BRLS MASTER TEST PLAN

Version Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Author** |
|  |  |  |  |
|  |  |  |  |

Management summary

|  |  |  |
| --- | --- | --- |
| **Project objective**  <…> | | |
| **Test objective and assignment**  <…> | | |
| **Short description of the test approach**  <…> | | |
| **Results to be realized** | | |
| *Result*   * < example: well executed and finished system­ test> * < example: well executed and finished user ­acceptance test> * < example: well executed and finished total test­ project> | *Document*   * ST Test report * UAT Test report * End report Testing | *Delivery date* <mm-dd-yyyy>  <mm-dd-yyyy>  <mm-dd-yyyy> |
| **Qualitative objectives**  <example: Each test level needs to be completed on time and it needs to be clear for each system object if it meets the acceptance criteria> | | |

Table of Contents

[1 Introduction 4](#_Toc17985169)

[1.1 Project and project objective 4](#_Toc17985170)

[1.2 Objective of the master test plan 4](#_Toc17985171)

[2 Documentation 5](#_Toc17985172)

[2.1 Basis for the master test plan 5](#_Toc17985173)

[2.2 Test basis 5](#_Toc17985174)

[3 Test strategy 6](#_Toc17985175)

[3.1 Risk analyses 6](#_Toc17985176)

[3.1.1 Product risk analyses 6](#_Toc17985177)

[3.1.2 Technical risk analyses 7](#_Toc17985178)

[3.2 Risk analyses 7](#_Toc17985179)

[3.3 Test strategy 7](#_Toc17985180)

[4 Approach 9](#_Toc17985181)

[4.1 Test levels 9](#_Toc17985182)

[4.2 The <name test level> 9](#_Toc17985183)

[4.2.1 Goal 9](#_Toc17985184)

[4.2.2 Short description 9](#_Toc17985185)

[4.2.3 Entrance and exit criteria for each test level 9](#_Toc17985186)

[4.2.4 Test environments 9](#_Toc17985187)

[4.2.5 Defects procedure 9](#_Toc17985188)

# Introduction

## Project and project objective

<< Describe concisely the project and the project objectives. >>

This master test plan fits to the project plan <name project plan>.

## Objective of the master test plan

The objective of the Master Test Plan (MTP) is to inform all who are involved in the test process about the approach, the activities, including the mutual relations and dependencies, and the (end) products to be delivered for the test project <project name>.

The master test plan describes this approach, the activities and (end) products that need further elaboration in the other system test plans. These system test plans need to be abstracted from this master test plan.

# Documentation

This chapter describes the documentation used in relation with the master test plan. The described documentation concerns a first inventory and will be elaborated, actualized and detailed at a later stage, during the separate test levels.

## Basis for the master test plan

<< Consider the Project Vision, requirements analyses, specific project or test planning, an implementation plan or other documents of importance. >>

The following documents are used as basis for this master test plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
|  |  |  |  |
|  |  |  |  |

## Test basis

The test basis contains the documentation that serves as basis for the tests that have to be executed. The overview below describes the documentation that is the starting point for testing.

<< Consider requirements analysis documents, technical designs, data models, system architecture, user manuals, ‘old’ testware and AO-procedures >>.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
|  |  |  |  |
|  |  |  |  |

<< If it’s already definite that the test basis is (partly) missing or is of poor quality, also mention here the measures taken in this area, for example interviews to get the necessary information on the table. It is also possible to mention the document type if the document is not yet available at the time of writing this document. >>

# Test strategy

The time available for testing is limited; not everything can be tested with equal thoroughness. This means that choices have to be made regarding the depth of testing. Also it is strived to divide test capacity as effective and efficient as possible over the total test project. This principle is the basis of the test strategy.

The test strategy is based on risks: a system has to function in practice to an extent that no unacceptable risks for the organization arise from it. If the delivery of a system brings along many risks, thorough testing needs to be put in place; the opposite of the spectrum is also true: 'no risk, no test'.

The first step in determining the test strategy is the execution of a product risk analyses. This is elaborated in §3.1.

The test strategy is subsequently based on the results of the risk analyses. The test strategy lays down what, how and when (in which test level) is being tested and is focused in finding the most important defects as early as possible for the lowest costs. This can be summarized as testing with an optimal use of the available capacity and time. The test strategy is described in §3.3.

