System & Test Strategy Plan for CTMS

Version 1.0

This particular document describes the System & Test Strategy Plan for the CTMS

Approved By

Robert Wycoff
CDOT ITS Office

Signature: ___________________________
Date: _______________________________

John Williams
CDOT ITS Office

Signature: ___________________________
Date: _______________________________

Prepared By:

EnRoute Traffic Systems, Inc.
### DOCUMENT VERSION HISTORY

<table>
<thead>
<tr>
<th>Version / Status</th>
<th>Date</th>
<th>Prepared By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>6/3/04</td>
<td>Venky Varadarajan</td>
<td>1st Draft</td>
</tr>
</tbody>
</table>

### DISTRIBUTION LIST

This Test Plan document will be distributed to the following persons for review and/or approval.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert Wycoff</td>
<td>Network Analyst</td>
<td>ITS</td>
</tr>
<tr>
<td>Carry Weiss</td>
<td>Dept. Project Management</td>
<td>ITS</td>
</tr>
<tr>
<td>John Williams</td>
<td>Project Management</td>
<td>ITS</td>
</tr>
<tr>
<td>Pawan Kharbanda</td>
<td>Project Lead Software, Design</td>
<td>EnRoute</td>
</tr>
<tr>
<td>Jason Westra</td>
<td>System Architecture, Design, J2EE lead</td>
<td>EnRoute</td>
</tr>
<tr>
<td>Rajat Chaudri</td>
<td>Analysis, System, Design</td>
<td>EnRoute</td>
</tr>
<tr>
<td>Trey Grubbs</td>
<td>UI- Architecture, Design</td>
<td>EnRoute</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

**DOCUMENT VERSION HISTORY** ..................................................................................................2

**DISTRIBUTION LIST** ..................................................................................................................2

**1 INTRODUCTION** .......................................................................................................................4

1.1 PURPOSE .................................................................................................................................4
1.2 BACKGROUND .........................................................................................................................4
1.3 SCOPE .......................................................................................................................................4
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS ...................................................................5
1.5 PROJECT PHASES .....................................................................................................................5
1.6 TEST APPROACH .....................................................................................................................5
1.7 PERSONNEL PARTICIPATION .................................................................................................7
1.8 TEST LOCATION .......................................................................................................................7
1.9 RELATED DOCUMENTS ...........................................................................................................8
1.10 ASSUMPTIONS .......................................................................................................................8
1.11 RISKS .......................................................................................................................................8

**2 TEST MODES** .............................................................................................................................9

2.1 TEST OBJECTIVES ...................................................................................................................10
2.2 FUNCTIONAL TESTING ...........................................................................................................10
2.3 REGRESSION TESTING ...........................................................................................................11
2.4 TEST EXECUTION ...................................................................................................................11
2.5 CLIENT SIDE AND SERVER SIDE TESTING ............................................................................12
2.6 CODE COVERAGE ..................................................................................................................12
2.7 PERFORMANCE TESTING .....................................................................................................12
2.8 LOAD TESTING ........................................................................................................................12
2.9 TEST COMPLETION CRITERIA ...............................................................................................12
2.10 TEST DELIVERABLES ............................................................................................................12
2.11 TEST SCHEDULE ..................................................................................................................12
2.12 TEST STATUS REPORTS .......................................................................................................13
2.13 DEFECT REPORTING ............................................................................................................13
2.14 TEST CODE BUILD ................................................................................................................15
2.15 REQUIREMENTS TRACEABILITY .........................................................................................15
2.16 TESTING METHODOLOGY AND TOOLS ............................................................................15

**3 TEST DATA SELECTION CRITERIA** .........................................................................................16

3.1 CTMS DATA SELECTION .......................................................................................................16

**4 TESTING ENTRANCE, SUSPENSION AND EXIT CRITERIA** ..................................................16

4.1 ENTRANCE CRITERIA .............................................................................................................16
4.2 SUSPENSION CRITERIA ..........................................................................................................16
4.3 EXIT CRITERIA .......................................................................................................................17

