

Politecnico di Milano

A.A. 2015-2016

Software Engineering 2

Test Plan

Version 0.1

Ivana Salerno

Alexis Rougnant

Daniel Vacca

January 21th 2016

Table of content

[1.Introduction 3](#__RefHeading___Toc861_1707713022)

[1.1. Revision History. 3](#__RefHeading___Toc863_1707713022)

[1.2.Purpose and Scope. 3](#__RefHeading___Toc865_1707713022)

[1.3.List of Definitions and Abbreviations 3](#__RefHeading___Toc867_1707713022)

[1.4. List of Reference Documents. 3](#__RefHeading___Toc869_1707713022)

[1.4.1.The project description 3](#__RefHeading___Toc871_1707713022)

[1.4.2.The RASD 3](#__RefHeading___Toc873_1707713022)

[1.4.3.The Design document 3](#__RefHeading___Toc875_1707713022)

[1.4.4.The documentation of any tool you plan to use for testing 3](#__RefHeading___Toc877_1707713022)

[2.Integration Strategy 3](#__RefHeading___Toc879_1707713022)

[2.1.Entry Criteria. 3](#__RefHeading___Toc881_1707713022)

[2.2.Elements to be Integrated. 3](#__RefHeading___Toc883_1707713022)

[2.3. Integration Testing Strategy. 3](#__RefHeading___Toc885_1707713022)

[2.4.Sequence of Component/Function Integration 4](#__RefHeading___Toc887_1707713022)

[2.4.1.Software Integration Sequence. 4](#__RefHeading___Toc889_1707713022)

[2.4.2.Subsystem Integration Sequence. 4](#__RefHeading___Toc891_1707713022)

[3.Individual Steps and Test Description 4](#__RefHeading___Toc893_1707713022)

[4.Tools and Test Equipment Required 4](#__RefHeading___Toc895_1707713022)

[5.Program Stubs and Test Data 4](#__RefHeading___Toc897_1707713022)

# Introduction

## Revision History

Record all revisions to the document.

## Purpose and Scope.

State the purpose and scope of the document.

## List of Definitions and Abbreviations

## List of Reference Documents.

List all reference documents, for instance:

* The project description
* The RASD
* The Design document
* The documentation of any tool you plan to use for testing

# Integration Strategy

## Entry Criteria

* Requirements specification
* Design specification
* Source code of the components
* Code inspection (optional)
* Unit test

## Elements to be integrated

Identify the components to be integrated, refer to your design document to identify such components in a way that is consistent with your design.

We will test everything in the *Component view* of the Design document. For the MTSModel we will also test the internal subcomponents.

## Integration Testing Strategy

Describe the integration testing approach (top-down, bottom-up, functional groupings, etc.) and the rationale for the choosing that approach.

We use a sandwich strategy guided by a prioritization of the critical components.

## Sequence of Component/Function Integration

NOTE: The structure of this section may vary depending on the integration strategy you select in Section 2.3. Use the structure proposed below as a non-mandatory guide.

### Software Integration Sequence.

For each subsystem: Identify the sequence in which the software components will be integrated within the subsystem. Relate this sequence to any product features/functions that are being built up.

For the general system

We have chosen a Top-Down strategy starting from the MTSModel Component. All components except the latter and the one with which the interface is tested are thus stubs until they are validated.

Note 1 : The external components, that are the Webbrowser, the MTSNotifier, the MTS\_DB, the MapServer, EmailServer and MilanoGovernment, are not implemented by this team project. However their integration might be tested anyway.

1. MTSModel → MTSIntegration

MTSIntegration validated

1. MTSIntegration → MapsServer

MapsServer valiated

1. MTSIntegration → EmailServer

EmailServer validated

1. MTSIntegration → MilanoGovernment

MilanoGovernment validated

1. MTSModel → MTSNotifier

MTSNotifier validated

1. MTSModel → MTS\_DB

MTS\_DB validated

1. MTSNotifier → PassengerNotificationsListener

PassengerNotificationsListener validated

1. MTSNotifier → TaxiDriverNotificationsListener

TaxiDriverNotificationsListener validated

1. MTSModel → MTSTaxiDriverMobileController

MTSTaxiDriverMobileController validated

1. MTSTaxiDriverMobileController → MTSTaxiDriverMobileView

MTSTaxiDriverMobileView validated

1. MTSModel → MTSPassengerMobileController

MTSPassengerMobileController validated

1. MTSModel → MTSPassengerWebController

MTSPassengerWebController validated

1. MTSPassengerMobileController → MTSPassengerMobileView

MTSPassngerMobileView validated

1. MTSPassengerMobileWebController → MTSPassengerWebView

MTSPassengerWebView validated

1. MTSPassengerWebView → WebBrowser

WebBrowser validated

**For the subsystem MTSModel**

For this subsystem, we also adopted a Top-Down strategy, starting this time with the RequestManager component. As for the previous case, all the components are stubs until they are validated.

1. RequestManger -> ReservationManager
2. ReservationManager -> RequestManager

ReservationManager validated

1. RequestManager -> SharingEngine
2. SharingEngine -> RequestManager

SharingEngine validated

1. TModel-> RequestManager
2. RequestManager -> Tmodel

TModel validated

1. RequestManager -> QueueManager
2. ShringEngine -> QueueManager
3. TModel -> QueueManager

QueueManager validated

1. PModel -> RequestManager

PModel validated

1. TModel -> DataManager
2. ReservationManager -> DataManager
3. PModel -> DataManager

### Subsystem Integration Sequence

Identify the order in which subsystems will be integrated. If you have a single subsystem, 2.4.1 and 2.4.2 are to be merged in a single section. You can refer to Section 2.2 of the test plan example [1] as an example of what we expect.

# Individual Steps and Test Description

For each step of the integration process identified above, describe the type of tests that will be used to verify that the elements integrated in this step perform as expected. Describe in general the expected results of the test set. You may refer to Chapter 3 and Chapter 4 of the test plan example [1] as an example of what we expect.

(NOTE: This is not a detailed description of test protocols. Think of this as the test design phase. Specific protocols will be written to fulfill the goals of the tests identified in this section.)

# Tools and Test Equipment Required

Identify all tools and test equipment needed to accomplish the integration. Refer to the tools presented during the lectures. Explain why and how you are going to use them. Note that you may also use manual testing for some part. Consider manual testing as one of the possible tools you have available.

# Program Stubs and Test Data

* Data base
* Zone definition