deliverables

This chapter will explain what kind of deliverables are needed for testing to be successful and to document planning and results of the testing period.

## Test Items

<What has been approved in the contracts that will be delivered to customer?> Build and tested data management system which allows to create, accept and save organizations documents. System allows to create user groups which can submit and accept documents.

<Is there some specific test data or migration data to be delivered?>

<What documents are required from testing side?> Test Plan, Test cases and Final Test Report

<Development environment, Test environment, Production environment>

<Code from X, Code from Y, Test cases for Module tests, Test cases for System tests> Automation code from Test team members, Test Cases as a separate document.

<Test cases for System Integration tests, Test cases for Acceptance tests> Separate document

## Test Documents

Master Test Plan

Test Cases (including Test Data)

Test status report presentation for each sprint review

Final Test Report

# Change history (document)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Reviewed by** | **Approved by** | **Change history** |
| V1.0 | 2020-01-24 | Inga Chamentauskienė | Gintarė Urbanavičiūtė | <Name> | First version |