|  |
| --- |
| Testing Document |
| The Client: Going Green |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
|  | 1.0 | Initial Draft | Team |
|  |  |  |  |
|  |  |  |  |

Table of contents

[1. Introduction 3](#_Toc359383600)

[1.1 Document Objective 3](#_Toc359383601)

[1.2 Document Overview 3](#_Toc359383602)

[2. Testing Scope 4](#_Toc359383603)

[2.1 Testing Scope 4](#_Toc359383604)

[2.2 Assumptions 4](#_Toc359383605)

[2.2.1 General assumptions 4](#_Toc359383606)

[2.2.2 Phase specific assumptions: 4](#_Toc359383607)

[2.3 Dependencies 4](#_Toc359383608)

[2.3.1 General dependencies 4](#_Toc359383609)

[2.3.2 Testing phase dependencies: 5](#_Toc359383610)

[2.3.3 Production promotion dependencies 5](#_Toc359383611)

[2.4 Constraints 5](#_Toc359383612)

[3. Testing organization 6](#_Toc359383613)

[3.1 Testing dates 6](#_Toc359383614)

[3.2 List of players 6](#_Toc359383615)

[3.3 Involvement in the test phases 6](#_Toc359383616)

[4. Components to be tested 8](#_Toc359383617)

[4.1 Inclusions 8](#_Toc359383618)

[4.2 Exclusions 8](#_Toc359383619)

[5. Testing design 9](#_Toc359383620)

[5.1 Test Level Entry and Exit Conditions 9](#_Toc359383621)

[5.2 Test Case Estimate 9](#_Toc359383622)

[5.3 Testing Deliverables 10](#_Toc359383623)

[5.4 Metrics 10](#_Toc359383624)

# Introduction

The objective of this project is to design, develop, and deploy a carpooling application that aligns with The Client’s Green Initiative. This application will provide easy access to carpool options for The Client’s “members”.

## Document Objective

This document provides The Client with the overall plan to manage the testing activities of each test level for the Going Green project.

This document is for:

1. The operations teams and CPMO, who needs to ensure that the deliverable milestones have been met and the application is properly tested;
2. The project manager, who needs to document all the test planning requirements (managerial process, organizational process, technical process, schedule, etc.) and monitor adherence to the test strategy and plan;
3. The test team, who needs to know:
   1. The responsibilities for the different test activities;
   2. What needs to be tested and when.

## Document Overview

This document is organized into the following sections:

1. Testing scope that covers the aspects specified below:
   1. Testing phases that are in scope for this project;
   2. Assumptions, dependencies and constraints related to the test phases of the project;
2. Testing organization that covers the following points:
   1. The target dates of each phases of testing;
   2. The list of players and their roles in the test phases;
   3. The involvement of the different roles in the various test phases
3. Components to be tested will specify which parts of the system (specific modules, functionalities, software features, interfaces, etc.) will be tested.
4. Testing design that covers the following:
   1. Entry and exit conditions for each test level;
   2. Estimated number of test cases for each component at each test level;
   3. Testing deliverables to be produced in the project;
   4. Test metrics.

# Testing Scope

## Testing Scope

| Testing Phase[[1]](#footnote-1) | In Scope | Environment | Notes |
| --- | --- | --- | --- |
| Unit Testing | Yes | Development |  |
| Integration Testing | Yes | Development |  |
| Data Foundation Testing | Yes | Development |  |
| System Testing | Yes | Development |  |
| User Acceptance Testing | Yes | Production |  |
| Support Model Dry Run | No |  |  |
| Client Acceptance Testing | Yes | Production |  |
| Stress (Performance) Testing | Yes | Production |  |

Table 1: Testing Phases in Scope

## Assumptions

### General assumptions

A maximum of 3 weeks is required for testing.

### Phase specific assumptions:

* A maximum of 2 days is required for the following testing phases:
  + Unit Testing
  + Integration Testing
  + Data Foundation Testing
  + System Testing
  + User Acceptance Testing
* Client Acceptance Testing requires a maximum of 5 days.

## Dependencies

### General dependencies

* Member database integration must be functional prior to testing;
* All testing phases must be completed on both Android and iOS devices.

### Testing phase dependencies:

* All Data Foundation bugs will be resolved prior to System Testing;
* All critical bugs must have been resolved before starting the User Acceptance Tests;
* At least 50% of all medium and low priority bugs have been resolved before starting User Acceptance Tests;

### Production promotion dependencies

* All critical bugs preventing implementation into production have been resolved;
* At least 75% of all the medium and low priority bugs have been resolved.

## Constraints

* User Acceptable Testing must be completed by August 1, 2014, because one full week is required for Client Acceptance Testing;
* All tests must be completed by August 8, 2014.

