Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Intended Audience 3

1.4 Document Terminology and Acronyms 3

1.5 References 3

1.6 Document Structure 3

2. Evaluation Mission and Test Motivation 3

2.1 Background 3

2.2 Evaluation Mission 3

2.3 Test Motivators 3

3. Target Test Items 3

4. Outline of Planned Tests 3

4.1 Outline of Test Inclusions 3

4.2 Outline of other candidates for potential inclusion 3

4.3 Outline of Test Exclusions 3

5. Test Approach 3

5.1 Initial Test-Idea Catalogs and other reference sources 3

5.2 Testing Techniques and Types 3

5.2.1 Data and Database Integrity Testing 3

5.2.2 Function Testing 3

5.2.3 Business Cycle Testing 3

5.2.4 User Interface Testing 3

5.2.5 Performance Profiling 3

5.2.6 Load Testing 3

5.2.7 Stress Testing 3

5.2.8 Volume Testing 3

5.2.9 Security and Access Control Testing 3

5.2.10 Failover and Recovery Testing 3

5.2.11 Configuration Testing 3

5.2.12 Installation Testing 3

6. Entry and Exit Criteria 3

6.1 Test Plan 3

6.1.1 Test Plan Entry Criteria 3

6.1.2 Test Plan Exit Criteria 3

6.1.3 Suspension and resumption criteria 3

6.2 Test Cycles 3

6.2.1 Test Cycle Entry Criteria 3

6.2.2 Test Cycle Exit Criteria 3

6.2.3 Test Cycle abnormal termination 3

7. Deliverables 3

7.1 Test Evaluation Summaries 3

7.2 Reporting on Test Coverage 3

7.3 Perceived Quality Reports 3

7.4 Incident Logs and Change Requests 3

7.5 Smoke Test Suite and supporting Test Scripts 3

7.6 Additional work products 3

7.6.1 Detailed Test Results 3

7.6.2 Additional automated functional Test Scripts 3

7.6.3 Test Guidelines 3

7.6.4 Traceability Matrices 3

8. Testing Workflow 3

9. Environmental Needs 3

9.1 Base System Hardware 3

9.2 Base Software Elements in the Test Environment 3

9.3 Productivity and Support Tools 3

9.4 Test Environment Configurations 3

10. Responsibilities, Staffing and Training Needs 3

10.1 People and Roles 3

10.2 Staffing and Training Needs 3

11. Iteration Milestones 3

12. Risks, Dependencies, Assumptions and Constraints 3

13. Management Process and Procedures 3

13.1 Measuring and Assessing the Extent of Testing 3

13.2 Assessing the deliverables of this Test Plan 3

13.3 Problem Reporting, Escalation and Issue Resolution 3

13.4 Managing Test Cycles 3

13.5 Traceability Strategies 3

13.6 Approval and Signoff 3

<Iteration/ Master> Test Plan

# Introduction

## Purpose

The purpose of the Iteration Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This *Test Plan* for the <Project Name> supports the following objectives:

* • [Identifies the items that should be targeted by the tests.
* • Identifies the motivation for and ideas behind the test areas to be covered.
* • Outlines the testing approach that will be used.
* • Identifies the required resources and provides an estimate of the test efforts.
* • Lists the deliverable elements of the test project.]

## Scope

* [Describe the levels of testing⎯for example, Unit, Integration, or System⎯and the types of testing⎯such as Functionality, Usability, Reliability, Performance, and Supportability⎯that will be addressed by this **Test Plan**. It is also important to provide a general indication of significant areas that will be **excluded** from scope, especially where the intended audience might otherwise reasonably assume the inclusion of those areas.
* **Note**: Avoid placing detail here that you will repeat in sections 3, Target Test Items, and 4,Outline of Planned Tests.]

## Intended Audience

* [Provide a brief description of the audience for whom you are writing the **Test Plan**. This helps readers of your document identify whether it is a document intended for their use, and helps prevent the document from being used inappropriately.
* **Note**: Document style and content often alters in relation to the intended audience.
* This section should only be about three to five paragraphs in length.]

## Document Terminology and Acronyms

* [This subsection provides the definitions of any terms, acronyms, and abbreviations required to properly interpret the **Test Plan**. Avoid listing items that are generally applicable to the project as a whole and that are already defined in the project’s Glossary. Include a reference to the project’s Glossary in the References section.]

