

Unit Test Plan

Eindhoven, January 15, 2010

UTP-1.0.3108

TU/e Technische Universiteit
Eindhoven
University of Technology

Where innovation starts



Project Manager:

Wilco Belgraver Thissen, 0514143

Quality Assurance Manager:

Jelle Hellings, 0592127

Senior management:

Mark van den Brand, HG 5.59

Lou Somers, HG 5.36

Advisor:

Erik Luit, HG 7.12

Customer:

Natalia Sidorova, HG 7.84

Project team:

Roy Berkeveld, 0608170

Gijs Direks, 0611093

Michael van Duijkeren, 0535368

Neal van den Eertwegh, 0610024

Dion Jansen, 0590077

Koen Kivits, 0608715

Sander Leemans, 0608896

Kevin van der Pol, 0620300

Nick van der Veecken, 0587266

Computer Science, TU/e

Abstract

This document is the Unit Test Plan (UTP) of GROUP QIS. This project is part of the Software Engineering Project (2IP35) and is one of the assignments at Eindhoven University of Technology. The document complies with the UTP from the Software Engineering Standard, as set by the European Space Agency [1].

This document provides the main guidance for the Unit Test (UT) during the Detailed Design (DD) phase for the QIS application. It describes the environment needed to perform the UT. When this environment is set up, all test cases must be executed according to their corresponding test procedures. After a test has been performed a report needs to be written.

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Overview	4
1.3	List of definitions	4
1.4	List of references	4
2	Test plan	6
2.1	Test items	6
2.2	Features to be Tested	6
2.3	Test deliverables	6
2.4	Testing tasks	7
2.5	Environmental needs	7
2.6	Test case pass/fail criteria	7
3	Test case specifications	8
4	Test procedures	9
5	Test reports	10

Document Status Sheet

Document status overview

General

Document title: Unit Test Plan
Identification: UTP-1.0.3108
Authors: Gijs Direks, Neal van den Eertwegh, Nick van der Veeken
Document status: Final

Document history

<i>Version</i>	<i>Date</i>	<i>Author</i>	<i>Reason of change</i>
0.0	10-11-2009	Neal	Draft
1.0	12-01-2010	Nick	First version

Document Change Records since previous issue

General

Date: 12-01-2010
Document title: Unit Test Plan
Identification: UTP-1.0.3108

Changes

<i>Page</i>	<i>Chapter/paragraph</i>	<i>Reason to change</i>
-------------	--------------------------	-------------------------

Chapter 1

Introduction

1.1 Purpose

The UTP document describes the plan for testing the developed software units against the detailed design, defined in the DDD[3]. The unit tests make sure that QIS complies with the design in the Detailed Design (DD) phase of the QIS project as described in the ESA software engineering standard[1].

1.2 Overview

Chapter 2 gives an overview of all items to be tested, and the general criteria for the UT. Chapter 3 specifies how tests are defined, chapter 4 specifies the test procedures and chapter 5 specifies how test results are reported.

1.3 List of definitions

DD	Detailed Design
DDD	Detailed Design Document
ESA	European Space Agency
SVVP	Software Validation and Verification Plan
UT	Unit Test
UTP	Unit Test Plan

1.4 List of references

- [1] ESA Board for Software Standardization and Control (BSSC). European space agency software engineering standards, February 1991. (ESA PSS-05-0 Issue 2).
- [2] GROUP QIS. Unit test plan. Technical report, Eindhoven University of Technology, Computer

Science, ? 2007.

- [3] GROUP QIS. Detailed design document. Technical report, Eindhoven University of Technology, Computer Science and Engineering, December 2009.
- [4] GROUP QIS. Software configuration management plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.
- [5] GROUP QIS. Software verification and validation plan. Technical report, Eindhoven University of Technology, Computer Science, September 2009.

Chapter 2

Test plan

2.1 Test items

The software to be tested is QIS. Information about the detailed design of QIS can be found in the DDD[3].

2.2 Features to be Tested

QIS must meet the design as stated in the DDD[3]. Each component should adhere to the interfaces given in the DDD[3].

2.3 Test deliverables

Before the testing starts, the following documents must be delivered:

- SVVP[5]
- DDD[3]
- This document
- UT input data

After completing the testing, the following documents must be delivered:

- UT report (UTP[2] chapter 5)
- UT output data
- Problem reports (if any)

2.4 Testing tasks

Before any testing in the UT phase can take place, the following tasks need to be done:

- Designing the unit tests.
- Tracing all test cases to components.
- All components mentioned in the DDD[3] need to be covered by test cases.
- Creation of the UT input data.
- Ensuring that all environmental needs for the UT have been satisfied.

When these tasks have been done a UT can be performed according to the procedures described in chapter 4.

2.5 Environmental needs

To be able to perform the UT the following resources are needed:

- A computer connected to the internet running Internet Explorer 7 or Firefox 3.5.
- A server running the QIS application.

Also look at the constraints described in the DDD[3].

2.6 Test case pass/fail criteria

Every test should describe the criteria that should be met to pass a specific test. An overall UT pass can only be achieved when all tests described in chapter 3 have been performed and passed.

Chapter 3

Test case specifications

The test case specifications can be found in `/trunk/src/qis/app/tests.py`. Each suite (a class in Python) should contain a description of the tests contained in that suite. The specification of each test (a function in Python) should be the first thing included in the function's docstring.

Chapter 4

Test procedures

The author responsible for the code in a workpackage is also responsible for its tests. This means tests should be written, specified and run - possibly several times if bugs are found. Only if *all* tests are passed a piece of code can be merged to trunk. Each workpackage should have its own test suite.

All tests can be run by executing `/trunk/src/qis/tests.py` (or the equivalent of this in a branch).

Chapter 5

Test reports

The Django test suite will output a simple OK if all tests are passed. Any errors found will be printed to the screen, these should be documented in Trac (see SCMP[4]).