**5 TESTING ENVIRONMENT** ...........................................................................................................17

5.1 TESTING ENVIRONMENT ......................................................................................................17
5.2 REQUIRED HARDWARE AND SOFTWARE .............................................................................17
5.3 TEST DATABASE INSTANCES ...............................................................................................18

**6 TEST PLAN SIGNOFF** ...............................................................................................................19

**7 CTMS TEST ENVIRONMENT** ...................................................................................................20

**8 APPENDIX A** ................................................................................................................................21
1 INTRODUCTION

1.1 Purpose

This document identifies the test criteria, the requirements being tested, the acceptance criteria, and the mechanism for reporting problems as they occur during testing CTMS application. The objective of the plan is to ensure that the delivered CTMS satisfies the detailed requirements, functional specifications delineated in the UseCases, Supplementary specification (SUPL) requirements and feature requirements (FEAT requirement type) detailed in Rational Requisite Pro.

The test plan describes the scope of the test effort and outlines the test cases that will be used to carry out testing of CTMS application. The actual test case scripts, which will include detailed test-execution and validation steps, will build on the outlines in this plan and will be maintained as separate documents. The test plan also identifies test assumptions and risks, describes test-data selection criteria and test entrance/exit criteria, and defines the hardware and software requirements of the proposed test environment. Finally, the document includes references to various process documents describing the test team’s standard operating procedures, communication methodology and Defect (IR) reporting and resolution.

1.2 Background

CTMS is the Colorado Traffic Management System and helps the State Highway Administration (SHA) manage Colorado’s diverse traffic management challenges. CTMS provides the operators with the tools to collect, process, and analyze traffic information throughout the state and use that information to make real-time traffic management decisions and initiate corrective actions from a command center or regional command centers, providing the DMS control functionality that adhere to the NTCIP protocol.

1.3 Scope

The scope of the CTMS Test activity covered in this document is limited to testing the functionality of CTMS introduced by CTMS Elaboration1, Construction 1,2,3. The CTMS test cases, and the subsequent execution of those cases, will verify that CTMS Elaboration1, Construction 1,2,3 performs all functions required of the CTMS Elaboration1, Construction 1,2, as specified in the requirement Documents. The primary functions of CTMS include but not limited to following use cases and other requirements:

- Login to System
- Logout of System
- Manage Users
- Add New User
- Edit User
- Change Password
- Navigate Desktop
- Issue Device Instruction
- View Device Instruction Queue
- Remove Instruction From Queue
- Log User activity
- Find DMS
- Control DMS
- View DMS
- Add Message to one or more DMS
- Use Message Library
- Adjust Brightness
- Clear DMS Message
- Poll DMS
- Test DMS Pixels
- Manage DMS
- Add DMS
- Configure Communications Pool
- Configure Communications Port
- Read Sign From DMS
- Configure DMS
- Add Alarm
- Search Map
- Check for Banned Words
- Check Spelling
- Remove DMS
- View Alarms
- Generate Fax Log
- Generate Equipment Failure Report
- View Logged Activity
- Integrate with co-trip web
- Protocol Handlers (like NTCIP)
- Clustering
- Database connections
- Communication
- MapObjects
- ArcSDE
The CTMS has been developed in conjunction with the development of other applications, including MapObjects and middleware. Although the testing of the integration middleware will require the use of application supporting the integration – MapObjects, it will not include the specific testing of those applications, per se.

### 1.4 Definitions, Acronyms and Abbreviations

Refer to CTMS/CTIS Glossary

### 1.5 Project Phases

CTMS project consists of a number of iterations. Each iteration is divided into phases namely Inception, Elaboration, Construction 1, Construction 2 and Construction 3 and Transition. Each phase is focused on a set of deliverables. For further details please refer to the documents detailed in Section “Related Documents”

Each construction Phase will be System tested and up on completion, promoted to the staging, where environment will be similar to the Production and tested for operational readiness. On successful completion the application will be promoted to the Production.

