Team Orange Master Test Plan

Version information

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Author** |
| 1.0 | 5/9/2017 | Initial Draft | Murray Neish |
|  |  |  |  |

Distribution list

|  |  |
| --- | --- |
| **Name** | **Company/Function** |
| Team Orange | Development Team |
| Jim Tulip | Project Manager |

Approval Client

|  |  |  |
| --- | --- | --- |
| Client: | | Signature |
| Name | Pinnacle Pty Ltd |  |
| Division | Executive |  |
| Department | Corporate Services |  |
| Function | Corporate Administration |  |
| Location | Santa Clara CA |  |
| Telephone | 1-888-847-8766. |  |
| E-Mail address | corporate@pinnaclecorp.com | Date: |

Management summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Project objective**  To deliver an in-house instant messaging platform, for which Pinnacle has full control of the source code and the ability to customize the platform to its unique requirements. | | | |
| **Test objective and assignment**  To exercise the system functions at both the component level and to varying degrees of integration up to and including user acceptance testing. | | | |
| **Short description of the test approach**   * Bespoke testing of various libraries and functions * Unit testing/mocking frameworks to exercise the system components and provide automated testing for interative develops | | | |
| **Results to be realized** | | | |
| *Result*   * well executed and finished system test * well executed and finished user acceptance test * well executed and finished testing summary (incorporating the above two items) | | *Document*   * ST Test report * UAT Test report * End report Testing | *Delivery date* |
| **Qualitative objectives**  Each test level (including building of the test) needs to be completed during the relevant iteration in order to provide feedback to the team on completed functionality and to ensure that changes to the system preserve the existing functionality. Test methodology needs to be clearly linked to use cases and work items. Test outcomes must unambiguously demonstrate satisfaction of the relevant acceptance criteria. Where possible, test driven/test oriented development methodology will be pursued in order to write easily testable code. | | | |
| **Estimate** | | | |
| **Test process risks and measures** | | | |
| *Test process risks*  • Testing unable to fully exercise the code base due to complex dependencies | *Measures to be taken*  • Developmental approach using SOLID principles. For particularly complex interactions, UAT/full system testing may be the only viable option | | |
|  |  | | |
| **Go/no-go decisions**  Each developer will be responsible for completion of tests relative to his specific feature(s) under development, and documenting whether these tests pass or fail. The development of these tests may be delegated to other team members based on workload/skillset. Test reporting will be a lightweight process with minimal ceremony, comments will be provided where a simple pass/fail doesn’t fully convey the outcome of the test.  Unit tests will be run as part of the build process to ensure that new features/functionality does not break existing functionality. In the case that major difficulties emerge due to architectural issues, incompatibility with API etc., the developer who finds the problem will alert the team via the team instant messaging platform. The project manager will be informed if any adjustment needs to be made to the project schedule because of the findings. At the team iteration meeting the resolution or ongoing actions relating to the issue will be presented to the project manager. The team will discuss if it is possible to progress to new work items or concentrate on resolving the outstanding work item(s) first. | | | |

