|  |
| --- |
| Expense tracker  Master test plan |

Table of Contents

[1 Introduction 1](#_Toc81236051)

[1.1 Project and project objective 1](#_Toc81236052)

[1.2 Objective of the master test plan 1](#_Toc81236053)

[1.3 Involved in creating the master test plan 1](#_Toc81236054)

[2 Assignment formulation 2](#_Toc81236055)

[2.1 Client 2](#_Toc81236056)

[2.2 Supplier 2](#_Toc81236057)

[2.3 Assignment 2](#_Toc81236058)

[2.4 Scope 2](#_Toc81236059)

[2.4.1 Within scope 2](#_Toc81236060)

[2.4.2 Out of scope 3](#_Toc81236061)

[2.5 Preconditions and assumptions 3](#_Toc81236062)

[2.5.1 Test environments 4](#_Toc81236063)

[2.6 Acceptants and acceptance criteria 4](#_Toc81236064)

[2.6.1 Acceptants 4](#_Toc81236065)

[2.6.2 Acceptation criteria 4](#_Toc81236066)

[3 DocumentatiON 5](#_Toc81236067)

[3.1 Basis for the master test plan 5](#_Toc81236068)

[3.2 Standards 5](#_Toc81236069)

[3.3 Test basis 5](#_Toc81236070)

[4 Test strategy 6](#_Toc81236071)

[4.1 Product risk analyses 6](#_Toc81236072)

[4.2 Test strategy 7](#_Toc81236073)

[5 Approach 9](#_Toc81236074)

[5.1 Test levels 9](#_Toc81236075)

[5.2 Evaluation 9](#_Toc81236076)

[5.3 The <name test level> 9](#_Toc81236077)

[5.3.1 Goal 9](#_Toc81236078)

[5.3.2 Short description 9](#_Toc81236079)

[5.3.3 Responsible 9](#_Toc81236080)

[5.3.4 << Optional: test environment to be used >> 10](#_Toc81236081)

[5.4 Phasing per test level 10](#_Toc81236082)

[5.5 Test products 11](#_Toc81236083)

[5.6 Review plan 11](#_Toc81236084)

[5.7 Entrance and exit criteria for each test level 12](#_Toc81236085)

[5.7.1 << Optional: Functional Acceptance Test >> 12](#_Toc81236086)

[5.7.2 << Optional: User Acceptance Test >> 12](#_Toc81236087)

[5.8 Go/No go 12](#_Toc81236088)

[6 Organization 14](#_Toc81236089)

[6.1 Organization structure 14](#_Toc81236090)

[6.2 Roles, tasks and responsibilities 14](#_Toc81236091)

[6.2.1 << Optional: Trainings- and coaching’s necessity >> 14](#_Toc81236092)

[6.3 Structure of meetings 14](#_Toc81236093)

[6.4 Structure of reporting 15](#_Toc81236094)

[6.5 Completion 15](#_Toc81236095)

[7 Infrastructure 16](#_Toc81236096)

[7.1 Test environments 16](#_Toc81236097)

[7.2 Test tools 16](#_Toc81236098)

[7.3 Office setup 17](#_Toc81236099)

[8 Management 18](#_Toc81236100)

[8.1 Test process management 18](#_Toc81236101)

[8.2 Test infrastructure management 18](#_Toc81236102)

[8.3 Test product management 18](#_Toc81236103)

[8.4 Defects procedure 18](#_Toc81236104)

[9 Test process risks and countermeasures 19](#_Toc81236105)

[10 Global Estimation & Planning 20](#_Toc81236106)

[10.1 Estimation 20](#_Toc81236107)

[10.2 Planning 20](#_Toc81236108)

[10.3 Milestones 21](#_Toc81236109)

[11 Glossary 22](#_Toc81236110)

Version information

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Author** |
| 1.0 | 00/00/2021 | Alpha | Oren S.Rauch |
|  |  |  |  |

Distribution list

|  |  |
| --- | --- |
| **Name** | **Company/Function** |
| All Stakeholders | SolveQA.com |
| All Students | SolveQA.com |