# Testing organization

## Testing dates

| Test phase | Planned start date | Planned end date |
| --- | --- | --- |
| Unit Tests | 07-21-14 | 07-22-14 |
| Integration Tests | 07-23-14 | 07-24-14 |
| Data Foundation Testing | 07-25-14 | 07-28-14 |
| System Tests | 07-29-14 | 07-30-14 |
| User Acceptance Tests | 07-31-14 | 08-01-14 |
| Support Model Dry Run | Not applicable | Not applicable |
| Client Acceptance Test | 08-04-14 | 08-08-14 |

Table 2: Testing dates

## List of players

| Role | Name | Contact information | Group |
| --- | --- | --- | --- |
| Test Prime | Doug Murphy |  |  |
| System Analysts | Emmanuel Smith |  |  |
| John Phillipp |  |  |
|  |  |  |
| Delivery Lead | Joshua Collins |  |  |
| Application Analysts | Chase Cupp |  |  |
|  |  |  |
| Business Analysts | Chris Hewitt |  |  |
|  |  |  |
| Testers | Amber Castle |  |  |
| Matt Leeson |  |  |
|  |  |  |
| Integrator | Amber Ehrett |  |  |
| Operations and CPMO | Bernard Sisk |  |  |
|  |  |  |
| User Acceptance Testers | Sara Addison |  |  |
| Dave Maloney |  |  |
|  |  |  |

Table 3: List of players

## Involvement in the test phases

| Roles →  Activities ↓ | Test Prime | System Analysts | Delivery Lead | Application Analysts | Business Analysts | Testers | Integrator | Operations and CPMO | User Acceptance Testers |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Unit Testing |  | R, A |  |  |  |  |  |  |  |
| Integration Testing | I | R, A | I |  |  |  |  |  |  |
| Data Foundation Testing | I |  | I | R, A |  |  |  |  |  |
| System Testing | I | C | I |  | R, A | R | I |  |  |
| User Acceptance Testing | I | C | I |  | C |  | R, A | R | R |
| Dry Run | I |  | I |  | C |  | R, A | R | R |
| Client Acceptance Testing | I | C | I |  | C |  | R, A | R |  |

Table 4: Test Team Involvement

Legend

R: Responsible

A: Accountable

C: Consulted

I: Informed

# Components to be tested

## Inclusions

### Log In Interface

### Member Profile Interface/Member Database

* “Join Carpool” Interface and Functionality
* “Create Carpool” Interface
* Carpool Messaging Feature
* Carpool Schedule
* Carpool Calculator Tool
* Carpool Route Functionality
* “Leaving Now” Notification Feature
* Driver Reminder Feature

## Exclusions

* Carpool Member Information Page

# Testing design

## Test Level Entry and Exit Conditions

Refer to Appendix E for the description of standard test level entry and exit conditions.

## Test Case Estimate

| Testing Phase | Component to be tested | Approximate Number of Test Cases |
| --- | --- | --- |
| Unit Testing | Log In Interface | 2 |
| “Join Carpool” Interface and Functionality | 4 |
| “Create Carpool” Interface | 3 |
| Carpool Messaging Feature | 4 |
| Carpool Schedule | 3 |
| Carpool Calculator Tool | 2 |
| Carpool Route Functionality | 5 |
| “Leaving Now” Notification Feature | 3 |
| Driver Reminder Feature | 2 |
| Integration Testing | Member Profile Interface/Member Database | 3 |
|  |  |
|  |  |
|  |  |
| Data Foundation Testing | Member Profile Interface/Member Database | 3 |
|  |  |
| System Testing | All | 6 |
|  |  |
|  |  |
|  |  |
| User Acceptance Testing | All | 6 |
|  |  |
|  |  |
|  |  |
| Support Model Dry Run | Not applicable |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Client Acceptance Testing | All | 10 |
|  |  |
|  |  |
|  |  |
| **TOTAL** |  |  |

Table 5: Test Case Estimate

## Testing Deliverables

| Document name | To be produced by | Required |
| --- | --- | --- |
| Client Test Plan | Chris Hewitt | X |
| Unit Test Report | Emmanuel Smith | X |
| Integration Test Report | Emmanuel Smith | X |
| Data Foundation Test Report | Chase Cupp | X |
| System Test Report | Chris Hewitt | X |
| User Acceptance Test Report | Sara Addison | X |
| Dry Run Test Report | Not applicable |  |
| Client Acceptance Test Report | Amber Ehrett | X |

Table 6: Testing documents to be produced

## Metrics

The quality measurements are documented in the appropriate Test Report of each test level.

You can find a description of the testing metrics normally used in the Appendix D.