The CTMS System Test Activity will test application as it functions in a fully integrated, production-like environment, operating in conjunction with the other applications supporting Map Objects. Test cases will be executed to simulate actual business processes, and the flow of system events will be initiated as in production.

### 1.6 Test Approach

The standard approach adopted for CTMS System Test is based on the notion of requirements testing. The first step in the overall test process is requirement definition, which will be defined in Rational RequisitePro. These requirements in the form of UseCases will be mapped to test Cases in Rational TestManager. The next step in the Test Process is the decomposition of the Test Areas into Test Cases (e.g., Login testing, Manage Users test case, etc.). Prior to the testing, the test steps and test data will be reviewed and discrepancies will be corrected.

CTMS Testing will focus on User Functionality, Security, DMS testing, for NTCIP and all other DMS types, and will include spot testing of other functional areas of the System. Test progress will be reported incrementally using Test Data Forms with detailed problem reports maintained in a ClearQuest database. Test Case pass/fail decisions will be made on the basis of the acceptance criteria outlined latter in this document.

Upon completion of the System Test, Project Manager will conduct an Operational Readiness Review. At the Operational Readiness Review, Problem Reports will be reviewed from the System Test. This review will determine whether CTMS software is ready for Next level of Promotion. Same process will be repeated for promotion from staging to production.

Any software modifications will be regression tested. The RequisitePro and ClearQuest will help to track and evaluate the software modifications submitted to, determine what tests must be re-tested and when a total regression test of all the test steps will be executed.

ClearCase will be used for the Software Configuration Management. Automated ant build process will be used to make the application build and shall uniquely labeled to identify the version numbers associated with the software delivered for testing.

The test schedule and a list of tests performed and their outcome will be maintained. All approved test procedures; test configurations, test data, and test reports will be formally maintained to ensure that the test results can be reproduced. The Configuration Manager will also formally control the software under test.
1.6.1 Rational Tools Integration

Rational Integration Overview
1.6.2 Testing of Other Applications

Since the system test will be conducted in an integrated environment, there is a possibility that the testing team will encounter IR’s (Bug Reports or Incident Report) related to External systems. Any External application software problems discovered as a result of the testing performed during the CTMS testing will be sent to the appropriate application custodian. These IRs will be distinctly classified, and a determination will be made as to their effect on the functionality defined in the CTMS requirements.

List of Other Applications
- MapObjects
- ArcSDE
- Client API’s
- JSuite
- Java COMM API’s
- Quartz (Scheduler)
- Oracle
- JBOSS

1.6.3 Data Migration/Conversion

Data migration/conversion will not be required to support the CTMS system test.

1.7 Personnel Participation

The personnel participating in this testing effort will come from the CDOT and EnRoute. All personnel participating in the testing effort will be familiar with the system and test scenarios. State Government personnel will receive training as needed in operating the system prior to participating in testing.

1.7.1 CDOT

During Operational Acceptance Testing, CDOT, Centennial Engineering will be part of the team that executes the tests.

1.7.2 Development Team

The Software Development Manager is responsible for assisting in testing and coordinating the correction of problems found during testing.

1.7.3 Test Team

The test team is responsible for developing the Test Plan and Test Procedures, executing the test, regression testing, reporting any problems found during testing, re-testing corrected problems, and documenting the test results. The test team is composed of testers from EnRoute.