Approval Client

<< Optional: the approval of the client can be represented here. In that case the signed hardcopy needs to be archived. >>

|  |  |  |
| --- | --- | --- |
| Client: | | Signature |
| Name | <Name client> |  |
| Division | <Division> |  |
| Department | <Department> |  |
| Function | <Function> |  |
| Location | <Location> |  |
| Telephone | <Telephone nr.> |  |
| E-Mail address | <email address> | Date: <date> |

Management summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Project objective** | | | |
| **Test objective and assignment**   * To understand all Client requirements. * To perform all test goals as explained later in this plan. * To perform all test cases as written in “Test-Cases-1.0-Alpha” Doc. | | | |
| **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> | | | |
| **Estimate**  TBA | | | |
| **Test process risks and measures** | | | |
| *Test process risks*  • <…> | *Measures to be taken*  • <…> | | |
| **Go/no-go decisions**  < Example: After each test level the test manager makes sure that a test report is drawn up. This report will, after review with the project manager, be presented to the key stakeholders, who then decide if it is possible to go to the next test level.  At the end of the test project a test end report will be drawn up, containing a risk based assessment of the test object. Based on this end report the key stakeholders make the final decision to go to production or not. > | | | |

# Introduction

## Project and project objective

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

This master test plan fits to the project plan “Expense Tracker 1.0”.

## 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.

## Involved in creating the master test plan

|  |  |  |
| --- | --- | --- |
| **Name** | **Function** | **Responsibility** |
| User A | Test Team Manager | <Write MTP> |
| User B | PM | <Review MTP> |
| User C | Lead Tester | <Execute MTP> |
|  |  |  |

# Assignment formulation

## Client

SolveQA Finance Corp.

## Supplier

SolveQA.com Testing Team

## Assignment

The goal is to Sign-off installable and no client side defects app that successfully passed all testing team exit cretiria and passed the testing metrics bar

## Scope

App frontend/

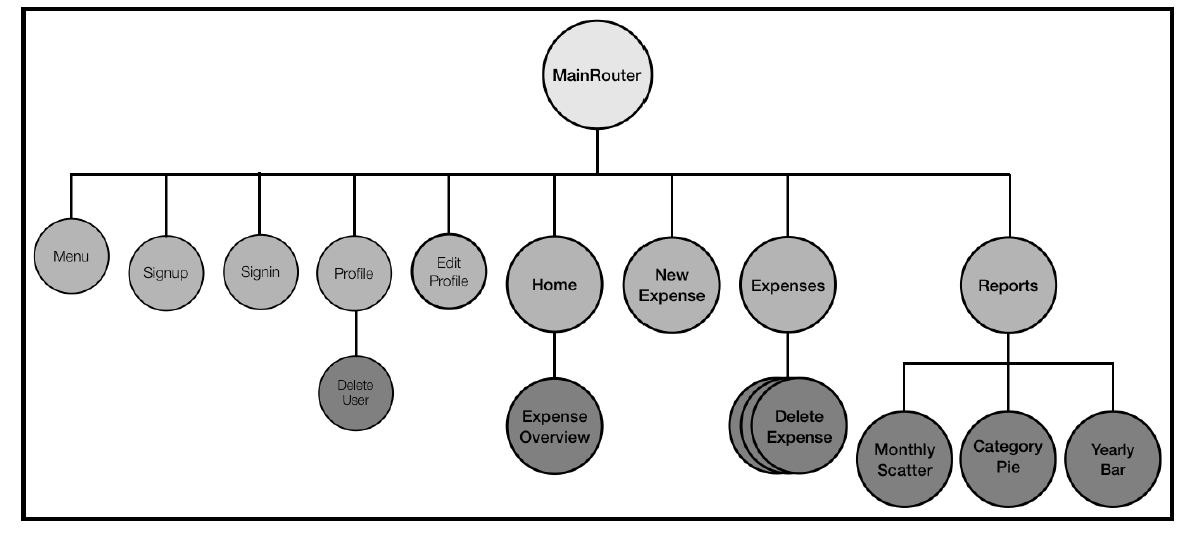
The scope of the project includes testing the following features of

Expensetracker.com web application.