## References

* [This subsection provides a list of the documents referenced elsewhere within the **Test Plan**. Identify each document by title, version (or report number if applicable), date, and publishing organization or original author. Avoid listing documents that are influential but not directly referenced. Specify the sources from which the “official versions” of the references can be obtained, such as intranet UNC names or document reference codes. This information may be provided by reference to an appendix or to another document.]

## Document Structure

* [This subsection outlines what the rest of the **Test Plan** contains and gives an introduction to how the rest of the document is organized. This section may be eliminated if a Table of Contents is used.]

# Evaluation Mission and Test Motivation

* [Provide an overview of the mission and motivation for the testing that will be conducted in this iteration.]

## Background

* [Provide a brief description of the background surrounding why the test effort defined by this **Test Plan** will be undertaken. Include information such as the key problem being solved, the major benefits of the solution, the planned architecture of the solution, and a brief history of the project. Where this information is defined in other documents, you can include references to those other more detailed documents if appropriate. This section should only be about three to five paragraphs in length.]

## Evaluation Mission

* [Provide a brief statement that defines the mission for the evaluation effort in the current iteration. This statement might incorporate one or more concerns including:
* find as many bugs as possible
* find important problems, assess perceived quality risks
* advise about perceived project risks
* certify to a standard
* verify a specification (requirements, design or claims)
* advise about product quality, satisfy stakeholders
* advise about testing
* fulfill process mandates
* and so forth
* Each mission provides a different context to the test effort and alters the way in which testing should be approached.]

## Test Motivators

* [Provide an outline of the key elements that will motivate the testing effort in this iteration. Testing will be motivated by many things⎯quality risks, technical risks, project risks, use cases, functional requirements, non-functional requirements, design elements, suspected failures or faults, change requests, and so forth.]

# Target Test Items

The listing below identifies those test items⎯software, hardware, and supporting product elements ⎯that have been identified as targets for testing. This list represents what items will be tested.

* [Provide a high level list of the major target test items. This list should include both items produced directly by the project development team, and items that those products rely on; for example, basic processor hardware, peripheral devices, operating systems, third-party products or components, and so forth. Consider grouping the list by category and assigning relative importance to each motivator.]

# Outline of Planned Tests

* [This section provides a high-level outline of the testing that will be performed. The outline in this section represents a high level overview of both the tests that will be performed and those that will not.]

## Outline of Test Inclusions

* [Provide a high level outline of the major testing planned for the current iteration. Note what will be included in the plan and record what will explicitly **not** be included in the section titled Outline of Test Exclusions.]

## Outline of Other Candidates for Potential Inclusion

* [Separately outline test areas you suspect might be useful to investigate and evaluate, but that have not been sufficiently researched to know if they are important to pursue.]

## Outline of Test Exclusions

* [Provide a high level outline of the potential tests that might have been conducted but that have been **explicitly excluded** from this plan. If a type of test will not be implemented and executed, indicate this in a sentence stating the test will not be implemented or executed and stating the justification, such as:
* “These tests do not help achieve the evaluation mission.”
* “There are insufficient resources to conduct these tests.”
* “These tests are unnecessary due to the testing conducted by xxxx.”
* As a heuristic, if you think it would be reasonable for one of your audience members to expect a certain aspect of testing to be included that you will not or cannot address, you should note it’s exclusion: If the team agrees the exclusion is obvious, you probably don’t need to list it.]

# Test Approach

* [The Test Approach presents the recommended strategy for designing and implementing the required tests. Sections 3, Target Test Items, and 4, Outline of Planned Tests, identified **what** items will be tested and **what** types of tests would be performed. This section describes **how** the tests will be realized.
* One aspect to consider for the test approach is the techniques to be used. This should include an outline of how each technique can be implemented, both from a manual and/or an automated perspective, and the criterion for knowing that the technique is useful and successful. For each technique, provide a description of the technique and define why it is an important part of the test approach by briefly outlining how it helps achieve the Evaluation Mission or addresses the Test Motivators.
* Another aspect to discuss in this section is the Fault or Failure models that are applicable and ways to approach evaluating them.
* As you define each aspect of the approach, you should update Section 10, Responsibilities, Staffing, and Training Needs, to document the test environment configuration and other resources that will be needed to implement each aspect.]

## Initial Test-Idea Catalogs and Other Reference Sources

* [Provide a listing of existing resources that will be referenced to stimulate the identification and selection of specific tests to be conducted. An example Test-Ideas Catalog is provided in the examples section of RUP.]

