

Master Project XXX

TESTPLAN: Integration test

ID: Integrationtest 2.0.0

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 integration test plan.
Rev. 1	Xx\06\2008		Updated dates in time plan.
Rev 2	xx\06\2008		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 integration 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 [1.2 Nr. 1].

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 integration test is to verify the interaction of the already tested components according to the corresponding Component Design Document and the Architectural Document.

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	Testcases_2.x	1.0	
5	Testmatrix_2.x	1.0	
6	Testscenario_2.x	1.0	
8	Testscripts_2.x	1.0	
9	Testpriority_2.x	1.0	~\Products\Testing\Documents\ integrationtest\...
10	Mastertimeplan	1.0	~\Products\Testing\Documents\Mastertimeplan.mpp
11	Mastertestplan	1.0	~\Products\Testing\Documents\Mastertestplan.doc

2 Test objects

The Architecture Document of the PD WMS 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

The integration of these components, inside the layer and between layers, should be tested. The integration strategy and the corresponding test strategy are explained in Section 4 of this document.

3 Test characteristics

- **Compatibility**

Implemented components should interact properly in compliance with the specifications of the Architecture Document of the Master Project OSP07 [1.2 Nr. 2] and the Component Design Document [1.2 Nr. 3]. Also the interface format and the data exchange have to be verified.

- **Maintainability**

The system should allow changing and adding: i) components, ii) services, and iii) the application front-end, when required.

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

The **Compatibility** will be tested by black-box tests that check the correctness of the interface communication, the interface format and the data exchange between the components. Therefore it has to be differentiated, between the following errors or failure:

- A component transfers no, or syntactically wrong data so that the receiving component can't work or crashes. (Functional error, incompatible interfaces)
- The communication is working, but the involved components interpret the data in different ways. (Functional error, conflicting specification)

4.2 Test environment

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

After the unit test, the component will be integrated and tested incrementally according to the steps defined in Figure 1. The test cases will be designed using functional equivalence partitions as selection criteria. So the test cases should cover the different input interfaces of the service provider and output interfaces of the service consumer according to the Architecture Document and the Component Design Documents [1.2 Nr. 3]. Furthermore possible inheritance of the components has to be considered during the creation of the equivalence partitions.

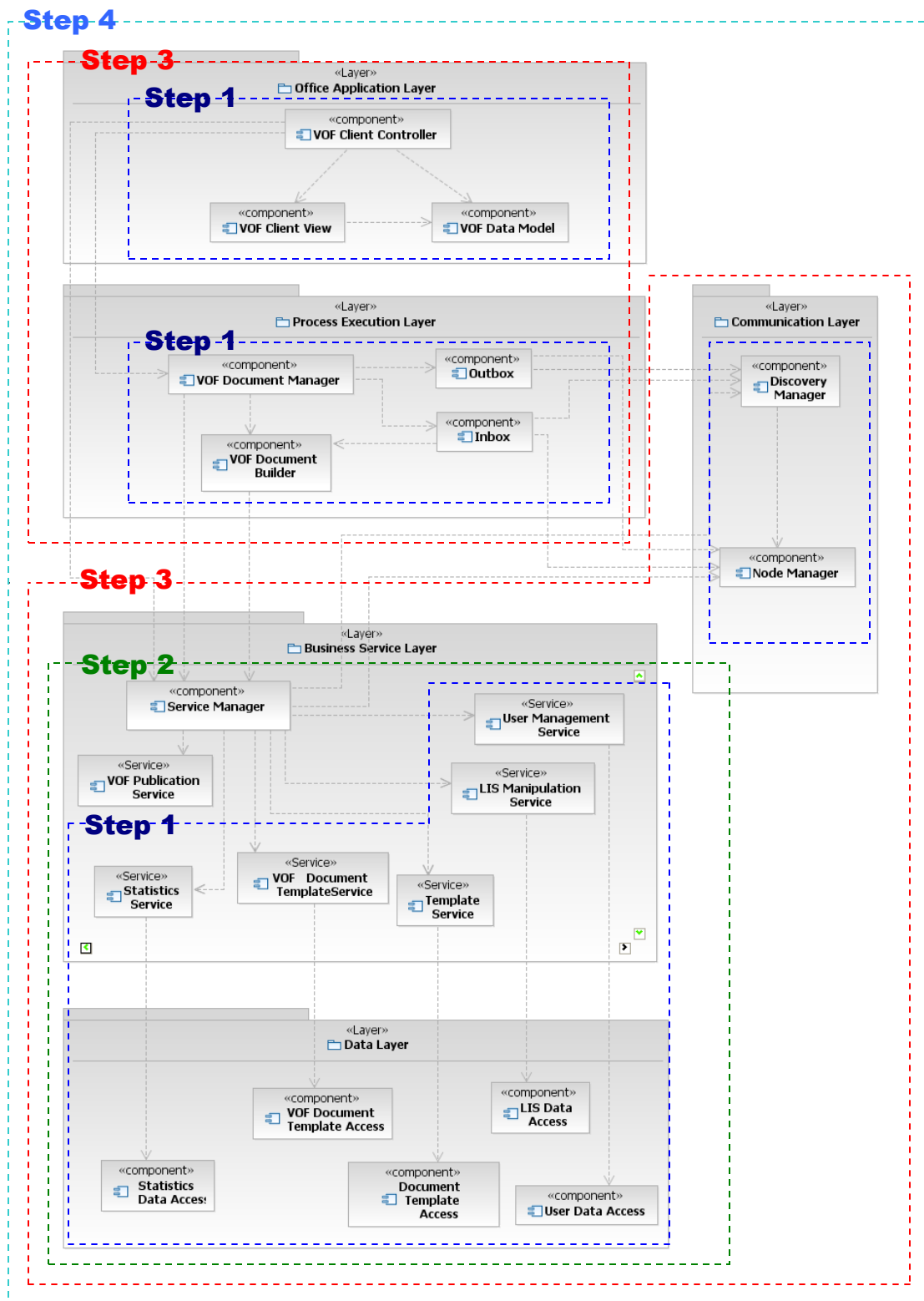


Figure 1. The diagram specifies the integration order of the components. Each color surrounds the components that should be integrated at once. Components surrounded with the same color should be integrated in parallel. The integration will be incrementally as follows: Step 1, Step 2, Step 3 and Step 4.

5 Pass \ Fail criteria

5.1 Pass criteria

- More than 90% of the integration test finished without a system crash.
- More than assumed 30% coding defects and 60% design defects should be found.¹

5.2 Fail criteria

- At least one test case fails.

6 Interrupt \ continue criteria

6.1 Interrupt criteria

- Failure of one critical test case.

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

- Integrationtestplan
- Integrationstates
- Testscenarios
- Testcases

8 Test activities

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

1. Designing the test cases.
2. Executing test cases.
3. Reporting and tracking bugs in Bugzilla.
4. Re-testing if necessary.
5. Reporting the final test results.

¹ See Mastertestplan [3], paragraph 4

9 Responsibilities

Test group	Remit	Name
Test management	Creating the integration test plan	
Test design	Specification of the integration test sequences, test scenarios and test cases	
Test implementation	Implementation of the integration test cases and the test bed	
Test execution	Performance of the integration test and writing the test reports	
Bug control	Control the rework and administration of the defect list (Bugzilla)	

10 Time plan

xx...