Politecnico di Milano

Software Engineering II project:
PowerEnjoy

Design Document

Gregori Giacomo and Ruaro Nicola

January 5, 2017
Version 0.1

Contents

Abstract

This document provides a more technical description about the PowerEnJoy system adopting the IEEE-1016 standard for DD documentation.
The scope of the Design Document is to discuss our architectural and algorithmic design choices and the user experience that PowerEnJoy should provide. It is based on the Requirement Analysis Specification Document presented in the previous delivery.

The Test Plan Document (ITPD) aims at describing how you plan to accomplish the integration test. This document is supposed to be written before the integration test really happens. Often it is written in parallel to the Design Document and takes the architectural description of the software system as a starting point. This document needs to explain to the development team what to test, in which sequence, which tools are needed for testing (if any), which stubs/ drivers/oracles need to be developed.

This document describes the Integration Test Plan (ITP) for the SPINGRID project and was made according to the software engineering standard provided by the European Space Agency described in [ESA]. This document contains the description of the integration tests for the project. This project is one of seven assignments for the course 2IP40 at Eindhoven University of Technology.

Introduction

1.1 Purpose

The Integration Test Plan Document (ITPD) is intended to provide the guidelines to accomplish the integration test phase planning in sufficient de- tail. This also includes determining which tools are needed and will be used during the testing process itself, as well as the required stubs, drivers and data structures that will be useful during said process.

This document describes the plans for testing the integration of the created components. The 15 purpose of this document is to test the interfaces between the components as described in [ADD, chapter 5]. Every team member who cooperates in the integration tests should read this document.

1.2 Scope

PowerEnJoy is a car sharing service that only employs electric vehicles; it is provided for a large city, and aims to support the sharing process and car management of the electric cars, as well as the booking and payments for the service itself.

The software implements a computational grid. This grid is able to execute jobs when it 20 receives an application accompanied by a set of data files. By hiding the complexity of grid technology the system will be easy to use. Usability is also increased by offering a web-based front-end for users to access the system.

Integration strategy

2.1 Entry criteria

Specify the criteria that must be met before integration testing of specific elements may begin (e.g., functions must have been unit tested).

2.2 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.

2.3 Integration testing strategy

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

2.4 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.

2.4.1 Software Integration Sequence

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.

2.4.2 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 required

Based on the testing strategy and test design, identify any program stubs or special test data required for each integration step.

Appendix A: Used Tools

A.1 LaTeX

Used to format and redact this document

A.2 *g*it

Used as version control system in order to lead development

A.3 *d*raw.io

Used to draw mockups and diagrams

Appendix B: Hours of work

These are the hours of work spent by each group member in order to redact this document:

- Ruaro Nicola: 0 hours
- Gregori Giacomo: 0 hours
- Total worktime: 0 hours

Appendix C: Revisions

These sections will be eventually redacted during future post-release updates in order to approach the ITPD modifiability providing a comfortable and higly effective way to trace changes:

Bibliography

[1] Luca Mottola and Elisabetta Di Nitto, *Software Engineering 2: Project goal, schedule and rules*, 2016

[2] Nicola Ruaro and Giacomo Gregori, *RASD: Requirements Analysis and Specification Document*, 2016

[3] Nicola Ruaro and Giacomo Gregori, *DD: Design Document*, 2016