## Risk analyses

## Product risk analyses

The product risks are determined in cooperation with the client and the other parties involved. Product risks are those risks associated with the final product failing to meet functional requirements and required system quality characteristics (NFRs) This product risk analyses (PRA) is comprised of two steps:

* Make an inventory of the risks that are of interest
* Classify the risks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Product Risk** | **Description** | **Characteristic** | **Classification** |
| <…> | << Description of the mentioned risk>> | <functionality, performance, user-friendliness, suitability, etc.> |  |
|  |  |  |  |
|  |  |  |  |

The extent of the risk (the risk class) is dependent on the chance of failure (how big the chance is that it goes wrong?) and it depends on the damage for the organization if it actually occurs.

## Technical risk analyses

Technical risks are determined in cooperation with the analyst/designers and programmers involved. Technical risks are development risks associated with failing to create a system that behaves according to specifications derived from requirements. (I.E. those aspects of development that pose particular challenges.) This technical risk analyses (TRA) is comprised of two steps:

* Make an inventory of the risks that are of interest
* Classify the risks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Technical risk** | **Description** | **Characteristic** |  |
| <…> | << Description of the mentioned test goal >> | <functionality, performance, user-friendliness, suitability, etc.> |  |
|  |  |  |  |
|  |  |  |  |

## Risk analyses

## Test strategy

For each risk from the product and technical risk analysis the risk class dwtwermines the thoroughness of the test. Risk class A is the highest risk class and C the lowest. The test strategy is subsequently focused on covering the risks with the highest risk class as early as possible in the test project.

<<Note: the content of the table below is only an example! Risk class A has to have in at least one test level a high thoroughness of the dynamic test (), risk class B has to have in at least one test level a medium thoroughness of the dynamic test () and risk class C has to have in minimal one test level a limited thoroughness of the dynamic test ()>>

<< **Attention**: There are some test levels mentioned in this table, but this is only done as an example. It can be possible that in your project there are more/less and/or other than the in this table mentioned test levels >>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk** | **RC** | **SR** | **Unit** | **Integration** | **ST** | **FAT** | **UAT** |
| Functionality | A/B/C |  |  |  |  |  |  |
| - part 1 |  |  |  |  |  |  |  |
| - part 2 |  |  |  |  |  |  |  |
| - total |  |  |  |  |  |  |  |
| User-friendliness |  |  |  |  |  |  |  |
| Performance |  |  |  |  |  |  |  |
| Security |  |  |  |  |  |  |  |
| Suitability |  |  |  |  |  |  |  |

Legend for the table above:

|  |  |
| --- | --- |
| RC | Risk class (from product and technical risk analysis, where A=high risk, B=average risk, C=low risk) |
| SR | Static Review of the various intermediary products (requirements, functional design, technical design) |
| Unit | Unit test and Unit integration test |
| Integration | Integration tests (low level (L), high level(H)) |
| ST | System test (functional scenario testing (F), system quality scenario testing (S)) |
| FAT | Functional acceptance test (alpha stage UAT) |
| UAT | User acceptance test (Beta stage UAT) |
|  | Limited thoroughness of the dynamic test |
|  | Medium thoroughness of the dynamic test |
|  | High thoroughness of the dynamic test |
| S | Static testing (checking and examining the products without executing the software |
| I | Implicit testing (including in another test type without creating specifically designed test cases |
| <blank> | If a cell is blank, it means that the relevant test or evaluation level does not have to be concerned with the characteristic |

# Approach

<< In this section each test level in the test strategy (the what) will be translated to a concrete test approach (the how). Make sure that the described test approach reflects the test strategy identified in Section 3! Each element from the test strategy has to return here!>>

## Test levels

<< List the several test levels (Static Review, Unit Test, Integration Testing, Functional Acceptance Testing, etc.) The details will be in a separate paragraph for each test level >>

For this MTP the following test levels are acknowledged:

|  |  |
| --- | --- |
| **Test level** | **Goal** |
|  |  |
|  |  |

## The <name test level>

### Goal

<< What is the goal of the test level. >>

### Short description

<<Short description on the contents of the test level (what characteristics, who specifies, what test goals are covered, who executes and on which kind of test environment). Subsequently describe for each characteristic how the risks concerned are being verified and/or tested for this test level.>>

## Entrance and exit criteria for each test level

## Test environments

<< Describe the demands that the test level makes to the corresponding environment. Mention here a description of the infrastructure components, test data/ files. Consider the demands that are being made from the test strategy and approach to the test environment.

## Defects procedure

<<Describe what you are going to do if/when a defect is detected at this test level.>>