## Testing Techniques and Types

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | * Test auf die vom Kunden gewünschten Funktionen, insbesondere die Funktionalität der Oberfläche |
| Technique: | * Nutzeraktionen (Mausklicks, Tastatureingaben) werden durch ein Tool emuliert und die danach vorherrschende Oberfläche mit einem definierten Soll-Zustand vergleichen |
| Oracles: | * Der Erfolg des Tests ist gegeben, wenn die grafische Oberfläche einen bestimmten Zustand eingenommen hat. Der Eintritt dieses Zustands kann festgestellt werden in dem man automatisiert auf die Existenz von GUI-Komponenten (Schaltflächen, Bilder, Eingabefelder) prüft |
| Required Tools: | * Calabash-Android (basiert auf Cucumber, Gherkin für Featurdefinitionen in der Domänensprache und Ruby für Step-Definitionen)   Android Emulator |
| Success Criteria: | * Alle Features müssen erfolgreich sein (Ausnahme siehe Special Considerations) |
| Special Considerations: | * Einige Benutzeraktionen können nicht mit der verwendeten Calabash-Version getestet werden (zum Beispiel: Popups oder Listenelemente) * Alle Feature-Definitionen wurden auf Deutsch geschrieben, da es die Muttersprache aller Stakeholder ist. |

### User Interface Testing

Siehe Function Testing

### Unit Testing

|  |  |
| --- | --- |
| Technique Objective: | * Test der einzelnen Module (Methoden) |
| Technique: | * Methoden werden unabhängig voneinander in speziellen Testklassen ausgeführt und ihre Ergebnisse mit vordefinierten Angaben verglichen und damit auf Korrektheit geprüft |
| Oracles: | * Der Erfolg ist gegeben, wenn die aufgerufenen Assert-Methoden keine Exception werfen, enden und“ true“ zurückgeben. |
| Required Tools: | * JUnit |
| Success Criteria: | * Alle Unit-Tests müssen erfolgreich sein. |
| Special Considerations: | * In Android können keine Dialoge getestet werden (als Issue in der offiziellen Dokumentation von Android belegt). |

# Entry and Exit Criteria

* Not applicable

# Deliverables

## Test Evaluation Summaries

* Not applicable

## Reporting on Test Coverage

* am Ende jedes Sprints werden alle Tests durchgeführt und die entsprechenden Log-Dateien erstellt
* mit Ant und Emma werden Log-Dateien der Unit-Tests in Form von fertig formatiertem HTML erstellt
* Calabash generiert einen Report in der Konsole, der manuell in eine Text-Datei kopiert wird

## Perceived Quality Reports

* Not applicable

## Incident Logs and Change Requests

* Not applicable

## Smoke Test Suite and Supporting Test Scripts

* Not applicable

## Additional Work Products

* Not applicable

# Testing Workflow

* Calabash-Tests am Ende der Implementierung jedes Use-Cases
* Unit-Tests nachdem eine Methode implementiert ist
* für jeden Use-Case sind beide Test-Arten fester Bestandteil der Projektplanung
* beide Test-Arten sind für den erfolgreichen Abschluss des Use-Cases erforderlich

# Environmental Needs

## Base System Hardware

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Tablet | 1 | Samsung Tab 2 7.0 (Android OS) |
| Smartphone | 1 | Samsung Galaxy S3 (Android OS) |

## Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| Android Emulator |  | Virtuelle Maschine |
| JUnit (integriert in IDE) |  | Test-Framework für Unit-Tests |
| Emma |  | Test Coverage Tool |
| Ant |  | Library für Automatic Build |
| Calabash-Android |  | Automated Functional Test Tool |

## Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| Test Management | JIRA |  |  |
| Defect Tracking | JUnit, Calabash |  |  |
| Project Management | JIRA, MS Project |  |  |

## Test Environment Configurations

* Not applicable

# Responsibilities, Staffing, and Training Needs

* [This section presents the required resources to address the test effort outlined in the **Test Plan**—the main responsibilities, and the knowledge or skill sets required of those resources.]

## People and Roles

This table shows the staffing assumptions for the test effort.

* [**Note**: Add or delete items as appropriate.]

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager |  | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst |  | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer |  | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester |  | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator |  | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager |  | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer |  | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer |  | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

## Staffing and Training Needs

This section outlines how to approach staffing and training the test roles for the project.

