POLITECNICO
MILANO 1863

POWER ENJOY

Integration Test Plan Document

Lorenzo Casalino - 877421

Tommaso Castagna

Document version 1.0

Contents

1 Introduction

1.1 Revision History

The history of document revisions is here recorded in tabular format, mapping the document version with the major changes brought to.

The current version of the document is highlighted by the version number in bold format.

Version	Revision
1.0	Document first final version.

1.2 Purpose and Scope

1.2.1 Purpose

The *Integration Test Plan Document*, also referred to with the acronym of *ITDP*, aims to provide to the development team the path to follow for the integration testing process of the software system through a complete description of the elements of the system to test, the integration strategy to adopt, the integration sequence forecasted and stubs/drivers and tools needed to accomplish the integration test phase.

1.2.2 Scope

The hereby Integration Test Plan Document presents a detailed description of the integration testing plan that the development team should follow to accomplish correctly the integration testing process.

The scope of this document covers four main topics regarding the integration phase. Here, the scope is presented reflecting the main structure of the ITPD, allowing the reader to better understand what is discussed in the following sections.

- **Integration Strategy:** the entry criteria, both general and specific, that must be met before the integration test take place, what are the subsystems to integrate, which integration strategies are adopted and the integration sequence to follow are deeply described and discussed, pointing out the rationale of each choice.
- **Individual Steps and Test Description:** the type of test performed in each step of the integration process and applied to each component and subsystem, along with the expected results, are presented.

- **Tools and Test Equipment:** the needed tools to perform the integration are presented, and briefly described their usage during the integration process.
- **Program Stubs and Test Data Required:** possible stubs/drivers or special data required to proceed in each step of the integration process are presented and described.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

- **Design Document**: document describing the software system architecture, the design choices and their rationale.
- **Integration Test Plan Document:** document used to guide the integration test phase.
- **Requirement Analysis and Specification Document**: document regarding the analysis of the goals of the project stakeholder and the functional and non-functional requirements of the software system to develop.

1.3.2 Acronyms

- **DD**: Design Document.
- **ITPD**: Integration Test Plan Document.
- **RASD**: Requirement Analysis and Specification Document.

1.3.3 Abbreviations

2 Integration Strategy

2.1 Entry Criteria

Before the integration testing phase of specific components may take place, it is fundamental that the following criteria (or conditions) here described are satisfied. The verification of these conditions is really important in order to have as output of the integration test phase meaningful result, useful to assess the quality of the software system designed and, possibly, improve it.

It's worthful point out that some of the criteria presented are strictly tied to the kind of strategy chosen to perform the integration testing of the system's components. For informations about the integration testing strategy picked out for this software system, please refer to the *Integration Testing Strategy* section.

Other criteria, instead, are more general and act as pre-conditions to the whole integration testing process.

GENERAL CRITERIA

- **RASD complete draw up:** the RASD has the main function of thoroughly document the functionalities and requirements of the software system. Because of its purpose, it is the first important source of informations and comparison that the development team has to refer to check the results obtained after the integration testing of components. Therefore, the complete draw up of the RASD document is an important condition to satisfy in order to proceed with the integration testing phase.
- **RASD positive assessment**: since the development team refers to the RASD to veify the results of the integration tests, it's fundamental that the RASD content reflects the goals of the stakeholders involved into the project. Documented functionalities diverging from the stakeholders' desires lead to wrong implementations, regardless of the integration testing results.

COMPONENT TESTING CRITERIA

- **Complete functionality-to-component(s) mapping:** the development team needs to know what components support a precise functionality. Hence, when a certain functionality is going to be tested, is necessary that all the components needed are known.
- **Complete component-related-to-functionality documentation:** the kind of support provided by a certain component to a certain system functionality must be described in the Design Document.
- **Complete components interaction description:** to test the integration and interaction between components the Design Document must report in depth how they interact and communicate to support a specific

functionality.

- **Complete component static and dynamic analysis:** the supporting functionalities provided by the components to be integrated must thoroughly inspected through static analysis methods, such as code inspection, and dynamic analysis methods, i.e. unit testing. This step is important because allows to discover possible fault in the implementation, easing and speeding up the integration testing process.
- **Regression testing:** before starting the integration testing of a certain functionality through new test cases, the old test cases should be run in order to verify the compatibility and the correctenes of the new integration. This process should be executed before the new functionality testing because if the old tests show an incompatibility, the components integration must be reviewed.

2.2 Elements to be Integrated

In the following, the system's elements to be integrated are spotted and described. With *elements* here we mean the subsystems composing the system's design. The description of a subsystem is recursive, meaning that if a subsystem is composed by other subsystems, even these are described. Description of atomic components is avoided since are well described into the DD.

2.3 Integration Testing Strategy

Since the system to be developed is quite large and there's a quite number of components that can work independently from others, the most indicated strategy to approach the integration testing process is a blending of **thread-integration** method and **bottom-up** method.

To be more specific, the **thread-integration** method is *functionality-oriented*, meaning that a system functionality at a time is tested. The testing of a system functionality is achieved by the development of specific components' parts that support that functionality (this is why the component-functionality mapping must be complete and correct) and integrating them together and testing. The use of this approach allows the developers to focus on one functionality at a time and to focus on the interaction between the involved components, easing the bugs discovery, interaction issues and performance issues.

The **bottom-up** approach, instead, consists in the development of complete components and integration of them. In this project this approach is used for the integration of atomic components, that is either components which functionalities are independent from other components or components already developed, such as external components. The main benefits brought by this strategy are the simplification of the scaffolding plan thanks to the reduction of needed

stubs (real components can be used for the testing), speeding up the integration process (real components are integrated from the beginning), ease the bugs discovery and interaction/performance issues discovery.

3 Effort Spent

The effort spent by each member of the group in terms of hours is shown in the following:

- 27 December 2016 - 1h 10m
- 28 December 2016 - 1h
- 29 December 2016 - 1h 50m