

Low-motivation: Master Test Plan

Michael Ghaben

Contents

1	Introduction	1
1.1	Document Identifier	1
1.2	Scope	1
1.3	System Overview and Key Features	1
2	Test Overview	2
2.1	Master Test Schedule	2
2.2	Integrity Level Schema	2
2.3	Responsibilities	2
2.4	Tools	3
3	Details	3
3.1	Process	3
3.2	Test Documentation	3
3.3	Test Administration Requirements	3

1 Introduction

1.1 Document Identifier

This document is the Low-Comotovation Master Test Plan(MTP). In this document we detail the testing specifications, requirements, and procedures for the implementation and evaluation of testing procedures.

1.2 Scope

In this document a number of assumptions regarding the projects scope and lifecycle are made. Specifically:

1. Due to the short development cycle associated with this project, some non-negligible defects will likely persist
2. Under the iterative development methodology this group is following, the development of additional tests may be completed to address defects is expected

To better address these constraints, we will utilize a continuous integration methodology around iterative development and testing. So that we may ensure rapid development with minimal time overhead, we shall utilize continuous integration tools and methodology. Testing will be broadly divided into two categories: subsystem, and integration. Subsystem testing will be primarily focused on testing an individual system to ensure minimal functionality of a given subsystem and be done primarily independently by each group member. Integration testing will be designated as tests which require two or more modules functionality to satisfy requirements.

Specifically, it is anticipated that software defects will be found and require that testing which was not anticipated. As a result, we shall primarily focus on defining general tests to be implemented and leave the exact testing requirements as implementation is completed. Additionally, will primarily focus on subsystems to be delivered to the end-user, the train company.

Examples of these subsystems would be the CTC module or the train controller, which are expected to be integrated into the final deliverable. Subsystems which are not examples of the final deliverables are the Train Model and Track Model subsystems, as they will ultimately be removed and replaced with the physical subsystems.

1.3 System Overview and Key Features

The purpose of the system under development is to provide a train system for the Pittsburgh North Shore Rail system.

This system broadly consists of 6 subsystems:

1. Track Model - a physical model of a track to be used for testing

2. Train Model - a physical model of a train to be used for testing
3. CTC System - A CTC system to be implemented on the final train system
4. MBO - ???
5. Wayside System - A wayside for coordinating with all models

For a further discussion of the system, we refer the reader to the project requirements and discussion board.

2 Test Overview

The test organization is broadly divided into two sections: subsystem and integration testing. Subsystem tests are regarded as tests associated with only a single system at a time. Integration tests broadly refer to the tests of the integration of more than a single subsystem. In this view, testing is accomplished by each subsystem independently at the discretion of the individual developing the subsystem. Then, as members of the team develop their subsystem, each member shall attempt to integrate and develop tests for their integration. Due to the nature of continuous integration, tests are expected to be developed in parallel to the development of the program modules.

2.1 Master Test Schedule

The test schedule will be implemented as follows:

2.2 Integrity Level Schema

For this project, we only utilize three integrity levels. The integrity levels for this project are as follows:

1. The lowest severity level. This is reserved for tasks which will ultimately not be passed onto the finished product and pose no threat to catastrophic failure.
2. This severity level is reserved for failures which may lead to errors in subsystem integration or incorrect information delivered to critical subsystems
3. The most severe integrity level. Is a vital system or otherwise threatens life or limb in an implemented subsystem

2.3 Responsibilities

TBD

Table 1: Test Plan

Task	Test Design
Methods	How?
Inputs	What Inputs?
Outputs	What is a successful output?
Expected Completion	When will it be done?
Risks and Assumptions	What are you assuming?
Responsibility	Who are you?

2.4 Tools

To implement the test environment and better facilitate a continuous integration test environment, we utilize Travis-CI¹ with GitHub² integration with Slack³ and Jupiter JUnit⁴ integration. By utilizing these tools, we allow automated tests to be run remotely using the Maven⁵. This allows for rapid feedback into the developmental process, allowing for better integration and testing of the team. The usage of these tools creates feedback loop between implementation, testing, and integration, leading to significant productivity gains.

3 Details

3.1 Process

The test run shall be detailed in the following manner:

3.2 Test Documentation

Each test shall be documented using the above table as well as any auxillary information by whomever holds testing responsibilities. Each module shall have a specified test plan for each module covering their individual component for testing. Each subsystem test plan shall detail unit tests for his or her own module.

Additionally, integration testing will be accomplished in a similar fashion completed by the group.

3.3 Test Administration Requirements

For a test to be considered complete, it must successfully build on the continuous integration server utilizing the Maven build system.

¹Travis-CI.org

²github.com

³slack.com

⁴junit.org

⁵maven.apache.org