**

Software Quality Assurance Plan

Version 1.0

February 24, 2012

Bazinga

|  |  |  |
| --- | --- | --- |
| List Of Revisions | | |
| Date | Name | Description |
| 2/23/2012 | Group | Initial creation of SQAP |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Table of Contents

List Of Revisions 2

Table of Contents 3

1 Purpose 4

2 Reference Documents 4

3 Documentation 4

4 Coding Standards 4

5 Reviews 4

6 Test 4

7 Problem Reporting / Corrective Action 4

8 Tools 4

9 Code Control 4

10 Training 4

11 Risk Management 4

# Purpose

The purpose of this document is to outline the software quality assurance plan for the Allegheny County North Shore Extension Transit System.

# Reference Documents

Bazing SRS

Bazinga Coding Standard

Bazinga Defect Tracking Policy

Bazinga Configuration Management Plan

Bazinga Software Design Document

Bazinga Risk Assessment

Bazinga Software Architecture Document

Bazinga Software Test Plan

# Management

## Organization

The system is split into the following five sub components, each with a responsible party:

### Train Model

Altay Hunter

### Track model

Ryan Mohan

### Track Controller

Sean Cardello/ Gilbert Liu

### Train Controller

John Selker

### CTC Model

Jeremy Nelson

# Documentation

The software system shall provide documentation for the following:

### Bazing SRS

### Bazinga Coding Standard

### Bazinga Defect Tracking Policy

### Bazinga Configuration Management Plan

### Bazinga Software Design Document

### Bazinga Risk Assessment

### Bazinga Software Architecture Document

### Bazinga Software Test Plan

### Module SRS

### Module Design Document

### Module Test Plan

### Module Help Files

### Module Instillation Guides

Documents will be reviewed by the author and one other developer

# Coding Standards

See Bazinga Coding Standards Document.

# Reviews

Software shall be reviewed by the author and one other developer. Comments from the review shall be documented in the defect tracking system and assigned to the author.

# Test

All classes will be Unit Tested by test plans laid out in advanced for each module. The system as a whole shall be tested at integration by running all individual Unit Tests. The system shall also be tested through human input to the CTC.

# Problem Reporting / Corrective Action

See Bazing Defect Tracking Policy.

# Tools

Visual Studio 2010 shall be used as the development environment and for running unit tests. GitHub shall be used for source code control and defect tracking. Gliffy and Visual Studio 2010 shall be used to develop UML documents.

# Code Control

GitHub shall be used as a means of version control management. Developers shall collaborate using a single online repository.

# Training

Training will be conducted according to the users outlined in the Bazinga SRS.

# Risk Management

See Individual module risk assessment documents.