Project Neco

Version <1.1>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 16/05/2018 | 1.0 | Document created | Alexander R. |
| 04/06/2018 | 1.1 | Document updated | Alexander R. |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Intended Audience 5

1.4 Document Terminology and Acronyms 5

1.5 References 5

1.6 Document Structure 5

2. Evaluation Mission and Test Motivation 5

2.1 Background 5

2.2 Evaluation Mission 5

2.3 Test Motivators 5

3. Target Test Items 6

4. Outline of Planned Tests 6

4.1 Outline of Test Inclusions 6

4.2 Outline of Other Candidates for Potential Inclusion 6

4.3 Outline of Test Exclusions 6

5. Test Approach 6

5.1 Initial Test-Idea Catalogs and Other Reference Sources 6

5.2 Testing Techniques and Types 6

5.2.1 UI Testing (Send Message) 6

5.2.2 Unit Testing 6

5.2.3 Installation Testing 7

6. Deliverables 7

6.1 Test Evaluation Summaries 7

6.2 Reporting on Test Coverage 7

6.3 Perceived Quality Reports 7

6.4 Incident Logs and Change Requests 7

6.5 Smoke Test Suite and Supporting Test Scripts 7

6.6 Additional Work Products 7

6.6.1 Detailed Test Results 7

6.6.2 Additional Automated Functional Test Scripts 7

6.6.3 Test Guidelines 8

6.6.4 Traceability Matrices 8

7. Testing Workflow 8

8. Environmental Needs 8

8.1 Base System Hardware 8

8.2 Base Software Elements in the Test Environment 9

8.3 Productivity and Support Tools 9

8.4 Test Environment Configurations 10

9. Responsibilities, Staffing, and Training Needs 10

9.1 People and Roles 10

9.2 Staffing and Training Needs 13

10. Iteration Milestones 13

11. Risks, Dependencies, Assumptions, and Constraints 14

12. Management Process and Procedures 15

12.1 Measuring and Assessing the Extent of Testing 15

12.2 Assessing the Deliverables of this Test Plan 15

12.3 Problem Reporting, Escalation, and Issue Resolution 15

12.4 Managing Test Cycles 15

12.5 Traceability Strategies 15

12.6 Approval and Signoff 15

# 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 Team Neco supports the following objectives:

* Unit test for the use case “create identity”
* UI test for the use case “send message” and “receive message”

## Scope

Functional Testing

Will blackbox test the application behaviour by using cucumber.

Unit tests

Will test the internal application logic.

## Intended Audience

* Students
* Professors
* Programmers

## Document Terminology and Acronyms

n/a

## References

n/a

## Document Structure

n/a

# Evaluation Mission and Test Motivation

Automated Testing is good for detect coding errors and will show them. Programmer are lazy people and test only things they have changed. To prevent error in the productive run, we must detect error in the development process.

## Background

Because our team includes some programmers, it could happen that while changing, deleting or updating code the already existing code is negatively affected. The automatic testing should discover these.

## Evaluation Mission

Testing is done to provide a stable software. And we will fulfill the goal by the following points.

* find as many bugs as possible
* find important problems
* certify a standard
* verify software specification

## Test Motivators

* Technical risks
* Functional requirements
* Non-functional requirements

# 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.

* Models
* Server
* Encryption logic

# Outline of Planned Tests

## Outline of Test Inclusions

* UI Tests
* Unit Tests

## Outline of Other Candidates for Potential Inclusion

n/a

## Outline of Test Exclusions

n/a

# Test Approach

* UI Tests
* Unit Tests

## Initial Test-Idea Catalogs and Other Reference Sources

## Testing Techniques and Types

### UI Testing (Send Message)

|  |  |
| --- | --- |
| Technique Objective: | Testing the UI for sending a message from the App. |
| Technique: | * Testing the app functionality with Gherkin .feature files |
| Oracles: | The test is successful, if it’s executed and did what was expected |
| Required Tools: | VS Plugin SpecflowVsIntegration  Visual Studio 2017 |
| Success Criteria: | All test are executed successfully |

### Unit Testing

|  |  |
| --- | --- |
| Technique Objective: | Testing the functionality of the code |
| Technique: | Testing the code of the testable classes  Cryptohandler |
| Oracles: | The tests are successful, if all correct answers are running through |
| Required Tools: | Visual Studio 2017 |
| Success Criteria: | All Tests pass |

### Installation Testing

|  |  |
| --- | --- |
| Technique Objective: | Testing the possibility to install the app |
| Technique: | Testing the installation process for simplicity |
| Oracles: | The test user has no trouble with installing the app and the app is running on the testuser’s device |
| Required Tools: | End device with Android 8.1 installed (e.g. smartphone) |
| Success Criteria: | The test user succeeds in installing and running the app |

# Deliverables

## Test Evaluation Summaries

Test evaluation is done by hand directly after the test is executed.

## Reporting on Test Coverage

Test coverage is reported in our IDE

## Perceived Quality Reports

n/a

## Incident Logs and Change Requests

n/a

## Smoke Test Suite and Supporting Test Scripts

n/a

## Additional Work Products

n/a

### Detailed Test Results

n/a

# Testing Workflow

Unit tests are run automatically. We are additionally performing an installation test.

# Environmental Needs

## Base System Hardware

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Server Name |  | Neco.it.dh-karlsruhe.de |
| Test Development PCs | 3 | Mirko, Alex, Maurice |
| Repository | 1 | GitHub |

## 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 | 8.1 | Operating System |
| Visual Studio | 2017 | IDE |

## 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** |
| --- | --- | --- | --- |
| Project Management | JIRA | Atlassian | 7.9.2 |
| Mockup | Balsamiq |  |  |

## Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration |  |  |
| Minimal configuration supported |  |  |
| Visually and mobility challenged |  |  |
| International Double Byte OS |  |  |
| Network installation (not client) |  |  |

# Responsibilities, Staffing, and Training Needs

## People and Roles

This table shows the staffing assumptions for the test effort.

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager | 1 | 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 | 2 | 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 | 3 | 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 | 3 | 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 | 1 | 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 |
| Designer | 1 | 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 | 3 | 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.

# Iteration Milestones

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| >20% Test Coverage |  |  |  |  |
| Have UI tests |  |  |  |  |
| Have Unit tests |  |  |  |  |
| Have installation test |  |  |  |  |

# Risks, Dependencies, Assumptions, and Constraints

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Test data proves to be inadequate | <Customer> will ensure a full set of suitable and protected test data is available.  <Test> will indicate what is required and will verify the suitability of test data | * Redefine test data * Review test plan and modify components (scripts) * Consider load test failure |
| Technical Problems | <Tester> needs to make sure everything is running fine | * Fix the problem |

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

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

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

# Appendix

To-Do

Screenshot VS Code metrics

Cyclomatic complexity