[The way to approach staffing and training will vary from project to project. If this section is part of a Master Test Plan, you should indicate at what points in the project lifecycle different skills and numbers of staff are needed. If this is an Iteration Test Plan, you should focus mainly on where and what training might occur during the Iteration.

Give thought to your training needs, and plan to schedule this based on a Just-In-Time (JIT) approach—there is often a temptation to attend training too far in advance of its usage when the test team has apparent slack. Doing this introduces the risk of the training being forgotten by the time it's needed.

Look for opportunities to combine the purchase of productivity tools with training on those tools, and arrange with the vendor to delay delivery of the training until just before you need it. If you have enough headcount, consider having training delivered in a customized manner for you, possibly at your own site.

The test team often requires the support and skills of other team members not directly part of the test team. Make sure you arrange in your plan for appropriate availability of System Administrators, Database Administrators, and Developers who are required to enable the test effort.]

# Iteration Milestones

* [Identify the key schedule milestones that set the context for the Testing effort. Avoid repeating too much detail that is documented elsewhere in plans that address the entire project.]

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed |  |  |  |  |
| Iteration starts |  |  |  |  |
| Requirements baselined |  |  |  |  |
| Architecture baselined |  |  |  |  |
| User Interface baselined |  |  |  |  |
| First Build delivered to test |  |  |  |  |
| First Build accepted into test |  |  |  |  |
| First Build test cycle finishes |  |  |  |  |
| [Build Two will not be tested] |  |  |  |  |
| Third Build delivered to test |  |  |  |  |
| Third Build accepted into test |  |  |  |  |
| Third Build test cycle finishes |  |  |  |  |
| Fourth Build delivered to test |  |  |  |  |
| Fourth Build accepted into test |  |  |  |  |
| Iteration Assessment review |  |  |  |  |
| Iteration ends |  |  |  |  |

# Risks, Dependencies, Assumptions, and Constraints

* [List any risks that may affect the successful execution of this **Test Plan**, and identify mitigation and contingency strategies for each risk. Also indicate a relative ranking for both the likelihood of occurrence and the impact if the risk is realized.]

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Prerequisite entry criteria is not met. | <Tester> will define the prerequisites that must be met before Load Testing can start.  <Customer> will endeavor to meet prerequisites indicated by <Tester>. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | <Customer> will ensure a full set of suitable and protected test data is available.  <Tester> will indicate what is required and will verify the suitability of test data. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | <System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>. | * Restore data and restart * Clear Database |

* [List any dependencies identified during the development of this **Test Plan** that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities You should consider responsibilities you are relying on other teams or staff members external to the test effort completing, timing and dependencies of other planned tasks, the reliance on certain work products being produced.]

| **Dependency between** | **Potential Impact of Dependency** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

* [List any assumptions made during the development of this **Test Plan** that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, and so forth].

| **Assumption to be proven** | **Impact of Assumption being incorrect** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

* [List any constraints placed on the test effort that have had a negative effect on the way in which this **Test Plan** has been approached.]

| **Constraint on** | **Impact Constraint has on test effort** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Management Process and Procedures

* [Outline what processes and procedures are to be used when issues arise with the **Test Plan** and its enactment.]

## Measuring and Assessing the Extent of Testing

* [Outline the measurement and assessment process to be used to track the extent of testing.]

## Assessing the Deliverables of this Test Plan

* [Outline the assessment process for reviewing and accepting the deliverables of this **Test Plan**]

## Problem Reporting, Escalation, and Issue Resolution

* [Define how process problems will be reported and escalated, and the process to be followed to achieve resolution.]

## Managing Test Cycles

* [Outline the management control process for a test cycle.]

## Traceability Strategies

* [Consider appropriate traceability strategies for:
* Coverage of Testing against Specifications — enables measurement the extent of testing
* Motivations for Testing — enables assessment of relevance of tests to help determine whether to maintain or retire tests
* Software Design Elements — enables tracking of subsequent design changes that would necessitate rerunning tests or retiring them
* Resulting Change Requests — enables the tests that discovered the need for the change to be identified and re-run to verify the change request has been completed successfully]

## Approval and Signoff

* [Outline the approval process and list the job titles (and names of current incumbents) that initially must approve the plan, and sign off on the plans satisfactory execution.]