Table of Contents

1 Introduction [1](#__RefHeading___Toc205713830)

1.1 Project and project objective [1](#__RefHeading___Toc205713831)

1.2 Objective of the master test plan [1](#__RefHeading___Toc205713832)

1.3 Involved in creating the master test plan [1](#__RefHeading___Toc205713833)

2 Assignment formulation [2](#__RefHeading___Toc205713834)

2.1 Client [2](#__RefHeading___Toc205713835)

2.2 Supplier [2](#__RefHeading___Toc205713836)

2.3 Assignment [2](#__RefHeading___Toc205713837)

2.4 Scope [2](#__RefHeading___Toc205713838)

2.4.1 Within scope [2](#__RefHeading___Toc205713839)

2.4.2 Out of scope [2](#__RefHeading___Toc205713840)

2.5 Preconditions and assumptions [3](#__RefHeading___Toc205713841)

2.6 Acceptants and acceptance criteria [3](#__RefHeading___Toc205713842)

2.6.1 Acceptants [4](#__RefHeading___Toc205713843)

2.6.2 Acceptation criteria [4](#__RefHeading___Toc205713844)

3 DocumentatiON [5](#__RefHeading___Toc205713845)

3.1 Basis for the master test plan [5](#__RefHeading___Toc205713846)

3.2 Standards [5](#__RefHeading___Toc205713847)

3.3 Test basis [5](#__RefHeading___Toc205713848)

4 Test strategy [6](#__RefHeading___Toc205713849)

4.1 Product risk analyses [6](#__RefHeading___Toc205713850)

4.2 Test strategy [7](#__RefHeading___Toc205713851)

5 Approach [9](#__RefHeading___Toc205713852)

5.1 Test levels [9](#__RefHeading___Toc205713853)

5.2 Evaluation [9](#__RefHeading___Toc205713854)

5.3 The <name test level> [9](#__RefHeading___Toc205713855)

5.3.1 Goal [9](#__RefHeading___Toc205713856)

5.3.2 Short description [9](#__RefHeading___Toc205713857)

5.3.3 Responsible [9](#__RefHeading___Toc205713858)

5.3.4 << Optional: test environment to be used >> [10](#__RefHeading___Toc205713859)

5.4 Phasing per test level [10](#__RefHeading___Toc205713860)

5.5 Test products [11](#__RefHeading___Toc205713861)

5.6 Review plan [11](#__RefHeading___Toc205713862)

5.7 Entrance and exit criteria for each test level [12](#__RefHeading___Toc205713863)

5.7.1 << Optional: Functional Acceptance Test >> [12](#__RefHeading___Toc205713864)

5.7.2 << Optional: User Acceptance Test >> [12](#__RefHeading___Toc205713865)

5.8 Go/No go [12](#__RefHeading___Toc205713866)

6 Organization [14](#__RefHeading___Toc205713867)

6.1 Organization structure [14](#__RefHeading___Toc205713868)

6.2 Roles, tasks and responsibilities [14](#__RefHeading___Toc205713869)

6.2.1 << Optional: Trainings- and coaching’s necessity >> [14](#__RefHeading___Toc205713870)

6.3 Structure of meetings [14](#__RefHeading___Toc205713871)

6.4 Structure of reporting [15](#__RefHeading___Toc205713872)

6.5 Completion [15](#__RefHeading___Toc205713873)

7 Infrastructure [16](#__RefHeading___Toc205713874)

7.1 Test environments [16](#__RefHeading___Toc205713875)

7.2 Test tools [16](#__RefHeading___Toc205713876)

7.3 Office setup [17](#__RefHeading___Toc205713877)

8 Management [18](#__RefHeading___Toc205713878)

8.1 Test process management [18](#__RefHeading___Toc205713879)

8.2 Test infrastructure management [18](#__RefHeading___Toc205713880)

8.3 Test product management [18](#__RefHeading___Toc205713881)

8.4 Defects procedure [18](#__RefHeading___Toc205713882)

9 Test process risks and countermeasures [19](#__RefHeading___Toc205713883)

10 Global Estimation & Planning [20](#__RefHeading___Toc205713884)

10.1 Estimation [20](#__RefHeading___Toc205713885)

10.9 Planning [20](#__RefHeading___Toc205713886)

10.10 Milestones [21](#__RefHeading___Toc205713887)

11 Glossary [22](#__RefHeading___Toc205713888)

# Introduction

## Project and project objective

Pinnacle Pty Ltd have commissioned Team Orange to develop an instant messaging platform “Confide” for corporate use with their company. Pinnacle is an investment company which specialises in providing management of superannuation products for high net worth individuals.

Pinnacle wish to develop a privately hosted instant messaging client-server application. The company has requested full control of and rights to the source code. The messaging client will be developed in accordance with Pinnacle’s specific requirements.

This master test plan fits to the project plan *ProjectPlan.docx*

## 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 *Confide Instant Messaging Platform*.

The master test plan describes this approach, the activities and (end) products that need further elaboration in the other iteration 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** |
| Murray Neish | Developer | Write MTP |
| Shaun Kozanic | Developer | Review MTP |
| Tim Bell | Developer | Review MTP |
| Jim Tulip | Project Manager | Approve MTP |

# Assignment formulation

## Client

The client is Pinnacle Corporation LLC

## Supplier

The supplier is Team Orange Pty Ltd

## Assignment

To deliver the test items as agreed in the Confide Instant Messaging System Scope of Works.

## Scope

### Within scope

#### API testing (included Open Source libraries leveraged by the project)

Babbler XMPP library has been chosen for this project. Functions being provided by Babbler which the project code relies on will be unit tested.

#### Systems – infrastructure which supports the functionality being provided. This applies to the functional requirements only.

