Master Project XXX

TESTPLAN: Unit test

ID: Unittest 1.x.x

Version 1.x
Rev. x

Authors:

Date\Signature: _____

Quality Assurer:

Date\Signature: _____

Project manager:

Date\Signature: _____

Head of project:

Date\Signature: _____

Document History

Version No.	Date	Authors

Revision Sheet

Review No.	Date	Reviewer	Revision Description
Rev. 0	xx\06\2008		Creation of this unit test plan
Rev. 1	xx\06\2008		Addition of further chapters
Rev 2	xx\06\08		Refinement of the test objects, test environment and test activities. Checking and maintaining consistency between architecture document and unit test plan.

1 Introduction

The purpose of this document is to introduce the unit test plan of the Publication Document Workflow Management System (PD WMS). The document summarizes the test objects, the relevant characteristics to be tested, the test strategy, the pass/fail criteria, the interrupt/continue criteria, the test environment and resources.

1.1 Background and goals

The PD WMS of the Fraunhofer IESE consists of five sub-processes: Initial approval, Writing Document, Quality approval, Pre-publish, and Publication and Dissemination. Details about the sub-processes of the PD WMS can be found in the Requirements Document of the Master project OSP07.

In order to make the system more usable for the employees of the Fraunhofer IESE, the system should be integrated with the tools that are most commonly used by the researchers at Fraunhofer IESE: Microsoft Office. Therefore, the PD WMS should work under the Operating System Microsoft Windows and will be integrated with Microsoft Outlook, as front end.

The goal of the unit test is to verify the functionality and completeness of each component according to the corresponding Component Design Document [1.2 Nr. 3] and the Architectural Document. For each component test cases will be designed using equivalence class and boundary values analysis as selection criteria.

1.2 Referenced documents

Nr	Name	Version	Document position
1	Requirements Document of the Master Project OPS07	1.0	~\Requirements_documentation\Requirements_Specification.doc
2	Architecture Document of the Master Project OPS07	1.0	~\Products\Architecture\ViewBasedArchitectureDocumentV_4.2.doc
3	Component Design Documents	1.0	~\Products\Component_Design\…
4	Mastertestplan	1.0	~\Products\Testing\Documents\Mastertestplan.doc
5	Testcases_1.x	1.0	
6	Testscripts_1.x	1.0	
7	Testpriority_1.x	1.0	~\Products\Testing\Documents\ unitest\...
8	Mastertimeplan	1.0	~\Products\Testing\Documents\Mastertimeplan.mpp

2 Test objects

The Architecture Document of the PD WMS [1.2 Nr. 2] defines 5 layers which should be specified during the component design. These layers are: Office application layer, Process execution layer, Business layer, Data layer and Communication layer. At the moment, the following components have been defined for each layer:

Layer	Preliminary Component
Office application layer	VOF Client Controller VOF Client View VOF Data Model
Process execution layer	VOF Document Manager VOD Document Builder Outbox Inbox
Business layer	Service Manager User Manager Service VOF Publication Service LIS Manipulation Service Statistics Service VOF Document Template Service Template Service
Data layer	VOF Document Template Access Statistic Data Access Document Template Access LIS Data Access User Data Access
Communication layer	Discovery Manager Node Manager

For each layer a Component Design Document [1.2 Nr. 3] will be created to refine and to specify the corresponding components. The test objects will be the components (units) defined in each Component Design Document [1.2 Nr. 3].

3 Test characteristics

- Completeness
  The implemented components should be completed in compliance with the specifications in the corresponding Component Design Document[1.2 Nr.3].
- Robustness
  The components should be able to handle non expected behavior from calling components like using the service of the called component not in the specified way, with an appropriate exception handling.
- Maintainability
  Although this characteristic is important for this kind of system, it is not

possible to test it in our scope, because there is no experience in testing this kind of characteristics and there are not enough resources in our project for this.

- Efficiency
  According to the requirements of this project it is less important to test this characteristic. And also no resources are available for this kind of tests.

4 Test strategy

4.1 Test approach

To test the **completeness** Black-Box tests have to be conducted.
According to the component design [1.2 Nr. 3], test cases should be generated for the specified functionality of each component. The test cases will be designed using equivalence partitioning combined with a boundary value analysis as selection criteria. The goal is to cover all the equivalence partitions for each component. Only if these components work properly, the system can provide its desired functionality. Furthermore the integration of two or more units can only start if the functionality of the components has been implemented and tested.

The Robustness will be tested by Black-box tests, according to the component design [1.2 Nr. 3]. To test the robustness, negative test cases will be used, to check the behavior of the component, regarding to irregular input combinations. These points have to be considered and included in the equivalence partitions from the completeness.

4.2 Test environment

The unit test cases should be designed and implemented by the developers using C#Unit for the Office Application layer and JUnit 4.1 for the remaining layers.

The implementation plan has assigned one team of developers for each layer. Because of resource constraints, the developers will design, implement and execute the unit test. To assure that the "testers" are independent of the developers, a team of developers should test the work of another team.

Layer	Tester team
Office application layer	Data layer
Process execution layer	Business layer, Communication layer
Business layer	2 developers of the Process execution layer
Data layer	Office application layer
Communication layer	2 developers of the Process execution layer

5 Pass \ Fail criteria

5.1 Pass criteria

- More than 90% of the unit test cases finished without a system crash.
- More than assumed 65% coding defects and no design defects should be found.[1]

5.2 Fail criteria

- Test case fails.

6 Interrupt \ continue criteria

6.1 Interrupt criteria

- Failure of one test case which block the continuity of the test sequence for one component.

6.2 Continue criteria

- Problems have been fixed.
- Start at the beginning of the test scenario, to discover all eventual dependencies of the former problem.

7 Result documents

- Unittestplan
- Equivalences classes
- Testcases
- Testprioritization
- Testreport

8 Test activities

The test activities need to assure the compliance of the defined goals including:

1. Preparing guidelines to design, implement and execute the test cases.
2. Providing instructions to the developers.
3. Designing the test cases.
4. Executing test cases.
5. Reporting and tracking bugs in Bugzilla.
6. Re-testing if necessary.
7. Reporting the final test results.

.

[1] See Mastertestplan [3]

9 Responsibilities

Test group	Remit	Name
Test management	Creating the unit test plan	
Test design	Specification of the unit test sequences, test scenarios and test cases	
Test implementation	Implementation of the unit test cases and the test bed	
Test execution	Performance of the unit test and informing the bugs	
Bug control	Report bugs and control the rework and administration of the defect list (Bugzilla)	

*See Section 4.2

10 Time plan