

Integration Test Plan Document

Politecnico di Milano

A.A. 2015-2016

Software Engineering 2

Raffaella Mirandola

Salvatore Perillo 853349

Claudio Sesto 841623

Summary

Introduction	3
Revision History.....	3
Purpose and Scope.....	3
Definitions and Abbreviations.....	3
Reference Documents.....	3
Integration Strategy.....	3
Entry Criteria.....	3
Elements to be Integrated.....	4
Integration Testing Strategy.....	4
Sequence of Component/Function Integration.....	5
Software Integration Sequence.....	5
Subsystem Integration Sequence.....	6
Individual Steps and Test Description.....	13
Tools and Test Equipment Required.....	14
Program Stubs and Test Data Required.....	14
Hours of Work.....	18

1. Introduction

1.1 Revision History

21/01/2016 Version 1.0

1.2 Purpose and Scope

The purpose of this Integration Test Plan Document is to provide a general guideline for the integration test of the different components.

The software implemented is the one described in the RASD and DD document, the integration test will ensure that all the components of this system will work and cooperate correctly.

1.3 Definitions and Abbreviations

- Central System -- The entire software system with all component linked
- Q/R Manager -- The manager responsible of all actions concerning the queue of one zone and the management of the all incoming requests.
- Taxi Mobile App -- The application used by the taxi driver uses to respond to the incoming requests and sets its own state.
- Technician Station -- the personal computer dedicated to implement new features to add to the main system.

1.4 Reference Documents

- Assignment 4 - Integration Test Plan
- R.A.S.D My Taxi Service
- D.D. My Taxi Service
- [Documentation of Selected Tool \(dopo averlo scelto\)](#)

2. Integration Strategy

2.1 Entry Criteria

Referring to the Design Document, we can divide our system in 7 subsystems:

- Central system
- Q/R Manager
- Taxi mobile App
- GPS
- Databases (static/dynamic)
- Web/Mobile App
- Technician station

Each of this subsystem must work before performing the integration test.

We assume that:

- The software used in the technician station have been bought and work properly
- The databases are implemented with an existing technology (MySQL or similar) so work correctly
- The GPS is provided by a company, we must just test the communication with the taxi driver's smartphone

Instead, for what concerning the entry criteria about the other component, and in particular the web/mobile app, before integration test they must be installed on the Central System and must be online/available for download; the same is for the taxi application.

About the Q/R Manager we also already have implemented the division of the city zones

2.2 Elements to be Integrated

Depending on the section above, the elements that need to be integrated for their correctness are:

- ❑ The GPS (is a external GPS placed on the taxicar) that must be integrated with the application installed on the taxi driver's smartphone.
- ❑ All the Technician Workstations that must be integrated with the central system.
- ❑ The databases (we have two databases, Static and Dynamic as described in the D.D.) that must be integrated with the "Taxi Software" and the Web/Mobile Application to guarantee a correct flow of data between the components and a safe saving.

2.3 Integration Testing Strategy

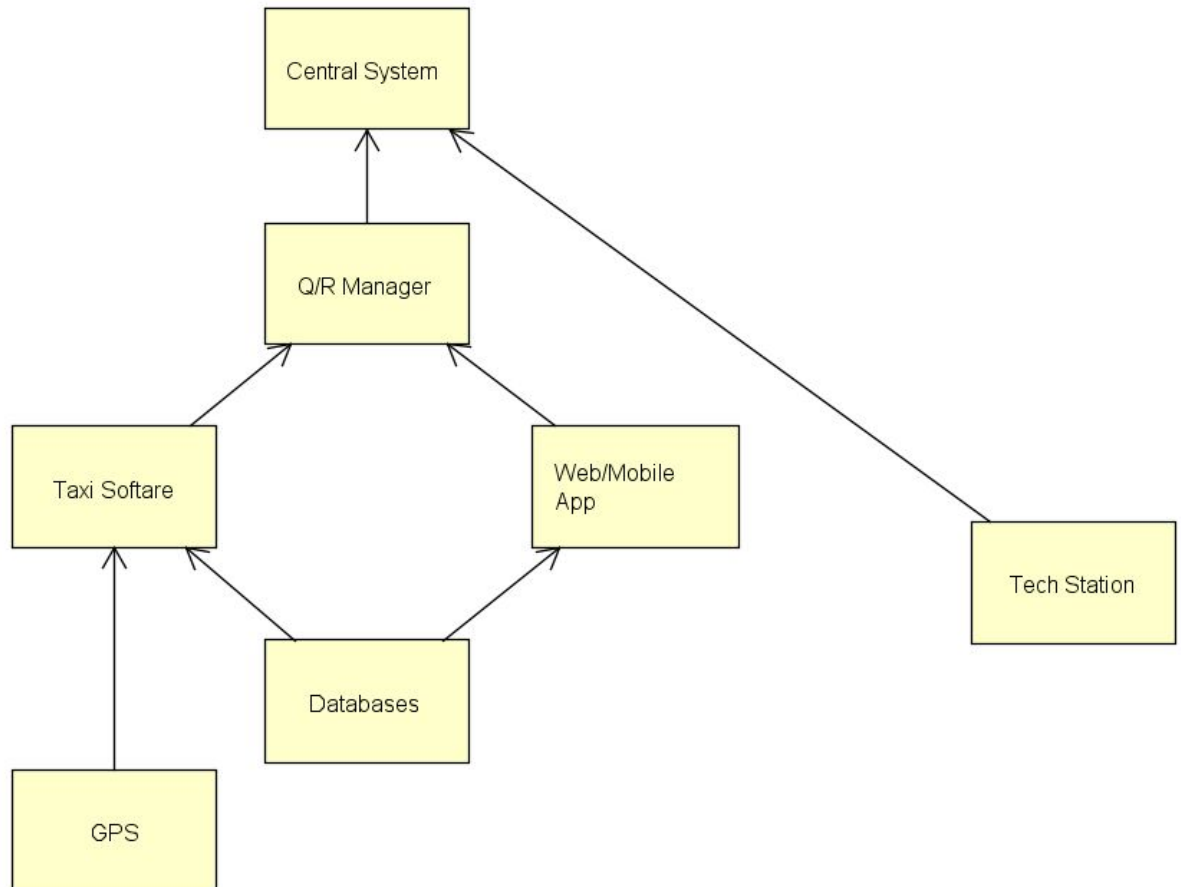
We 've decided to use the standard bottom up testing strategy: in this way the integration of the top component will be simpler once the lower ones are integrate without creating stubs.

2.4 Sequence of Component/Function Integration

2.4.1 Software Integration Sequence

Due the choice we made in the previous paragraph, the testing part will start from the lower levels, that are also the smaller components: each component has also some internal functions that must be tested in order to guarantee the correctness of the component.

This detail is shown in the next subparagraph.



2.4.2 Subsystem Integration Sequence

In the following graph you can see the components with their main functions that has to be tested: the arrows shows the order in which the test has to be performed. Obviously the

- 3. Individual Steps And Test Description**
- 4. Tools and Test Equipment Required**
- 5. Program Stubs and Test Data Required**
- 6. Hours of Work**