OpenFire XMPP server has been chosen. Server testing will be in the context of verifying that client actions have been successfully received by the server.

#### Interfaces

Confide Instant Messaging Application follows a MVC architecture. Integration testing will be conducted to confirm that project code is successfully interfacing with the project library.

#### User acceptance testing of UI elements

Tests will be conducted with a representative user group to confirm that the UI is straightforward and efficient to use.

### Out of scope

* System changes not included in the project (e.g. hardware/TOS changes to the server hosting platform);
* Non-functional requirements related to the hosting platform

## Preconditions and assumptions

### Preconditions

No pre-conditions have been identified. Project is green-field without internal or external dependencies.

### Assumptions

* 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.
* The test environments are delivered on time and correctly working in conformity with the environment plan and specification.
* Test specification for a test level will start no earlier than when the entrance criteria for it are met
* Test execution for a test level will start no earlier than when the entrance criteria for it are met

### Verifiers

The table below states the verifiers of the Confide Instant Manager System:

|  |  |  |
| --- | --- | --- |
| **Name** | **Function** | **Department** |
| **Joe Warren** | Pinnacle CEO | Executive |
| **Maggie Franks** | Communications Manager | Executive |

### Acceptance criteria

The table below states which acceptance criteria there are for the Confide Instant Manager 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.

## Basis for the master test plan

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| [ProjectPlan.docx](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration2/ProjectPlan.docx) | 1.0 |  | Team Orange |
| [Project Proposal.odt](https://github.com/csuproject/instantmessenger/blob/master/docs/ProjectInitiation/ProjectProposal.odt) | 1.0 |  | Team Orange |
| [Project Vision](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration1/ProjectVision.docx) | 1.0 |  | Team Orange |
| [Use Case Revision](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration1/UseCaseRevision.docx) | 1.0 |  | Team Orange |

## Standards

No specific standards have been applied for this testing regime.

## 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| [Testplan1.docx](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration2/TestPlan1.docx) | 1.0 |  | Team Orange |
| [Testplan2.docx](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration2/TestPlan2.docx) | 1.0 |  | Team Orange |
| [Testplan3.docx](https://github.com/csuproject/instantmessenger/blob/master/docs/InceptionIteration2/TestPlan3.docx) |  |  |  |

Ongoing testing documentation will be added to this table as the information becomes available. Test results will also be recorded in these test basis documents.

# Test strategy

The time available for testing is limited; not everything can be tested with equal thoroughness. This means that choices must be made regarding the depth of testing. The test strategy is designed to be as effective and efficient as possible over the total test project. These principles are the basis of our test strategy.

## 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.

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** |
| Technology stack meets critical use case requirements | Verify that system satisfies the minimum agreed critical use cases | Functional use cases |
| Product is easy to use | User acceptance testing of final system | Non-functional requirements |
|  |  |  |

Pinnacle LLC in conjunction with the Project Manager and Team Orange have determined the product risks. The extent of the risk (the risk class) is dependent on the chance of failure the damage to the organisation if it 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.

#### Risk table

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Part** | **RC** | **Comments** |
| Functionality |  | A | Product must meet minimum functional requirements |
| User-friendliness |  | B | Product must be efficient and easy to use |
| Performance |  | C | Technology is demonstrated to operate at scale. Risk is low of poor performance. |
| Security |  | A |  |

## 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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

## Test levels

For this MTP the following test levels are acknowledged:

|  |  |
| --- | --- |
| **Test level** | **Goal** |
| API | Confirm that dependencies in use perform as expected |
| Integration | Project code which leverages API code performs the intended functions |
| System | Project team confirms main use cases are satisfied |
| UAT | Alpha/Beta testing of the software with the aim to deploy into production |

## Evaluation

## 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. >>

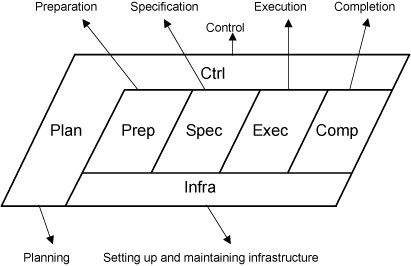
### Responsible

The Team Orange development team have assumed responsibility for the testing

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

<< The <test level> will be executed on the <test environment>. This will be elaborated in chapter 7. >>

## 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

* 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 10. 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 §8.2 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 §8.2 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!>>