Appendix A – Bug record Priority/Severity

When a bug is encountered, a detailed record is created which assigns a tracking number. The record includes a description and guidelines that describe the steps that resulted in determining the bug. In addition, any necessary attachments (i.e. files and print screens) will be inserted into the record.

The record will be assigned an initial severity level which is based on the following classification:

| Severity | Description |
| --- | --- |
| Blocker | System is out of service. A defect that makes the product unusable. |
| Critical | A system, subsystem, major component, major function problem has occurred with no workaround |
| Major | A system, subsystem, major component, major function problem has occurred. There is a temporary bypass or workaround. |
| Normal | A system, subsystem, major component, major function problem has occurred, but the situation is manageable. |
| Minor | Components are not functioning to specification but the situation is manageable. |
| Trivial | Cosmetic issue. The system performs as specified, but has undesirable aesthetic attributed, e.g.  spelling mistake or color. |
| Enhancement | Request for product improvement |

Table 7: Bug record severity

The record will also be assigned an initial priority level which is based on the following classification:

| Priority | Description |
| --- | --- |
| P1 | Drop everything and take care of it as soon as you see this |
| P2 | Fix before next build to test |
| P3 | Fix before final release |
| P4 | Fix can be instituted in a future release |
| P5 | Fix would be nice to have but there is no defined timeline to resolve this |

Table 8: Bug record priority

Appendix B – Description of Testing Phases

Unit Testing

Unit tests verify that individual units of [source code](http://en.wikipedia.org/wiki/Source_code) are fit for use. A unit is the smallest testable part of an application.

Integration Testing

Integration testing verifies the proper execution of application components that interface with other applications.

Data foundation Testing

Data foundation testing verifies that all data is loaded/modified as specified.

System Testing

System tests verify that all components (software, hardware and network) of the entire application execute properly. The tests also verify the system's ability to read interfaces from other applications and to generate interfaces to other applications.

User Acceptance Testing

User acceptance tests verify that the system meets user requirements as specified, and validate that the system fulfills its intended use. User acceptance testing takes place in UAT environment.

Support Model Dry Run

A Dry Run is a mental run of a program. It is a testing process where the effects of a possible failure are intentionally mitigated.

Client Acceptance Testing

Client acceptance tests verify that the system meets client requirements as specified, and validate that the system fulfills its intended use. Client acceptance testing takes place in, or simulates the client's operational environment.

Appendix C – Roles and Responsibilities

| Role | Skills | Responsibilities |
| --- | --- | --- |
| **Test Prime** | * Clear understanding of project scope, technological limitations and system architecture; * Good understanding of impacted business processes; * Good understanding of the Environment Management process of the Service Management solution; * Knowledge of “Bugzilla/Testopia” application. | * Leads the team, coordinates and oversees the entire test process; * Reviews the test plans and test reports; * Ensures entry/exit conditions for testing levels are met; * Tracks progress and reviews results with management, development entities and users; * Coordinates all the defect resolution, tracking and communicates defects through “Bugzilla/Testopia” to CGI delivery team; * Provide approval, upon successful test, to promote packages to next environment from ST to Production; * Promote code to test environments; * Install packages in the required environments. |
| **System Analysts** | * Clear understanding of the Service Management Solution and the customization done for the release; * Clear understanding of the database activities; * Good understanding of the Environment Management process of the Service Management solution; * Knowledge of “Bugzilla/Testopia” application. | * Delivery of all testing deliverables; * Support CGI Business and Application Analysts and testers during integration/system and user acceptance testing; * Build and promote code to Dev and SI environments; * Prepare code packages as per the Service Management Environment Management process; * Address all code related issues; * Obtain approval from Technical Lead for packages. |
| **Delivery Lead** | * Able to meet aggressive deadlines despite roadblocks; * Analytical and able to focus on key areas; * Self-starter (independent, self-sufficient and self-reliant); * Resilient and flexible, to respond to changing schedules; * Business or systems analysis skills. | * Addresses all escalation issues. |
| **Application Analyst** | * Clear understanding of the Service Management Solution and the data loaded for the release; * Good understanding of the Environment Management process of the Service Management solution; * Knowledge of “Bugzilla/Testopia” application. | * Prepare dataload as per the Service Management Environment Management process; * Address all data related issues. * Obtain approval from Technical Lead for packages; * Creates the Data Foundation test cases; * Execute the Data Foundation test cases; * Records the anomalies during the Data Foundation tests. |
| **Business Analysts** | * Clear understanding of the project scope, technological limitations, system architecture, Service Management Solution and the customization done for the release as well as the Business/Systems requirements, target solution, interfaces and processes; * Good understanding of the Environment Management process of the Service Management solution and of the impacted business processes; * Knowledge of “Bugzilla/Testopia” application. | * Creates the test plans and test report; * Leads the team, coordinates and oversees the entire test process; * Supports the test prime in his functions; * Helps the client to create its own test cases; * Manages communications between the Service Managememt team and the Operations and CPMO Champions; * Ensures the integration of the project within operations; * Ensures that each operations group is prepared to implement the solution; * Creates test cases; * Creates test report; * Records anomalies during UAT testing and Client acceptance testing. |
| **Testers** | * Good understanding of the business requirements, target solution and processes; * Good understanding of the Environment Management process of the Service Management solution and of the impacted business processes; * Knowledge of “Bugzilla/Testopia” application. | * Executes test cases; * Records the anomalies. |
| **Integrator** | * Good understanding of the business requirements, target solution and processes; * Clear understanding of the Service Management Solution and the customization done for the release; * Good knowledge of the operations and clients realities. | * Supervise the execution of the test phases starting with the user acceptance tests and following phases; * Act as a bridge between the users or CPMO/clients and the project team. |
| **Operations and CPMO** | * Good understanding of the business requirements, target solution and processes; * Clear understanding of the Service Management Solution and the customization done for the release. | * Identify and help manage project risks and issues at their level; * Provide deliverable sign-offs as specified in the Service Management RACI. |
| **User Acceptance testers** | * Good understanding of the business requirements, target solution and processes; * Clear understanding of the Service Management Solution and the customization done for the release. | * Create the user test cases under guidance of the BA; * Execution of the user acceptance test cases. |