### Within scope

* SignUp
* SignOut
* SignIn
* Add Expense
* Expenses
* Reports
* Home Page
* My Profile

As per our understanding, we believe the areas above need to be tested.



### Out of scope

<< Describe in concrete terms what parts are out of scope for testing. In addition to the things from the previous paragraph (§2.4.1) also think of:

* System changes not included in the project (e.g. hardware changes to the Mainframe platform);
* Test activities executed by others;
* Reorganizations;
* Possible future projects that influence the current project (particularly if there are still uncertainties about the other project).

Also include who is responsible for this. Also think of system changes that are not included in the project, test levels/-activities that are executed by other parties, reorganizations and possible future projects that can be of influence.

See TMap® Next 5.2.1 >>

## Preconditions and assumptions

Preconditions concern conditions that third parties like the client, the project or the users, impose to the test process and within which the test process must operate (definition TMap® Next). The following demands are enforced:

<< Preconditions, for example:

* The test project has to be finished not later than <determined date>;
* The prevailing project plan is boundary for this master test plan and the execution of the test project is based on this >>

Assumptions are external circumstances or events that must occur to ensure the test process’ success, but that cannot be controlled by the test process. In other words, these are the requirements of the test process vis-à-vis others (definition TMap® Next).

<< Assumptions, for example:

* The line organizations are responsible for the execution of the following activities:
  + Reviewing test specifications / test scripts, within the fixed period of xxx days;
  + Providing resources for UAT that are dedicated and prepared
  + ….
* The delivery planning of the project has to be tuned with, and where necessary adapted to, the sequence that is desired from the test project.
* The documents identified as test basis need to be accepted by stakeholders (including the test team), before the test specification can be started.
* Changes on baselined documents need to follow the formal change procedure.

### Test environments

* Windows 10 – Browsers: Chrome, Firefox, Edge
* Mac OS – Brower: Safari
* Android Mobile OS – Chrome, Brave
* Iphone OS – Safari, Chrome, Brave

## Acceptants and acceptance criteria

<< See TMap® Next 5.2.2 >>

### Acceptants

The table below states the acceptants of <system>:

|  |  |  |
| --- | --- | --- |
| **Name** | **Function** | **Department** |
|  |  |  |
|  |  |  |

### Acceptation criteria

The table below states which acceptance criteria there are for <system> and to which standard they should apply:

|  |  |
| --- | --- |
| **Description** | **Standard** |
|  |  |
|  |  |

# 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.

<< See TMap® Next 5.2.2. >>

## Basis for the master test plan

<< Consider a Project plan or a Plan of Approach for the project, 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** |
|  |  |  |  |
|  |  |  |  |

## Standards

<< With regards to testing consider TPI or test guide books. Development standards, documentation standards or quality conventions can also be used. >>

The following conventions and standards are applied for this test plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| TMap® Next for result driven testing | 1e edition | 2006 | T. Koomen, L. van der Aalst, B. Broekman en M. Vroon |
|  |  |  |  |

## 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 a functional or technical design, 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 nterviews 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 §4.1.

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 §4.2.

<< See also

* Product risk analyses: TMap® Next 5.2.3 and 9
* Test strategy: TMap® Next 5.2.4 >>

## Product risk analyses

The product risks are determined in cooperation with the client and the other parties involved. This product risk analyses (PRA) is comprised of two steps:

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

The complete product risk analysis is mentioned in appendix <appendix number>.

During the risk assessment the test goals were also formulated. These can be found together with the corresponding characteristics in table below.

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

The acceptants <optional: and other parties involved with the project> have determined the product risks. 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.

The risk class (RC) determines the thoroughness of the test. Risk class A is the highest risk class and C is 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.

First the chance of failure and damage are determined for each risk. The risk class has been taken directly from this. <<Note: This is the contents from the TMap® Next table (See TMap® Next 9.6). Evidently it’s possible to deviate from this in consultation with the client.>>.