1.8 Test Location

All Testing will be performed at the EnRoute and CDOT facilities.
1.9 Related Documents

The requirements, architecture, design, project documents (and their authors) on which this test plan is based are:

- UseCases
- Supplementary specification (SUPL) requirements
- Feature requirements (FEAT requirement type)
- Clear Quest
- UseCase Abstract.doc
- Test Manager documents

1.10 Assumptions

All assumptions currently listed in the CTMS Architecture Definition Design documents apply to the Test Plan. In addition, the following assumptions have been added:

<table>
<thead>
<tr>
<th>Item #</th>
<th>Assumption Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The CTMS will be successfully unit tested by the development group</td>
</tr>
<tr>
<td>2</td>
<td>The Requirements Specification and Architecture Definition Design documents will be completed, approved and signed prior to delivery of software to the test team.</td>
</tr>
<tr>
<td>3</td>
<td>All open issues and concerns relating to analysis, design and development of the software will be satisfactorily closed prior to delivery of software to the test team.</td>
</tr>
<tr>
<td>4</td>
<td>A complete CTMS Test Environment will be available prior to the start of test execution.</td>
</tr>
<tr>
<td>5</td>
<td>The External software development required to support CTMS will be completed, tested and delivered, without open Severity Critical or Medium IRs, prior to the start of test execution.</td>
</tr>
<tr>
<td>6</td>
<td>Acceptance Testing will be done by the Centennial Engineering</td>
</tr>
</tbody>
</table>

1.11 Risks

All risks currently listed in the CTMS Requirement Specifications and Architecture Definition Design documents apply to the Test Plan. In addition the following risks have been identified as having an impact upon the test planning, environment, execution, completeness, and/or analysis of this application:

<table>
<thead>
<tr>
<th>Item #</th>
<th>Risk and Mitigation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>If all requirements documents (UseCases etc.) have not been completed, and/or if these documents have not been approved by the specified parties prior to the beginning of test activities, then the test plan, test cases, test schedule and all testing activities specified in this document will be reviewed and revised, as needed, upon completion of the documents identified above. Delays to the test execution activity may be required.</td>
</tr>
<tr>
<td>2</td>
<td>If the requirements agreed to within the CTMS requirements documents are revised during the course of testing activities, test cases will be redesigned, and additional time may be added to the test schedule, as needed.</td>
</tr>
<tr>
<td>3</td>
<td>If a test environment, test tools, test network, test connectivity and/or test data is not provided according to the schedule set forth within the CTMS Project Schedule, the test schedule will be expanded, as needed.</td>
</tr>
<tr>
<td>4</td>
<td>If the staffing of the test team is reduced, testing activities will be delayed accordingly, and the test schedule may be expanded, as needed.</td>
</tr>
<tr>
<td>5</td>
<td>If the project schedule is altered so that testing deliverables are required prior to the previously agreed-to project schedule, additional test resources will be needed.</td>
</tr>
<tr>
<td>6</td>
<td>If the resolution of IRs is not handled efficiently, completely, according to their priority and the associated priority-governed schedule, and by the individuals to whom the IRs are assigned,</td>
</tr>
</tbody>
</table>
testing activities will be delayed and schedule expanded, as needed.

7 The External software development required to support CTMS will be completed, tested and delivered, without open Severity Critical or Medium IRs, prior to the start of test execution.

8 The Database required to support CTMS will be completed, tested and delivered, without open Severity Critical or Medium IRs, prior to the start of test execution.

9 If development resources required to support test execution are not available during scheduled test execution time frames, testing activities will be delayed accordingly, and the test schedule may be expanded, as needed.

2 TEST MODES

The testing of CTMS will adhere to the following format:

- A single iteration of testing is planned, corresponding with one planned development iteration. The CTMS Systems Test Team will test all required functions delivered by the CTMS Development Team (as delineated in requirements for CTMS and re-test fixed IRs during the testing iteration. Test cases will be developed and executed for these functions according to the detailed requirements set forth in the UseCases and other requirement documents. Test cases will be updated as these documents are revised.

- The test team will carry out 3 – 5 rounds of testing over the course of the test iteration. Round 1 will begin when the development group has completed the development and successful unit testing of all functions required of the application. In Round 1, the test team will execute the entire set of “normal path” CTMS functional test cases. The “normal path” functional tests will validate that CTMS performs its basic required functions.