Table 9: Roles and Responsibilities

Appendix D – Testing Metrics

|  |  |
| --- | --- |
| Metric | Description |
| Test progress | Specifies the number of test cases planned versus executed. |
| Test success | Specifies the number of test cases passed versus failed. |
| Test sub-level metrics | Specifies whether sub-levels reach target levels of quality. |
| Test variance impact | Specifies the number and the rate of defects by status and severity reported during testing. |
| Defect detection | Specifies graphically the number of defects by severity reported over time during testing. |

Appendix E – Description of Test Level Entry and Exit Conditions

| Test Level | Entry Conditions | Exit Conditions |
| --- | --- | --- |
| **Unit Test** | * Source code is clean and compiled; * Suggested code changes have been completed according to the code walkthroughs; * Overall Test Plan is complete; * All review reports are complete; * Unit test cases, conditions and scripts are complete. | * All unit test cases specified in the have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the Unit Test Report and all fixes have been re-tested; * Source has been integrated into the change-controlled environment according to configuration management procedures. |
| **Integration Test** | * System Design review document is complete; * Integration test cases, conditions and scripts are complete; * Unit testing is complete; * Machines and environment are set up and configured for integration testing. | * All integration test cases have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the Integration Test Report and all fixes have been re-tested; * Source has been integrated into the change-controlled environment according to configuration management procedures. |
| **Data Foundation Test** | * Data has been loaded; * Data Foundation test cases, conditions and scripts are complete; * Testers’ machines have been configured as required for client workstations. | * All data foundation test cases have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the Data Foundation Test Report and all fixes have been re-tested. |
| **System Test** | * Code has been fully integrated; * System test cases, conditions and scripts are complete; * Integration testing is complete; * Testers’ machines have been configured as required for client workstations. | * All system test cases have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the System Test Report and all fixes have been re-tested; * Source has been integrated into the change-controlled environment according to configuration management procedures. |
| **User Acceptance Test** | * System testing is complete; * Testers’ machines have been configured as required for client workstations. | * All User Acceptance test cases have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the User Acceptance Test Report and all fixes have been re-tested; * Source has been integrated into the change-controlled environment according to configuration management procedures. |
| **Support Model Dry Run** | * Support model finished and signed-off. | * All Support Model Dry Run scenarios have been executed; * Exit conditions for the severity of process issues have been met; * Process Issues have been recorded in the Support Model Dry Run Report; |
| **Client Acceptance Test** | * Product requirement documents are complete; * Client Acceptance Test Plan document is complete; * User acceptance test cases, conditions and scripts are complete. | * All user acceptance test cases have been executed; * Exit conditions for the severity of defects have been met; * Defects have been recorded in the Acceptance Test Report and all fixes have been re-tested; * Source has been integrated into the change-controlled environment according to configuration management procedures; * Client has signed off. |

Table 10: Test Level Entry and Exit Conditions

Appendix F – Issue Lifecycle Processes



Figure 1: Issue Lifecycle during System Testing



Figure 2: Issue Lifecycle during User Acceptance Testing

1. Refer to Appendix B for description of testing phases [↑](#footnote-ref-1)