#### Risk table

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Part** | **RC** | **Argumentation** |
| Functionality | … | A/B/C | … |
| User-friendliness |  |  |  |
| Performance |  |  |  |
| Security |  |  |  |
| Suitability |  |  |  |
| Etc. |  |  |  |

## Test strategy

For each risk from the product risk analysis the risk class is qualifying 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 >>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Characteristic /object part** | **PRA-RK** | **Evaluation** | **Development test** | **ST** | **FAT** | **UAT** | **Impl** |
| Functionality | A/B/C |  |  |  |  |  |  |
| - part 1 |  |  |  |  |  |  |  |
| - part 2 |  |  |  |  |  |  |  |
| - total |  |  |  |  |  |  |  |
| User-friendliness |  |  |  |  |  |  |  |
| Performance |  |  |  |  |  |  |  |
| - online |  |  |  |  |  |  |  |
| - batch |  |  |  |  |  |  |  |
| Security |  |  |  |  |  |  |  |
| Suitability |  |  |  |  |  |  |  |

<Explanation for the table above:

|  |  |
| --- | --- |
| PRA-RC | Risk class (from product risk analysis, where A=high risk, B=average risk, C=low risk) |
| Evaluation | Evaluation/review of the various intermediary products (requirements, functional design, technical design) |
| Development test | Unit test and Unit integration test |
| ST | System test |
| FAT | Functional acceptance test |
| UAT | User acceptance test |
| Impl | Implementation |
| ● | 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 chapter 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 from chapter 4! Each element from the test strategy has to return here! This paragraph can be more concise if there will be test plans (TP) drawn up for each test level. (Refer to the TP’s that have to be written). There are two important factors that determine whether TP’s are being written or not:

* The size of the project;
* The level of uncertainties and ambiguities that are there at the moment of writing the MTP.>>

## Test levels

<< List the several test levels (System Test, Functional Acceptance Test, Production Acceptance Test, etc.) and evaluations where appointed in the test strategy. The details will be in a separate paragraph for each test level >>

For this MTP the following test levels are acknowledged:

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

## Evaluation

<< Describe the evaluation strategy, if evaluation belongs to the scope of this master test plan. >>

## 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. >>

### Responsible

<< The responsible and/or people involved or department for the execution of the test level. Consider the following:

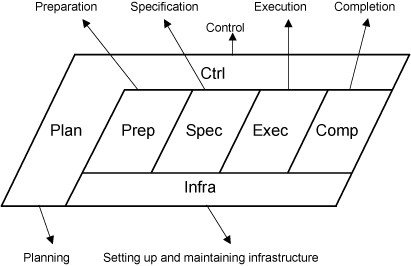
* The test manager is **actually** responsible for these tests: in that case there should not be anyone else mentioned as responsible. In that case it concerns the 'involved' or, even better, the 'executers' (from that external party), who are managed by the test manager (otherwise the test manager can’t be responsible)
* The test manager is **not** responsible for these tests (in that case these tests or test levels have to be listed in §2.4.2 “Out of scope”): it is possible to make certain demands (mainly in the form of exit criteria that have to be met) to these tests, naturally after obtaining agreement with the external parties.
* The test levels concerned are described in the test strategy of the MTP, but the test manager is **not** responsible for these tests. He is responsible for obtaining agreement regarding the tests with the external parties, as well with the realization of the test strategy and with the execution of it, but not for the result and quality of these tests, as he has no mandate in this. Appoint this construction explicitly with paragraph 2.4.1 and 2.4.2 (In that case the design of the test strategy is within scope and is including the 'external tests' and verification of the exit criteria. Out of scope is the actual management and execution of these tests).

See for more in chapter 12 of this MTP. >>

### << Optional: test environment to be used >>

<< The <test level> will be executed on the <test environment>. This will be elaborated in chapter 13. >>

## Phasing per test level



In the **Planning** phase, the test manager formulates a coherent approach that is supported by the client to adequately execute the test assignment. This is laid down in the test plan. In the **Control** phase the activities in the test plan are executed, monitored, and adjusted if necessary. The **Setting up and maintaining infrastructure** phase aims to provide the required test infrastructure that is used in the various TMap phases and activities. The **Preparation** phase aims to have access to a test basis, agreed with the client of the test, of adequate quality to design the test cases. The tests are specified in the Specification phase and executed in the **Execution** phase. This provides insight into the quality of the test object. The test assignment is concluded in the **Completion** phase. This phase offers the opportunity to learn lessons from experiences gained in the project. Furthermore activities are executed to guarantee reuse of products.

