**Test Plan Report Outline**

Project: Graffiti Incident Tracking System

Team: Michael Jannain, Julian Nunez, Jimmy Doan, Jose Garcia

Date: 3/15/2016

**Note: Presentation should summarize the information below and the more detailed descriptions should be provided in the appropriate sections of the Team Project Notebook.**

1. **Cover Page identifying test plan report, project, team, and date of report**
2. **Introduction**

**Graffiti Incident Tracking System** - developed for Central City Public Works personnel and Law Enforcement Officials. Enables Public Works personnel to document graffiti info and send it to investigating Law Enforcement officials through an Internet Portal. Allows Law Enforcement officials to identify “hot spots” and view/edit/update graffiti & suspect info through a User Interface. Authorized personnel will also be able to request, view, or print reports based on information in the database.

1. **Software Test Environment**

Describes the software, hardware, and firmware items required to support the testing activities. This includes operating systems, compilers, code auditors, dynamic test analyzers, drivers, test data generators, data analysis/reduction tools, computer platform and devices, interface simulators, etc.

*This can be presented as several lists: a list of hardware items, system software items, and support software tools. For support software tools, you should indicate which ones (if any) need to be developed by the test team/person. And don’t forget to include a problem reporting system.*

* Hardware Items:
  + Ethernet Adapter
  + Wifi Adapter
  + PC (windows OS preferred)
* System Software Items:
  + Pycharm Debugger
  + Pycharm Python Unit Test
  + Python Interpreter (2.7)
  + Django Python Library
  + Jinja2 Python Library
  + Tkinter Python Library
  + Windows OS
* Support Software Tools:
  + Figleaf code tester from PYPY
  + Rollbar (problem reporting)
    - https://rollbar.com/

1. **A. Test Description**

Generally describes the type of testing to be performed for each software item. This description includes the following for each software test:

1. Test objective: The test objective is to make sure all of our functions run properly. Since we are working now on getting the data base up. We would test the efficiency of the database and if it can successfully store our data.

2. Requirements tested by test (use ID numbers from requirements matrix):

3. Type(s) of testing to be used in test: We will use database testing. We will be using this method because it uncovers problems with COTS interfaces, Data corruption, and Unauthorized access.

4. Pass/fail criteria:

5. Assumptions and constraints:

This should be a high level description of the tests that will be used to test software functionality, software systems performance (timing, throughput, etc.), and interfaces to other (external) systems. For example using the Mountain Lion Detection System, you might define the following as one of your tests:

*Animals-R-Here tests – tests the functionality of the COTS product needed by the Mountain Lion Detection System. The objective of the test would be to confirm the COTS capabilities with respect to detecting mountain lions using the Animals-R-Here sensors and providing accurate information about the detections to the MLDS controlling computer system. The requirements tested would be 1 a through d. The type of testing would be equivalence class testing (based on types of animal noises). The pass/fail criteria would indicate that the Animals-R-Here can detect mountain lion noises within the area covered by the sensors and provide information to the controlling computer that indicates the location of the mountain lion within 3 meters and the strength of the noise. (You would not need to test all animal noises, since MLDS is only interested in mountain lions. Later in your test plan, though, you would include test data to make sure there are no false positive reports.) You would have an assumption about the availability of the Animals-R-Here System and might constrain the test to a simulated environment (e.g., you would not need to use a real mountain lion).*

*Your other test might be a test to evaluate the controlling computer functionality (you would need to describe it as above). Defining these tests at a very high level like this helps you to figure out what test resources you need.*

**B. Data recording, reduction, and analysis**

**Response Time:** GITS will respond within 5 seconds after user completes all steps for a transaction and commits that transaction to the database.

**Validation of Input:** Input for every field will be validated to comply to a combination of user-defined and user-maintained parameters.

**Single Entry of Information:** All information will only need to be entered once to be seen by City Crew and Law Enforcement through the database.

**Entry and Processing of names:** Hyphenated names will be processed correctly and name searches using standard wildcard characters will be allowed.

**GITS Input:**

* User name, User employee number, User’s job title, User password.
* Crew information (Crew ID, Supervisor Name, Date on site, Scale of cleanup effort).
* Graffiti information (Type of building or structure, Street Address of building or property, Nearest Cross Streets, GPS Coordinates, Moniker, Number of images submitted).
* Status of investigation (new, in process, in litigation, resolved).
* Suspect information (Suspect name(s), Suspect image, Gang or crew name, Status: unknown, identified, in custody, released).