## Test products

<< See TMap® Next 5.2.7 >>

The deliverables are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Product** | **Comment** | **Delivery Date** |
| <Planning> | <Master test plan> |  |  |
|  | <Test plan for each test level> |  |  |
|  |  |  |  |
| <Management> | <Risk report> |  |  |
|  |  |  |  |
| <Setting up and maintenance infrastructure> | <Detail specification test environment> |  |  |
|  |  |  |  |
| <Preparation> | <Report detail intake for each test level> |  |  |
|  |  |  |  |
| <Specification> | <Test script pretest> |  |  |
|  | <Test script for each test level> |  |  |
|  | <Test script for each test level> |  |  |
|  |  |  |  |
| <Execution> | <Defect log> |  |  |
|  | <Status report> |  |  |
| <Completion> | <End report> |  |  |
|  | <Release advice (for each test level)> |  |  |

## Review plan

List the deliverables that have to be reviewed by the stakeholders.

<< Take up in this at least the following deliverables:

* Master test plan
* Test plan for each test level
* End report
* Defect log

| **Deliverable** | **Authors** | **Type review** | **Reviewers** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Entrance and exit criteria for each test level

<< Note: Only a limited number of test levels have been given as an example. All test levels from the test strategy have to be detailed out here. >>

### << Optional: Functional Acceptance Test >>

For the phase Specification and Execution the following entrance criteria are defined:

Entrance criteria for Specification phase:

Entrance criteria for Execution phase:

The following exit criteria are defined for the FAT:

### << Optional: User Acceptance Test >>

For the phase Specification and Execution the following entrance criteria are defined:

Entrance criteria for Specification phase:

Entrance criteria for Execution phase:

The following exit criteria are defined for the UAT:

## Go/No go

<< Describe here the Go/No decision process, for example:

There will be a release advice written after the finishing of the total test project. This summarizes the results of the several test levels and, based on that, what risks are identified when the <name system> is deployed to production. This also details if the acceptance criteria for each test level are met.

Based on this release advice the stakeholders can take a Go/No decision.

For a Go there should be no open defects that have the impact that the system is no longer useable for the user organization or is not manageable by the functional and technical administrators.

The mentioned norms for defects that are preventing the system going into production is not measurable before hand. However the weight and impact of the defects shall be determined during the defect management process. This will be done by the test manager in close cooperation with the stakeholders. Defects that are preventing the system from going into production will be marked as such in the defects tracking system. They will be addressed in the decision process as described earlier. >>

# Organization

Organization<< See TMap® Next 5.2.8 and 8.6.5 >>

## Organization structure

<< Provide the organization chart of the test organization and the relation to the project organization. >>

## Roles, tasks and responsibilities

Describe for each role the tasks and the responsibilities.

<< Describe for each role which department/who will fill these positions, for how many hours per week and during what period. >>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Department / Name employee(s)** | **# hours**  **per week** | **Period** | **Description of tasks and**  **responsibilities** |
| <Test manager> |  |  |  | <-Write MTP  -Coordinate overall test process> |
| <Test coordinator> |  |  |  | <-Write test plan  -Coordinate test> |
| <Tester> |  |  |  | <-Make test specifications  -Execute (re)tests> |
| <Optional: Functional administrators> |  |  |  | <Support testing> |

<< The table above is just an example. Distinguish the overall level and for each test level. Add extra resources and specify for each role the specific tasks and responsibilities. All involved in the test approach need to be present in this table. Another option is to use a RACIS model. >>

<< Optional: Describe for each role not only the tasks and responsibilities, but also the authorizations. “Without authorizations no responsibilities”. >>

### << Optional: Trainings- and coaching’s necessity >>

<< Describe the trainings- and coaching needs that is required to obtain the right subject and/or testing knowledge. >>

## Structure of meetings

Mention all types of meetings within the test project, the objective of the meeting, the frequency and who needs to be present.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Goal** | **Frequency** | **Who** |
| <Project meeting> | <Discuss overall project progress > | <Weekly> | <Project manager  Test manager> |
| <Progress meeting for each test level> | < Discuss progress for each test level> | <Weekly> | <Test manager  Testers> |
| <defect triage> | <Discuss and prioritize defects found during test> | <Weekly> | <Test manager – owns>  <Technical lead>  <Project Manager>  <Business lead> |

<< This is just an example. Add additional meetings as needed. >>

## Structure of reporting

Mention all types of written communication.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Goal** | **Frequency** | **Who** |
| <Risk report> | <Gives insight in the recognized risks of the tests> | <Ad-hoc> | <Test team> |
| <Release advice> | <Gives advice about quality/risks of the implementation of the test object> | <Once-only> | <Project manager> |
| <End report> | <Gives insight in the evaluation of the test process and test object> | <Once-only> | <Project manager> |
| <Progress report> | <Gives insight in the progress of thee tests and quality/risks of the test object> | <Weekly> | <Test manager to key stakeholders> |
| <Defect reports> | <Gives insight to defects and their status’. > | <Weekly> | <Test manager to key stakeholders> |

## Completion

This describes the procedures for the completion process at the end of the project.

# Infrastructure

<< See TMap® Next 5.2.9, 6.4 and 7.3.3.

An important starting point is that different test levels are executed in separate test environments, like a DTAP model. This separation is made to facilitate the different demands of the test levels; a performance test has different demands than a usability test. In a test project not only one environment has to be described, but multiple!

Refer for a further elaboration to the detail test plans and for the infrastructure planning to chapter 16. See for checklists http://eng.tmap.net/Home/TMap/Downloads/index.jsp.>>

## Test environments

<< Describe the demands that each 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.

Mention when each environment has to de delivered, and for how long this environment is needed. Refer to §14.2 where the management of the test infrastructure is described. >>

| **Test level** | **Environment** | **Requirements** | **From** | **To** |
| --- | --- | --- | --- | --- |
| <ST> |  |  |  |  |
| <FAT> |  |  |  |  |
| <UAT> |  |  |  |  |

## Test tools

<< Describe the test tools that have to be used in the test project, for example tools to test services or to measure performance. Mention when (test level, test phase, dates) the tools are needed. Also mention the costs and take this up in the budget if it is part of the test project. Refer to §14.2 where the management of the test infrastructure is described. Describe for each test level which test tools are needed. See <http://eng.tmap.net/Home/TMap/Downloads/index.jsp> for checklists http://eng.tmap.net/Home/TMap/Downloads/index.jsp. >>

|  |  |  |
| --- | --- | --- |
| **Test level** | **Test tool** | **Comment** |
| <ST> | <hardware>  <software>  <means of communication>  <facilities for the build and use of files>  <procedures> |  |
|  |  |  |

## Office setup

<< Every test level requires a specific office setup. Describe the minimal requirements. A detailed description follows in the detailed plans. In some cases it is necessary to do parallel testing with many users. A specific office environment might need to be created for that. Availability of fax machine or printers can be written down here as well. Any office needs will be mentioned here.

>>

|  |  |  |
| --- | --- | --- |
| **Test level** | **Components** | **Comment** |
| <ST> |  |  |
| <…> |  |  |

# Management

<< See TMap® Next, 5.2.10 and 6.2.12. >>

## Test process management

The management of the test process can be divided into three parts:

* Progress and expenditure of budget and time: the management of the planning and guarding of the progress in terms of time, resources and means. This has been arranged as followed: < short description >;
* Quality indicators: the aim of testing is to provide information and advice on the risks and quality of the object to be tested. To be able to provide this information, quality indicators are registered. This has been arranged as followed: < short description >.
* Test statistics: the test manager builds statistics based on the above information. Statistics can supply insight into the progress of the test process and quality of the test object, including any trends. This has been arranged as followed: < short description >.