**Frequency of Incidents Calculated by:** range of dates, range of GPS coordinates, suspect name, gang or crew name.

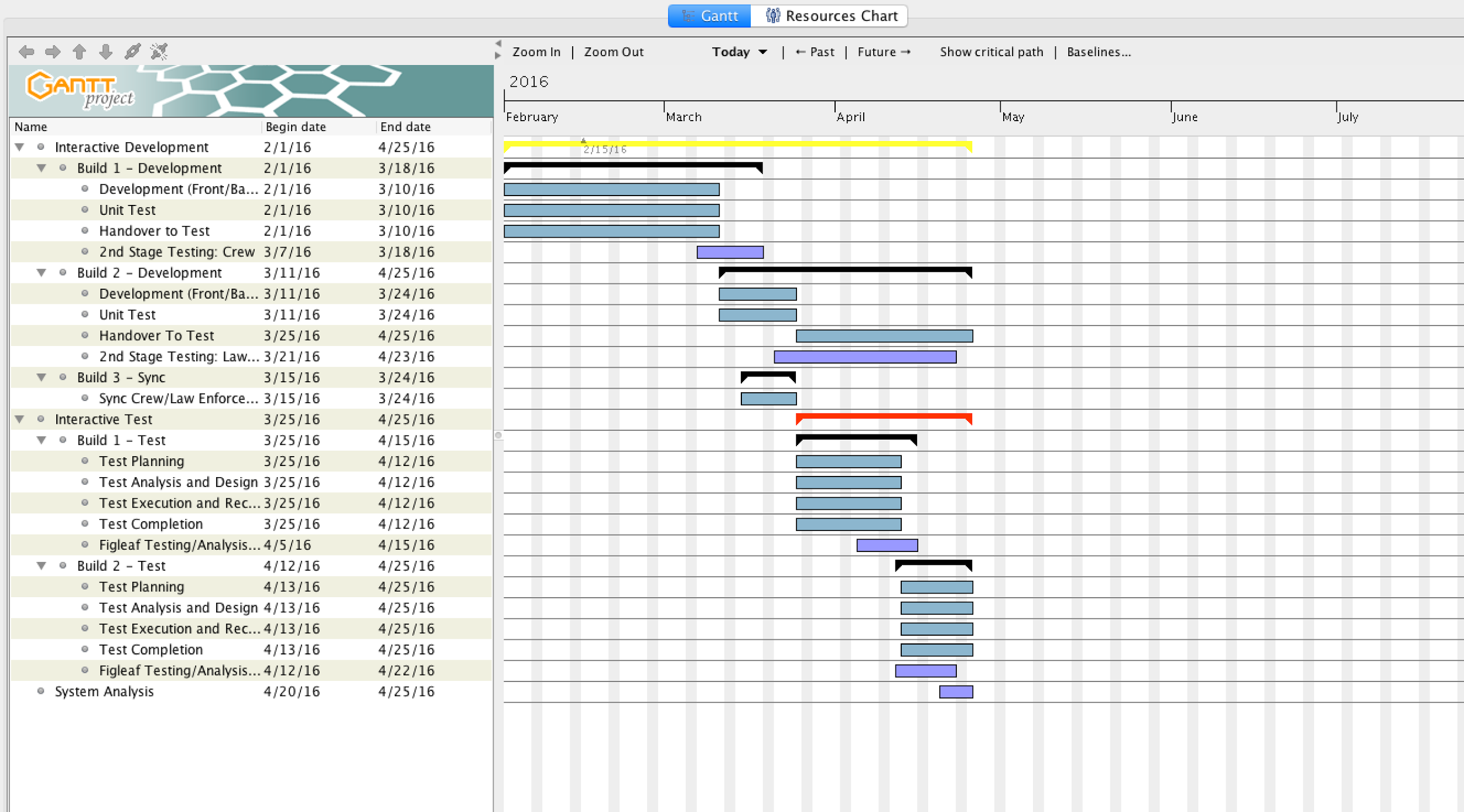
**Hot Spot Calculation by:** frequency threshold in days (default is 7.0), GPS coordinate (default is Central City Town Hall), range in miles (default is 0.5), hotspot = (range / frequency threshold).

1. **Test Condition and Schedule**

Updated project schedule that provides a schedule of key testing activities to be performed in the planning, design, execution, and analysis of test results.

*If your project schedule does not indicate these test activities, they should be added to your project schedule and you should show the relationship between test and development activities (e.g., you need the requirements baseline before you start the initial test plan).*

*Purple Sections: Testing (see labels on left side)*



1. **Updated requirements matrix showing the test qualification method(s) that will be used to test each requirement to be implemented in your first software increment. For those requirements tested by a specific test (one or more) and/or analysis techniques, provide the ID number of those test(s).**

*See requirements matrix example – this should be maintained as a separate file since it will evolve as you continue through the development/test phases.*