## Test infrastructure management

<< Describe how and by whom the test infrastructure is managed:

* Procedure office setup
* Procedure test tool
* Procedure test environment

Describe for each system part (platform, database, software, etc.) what authorizations are needed for the test team. During which period they are needed, and who’s responsible. See TMap® Next 6.4.5. >>

## Test product management

<< Describe how the test products, conventions and standards, will be maintained and guarded. Do this using procedures, templates and tools. Consider:

* Products like testware and test project documents
* External products like the test basis and object under test >>

## Defects procedure

The defects management has been arranged in conformity with the defect procedure that is described in TMap® Next 12.4., or in conformity with defect procedure as it is used within the organization. For the registration and maintenance of defects the following tool is being used: < tool >.

The responsibility for the observance of this defects procedure lies with the <defect administrator>.

<< Diagram TMap® Next p. 568. >>

# Test process risks and countermeasures

This chapter makes an inventory of the most important potential project risks for the testing of <Name project>. By anticipating what possibly might occur, it’s possible to mitigate the risk by taking the appropriate countermeasures. The risks apply directly to the test process, or apply to risks that can be of direct consequence for the test project. Registration and monitoring of these risks continues after the MTP has been written, it is a continuous process.

The following risks have been recognized for the test process. See also <name risk log>.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nr** | **Risk Event** | **Consequence** | **Impact** | **Chance** | **Score** | **Countermeasures** | **Owner** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

The test manager is aware of these points and monitors the countermeasures.

<< The countermeasures that can’t be taken within the test organization can be discussed and assigned owners during the project meeting. This will be recorded in the action list and is monitored during the project meeting. See TMap® Next, 5.2.11. >>

# Global Estimation & Planning

<< See TMap® Next 5.2.5 and 5.2.6>>

## Estimation

The estimation is as follows: << The estimation divided in personal and infrastructural costs. >>

| **Test level** | **Who** | **P** | **C** | **I** | **P** | **S** | **E** | **C** | **Totals** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Overall | Test manager |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| ST | Test coordinator |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| FAT | Test coordinator |  |  |  |  |  |  |  |  |
|  | End users |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| UAT | Test coordinator |  |  |  |  |  |  |  |  |
|  | End users |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Totals:** | |  |  |  |  |  |  |  |  |

<< Preferably make a foundation for the estimation, elaborating on the used estimation technique and the basis for the estimation, for example the estimation of the development team. Describe, as much as possible, for each test level the planning for the infrastructure.

This table is an example. **Attention!** All resources mentioned with the test approach need to be listed here. >>

This estimation will be divided in sub activities in the detail test plans for each test level.

## Planning

<< The global planning has to consist at least of the following parts:

* The activities to be executed (phases for each test level)
* Relations with and dependencies with other activities (inside or outside the test process and between the test levels)
* The amount of time for each test level
* Needed and available resources (organization and infrastructure)
* Needed and available duration>>



<< The content of the table is an example. >>

The activities to be executed are in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Name** | **Start Date** | **End Date** | **Duration** | **Relations** |
|  |  |  |  |  |  |

## Milestones

The milestones of the test process of <system> are detailed in the table below.

|  |  |
| --- | --- |
| **Mile stone description** | **Date** |
|  |  |

<< Think of the defined products from §5.5 >>

# Glossary

|  |  |
| --- | --- |
| PRA | Product risk analysis; analyzing the product under test with the goal that the test manager and the other stakeholders achieve a joint view of what the more and less risky parts and characteristics of the system are. This with the purpose to relate the thoroughness of testing to it. |
| ST | System test, by the vendor of the solution in a (good controllable) laboratory environment executed test, which has to demonstrate that the developed system or parts of it comply with the functional and non functional specifications and the technical design. |
| UT | Unit test, by the developer in the development environment executed test, which has to demonstrate that a unit complies with the technical specifications. |
| DTAP | Development, Test, Acceptance and Production environment in a so called following, logical ‘street’. |

**<<Attention**: The list is only an indication and has to be checked for completeness!>>