- Round 2 will begin upon completion of the fixing (may be partial, by the decision of triage) of IRs found in Round 1. In Round 2, the test team will re-execute the “normal path” cases, while adding the “alternate path” cases, described in Section 4.2.1 below.

- Rounds 3 and 4 (Round 5, if needed) will begin upon completion of the fixing (may be partial, by the decision of triage) of IRs found in Rounds 2 and 3, respectively. In both Rounds 3 and 4, the test team will carry out a regression test of CTMS functionality by running “normal path” test cases, and adding “alternate path” test cases, along with running test cases that resulted in IRs in previous rounds.

- In all testing rounds subsequent to Round 1, the team will re-test IRs and re-execute “failed” test cases from prior rounds. IRs from a given round of testing will be repaired only after that round is completed, unless test-stopping test IRs are encountered. The test team will verify IR fixes by re-testing the failed function and testing around the associated functions, if needed. Repairs of critical (Severity 1) errors during any round of testing will be completed immediately.
2.1 Test Objectives

The specific testing objectives are:

- 100% system requirements coverage
- 100% function coverage
- Valid inputs are accepted
- Invalid inputs are rejected
- All outputs are observed for correctness
- Transactional Integrity
- Persistence

2.2 Functional Testing

Functional tests will be classified according to the Use Cases outlined in the Requirements for CTMS document, as they relate to CTMS. The CTMS functional test cases will focus on these use cases, which are identified as “Functional Scenarios” in this document and listed in Appendix A.

2.2.1 Functional Test Cases To Be Performed

Functional test cases will be developed to cover variations of the two main functional scenarios. “Normal Path” test cases will cover the various CTMS functions. “Alternate Path” test cases will be executed to ensure that the CTMS operates appropriately for “alternate path scenarios. There may be scenarios that are not covered by the UseCases, they are to be identified and Test cases are to be made before the start of the testing. The functional test cases that may be executed during System Test are outlined in Appendix A. All of these test cases will be considered for the scope of System Test. However, some of the cases listed there might not be executed during System Test due to logistics, environmental limitations, and/or time constraints. They may be altered or removed from the test case set altogether. The test case set will be finalized shortly before the start of testing. Note that the test case descriptions provide summary-level information only. Detailed information covering test objectives, test data, test case steps, and validations will be included in the actual test cases, which will be maintained separately.

For the list of UseCases please refer to the UC Abstract document
2.2.2  Functional Test Cases Not To Be Performed

System Test will not verify functionality of the External systems per se. However, if errors are found in these applications during CTMS test activities, an IR will be created and assigned to the appropriate development group.

2.3  Regression Testing

Regression testing for the purpose of CTMS testing means testing around the CTMS changes to the code to verify that no adverse effects and/or eliminations of basic functions have occurred due to the mentioned changes. In addition, Regression Testing will be executed to verify the basic functions of the CTMS have been neither eliminated nor adversely impacted as new versions of the CTMS are delivered for test rounds subsequent to Round 1. “Normal path” and basic functions test cases related to CTMS modifications will be executed during Rounds 2 - 5 of the test execution effort.

2.4  Test Execution

Once the build is available for testing, the build should accompany with documentation, the list of defects being addressed in that build.

The available latest/required build will be moved from build repository to the Testing Environment and the testing will be carried on. Once the testing is completed, the identified defects if any will be communicated to the development team via Clear Quest during and after completion of that testing round.

Since the frequent builds are planned (refer section Test code build), the test build and current available build may not be synchronous. The next round of testing will be continued on the latest available Build.

To check that, the new build is not affecting the passed test cases, the test cases need to be automated progressively and need to be scheduled as required for Regression testing.

TEST CASE LIFE CYCLE
2.5 **Client side and Server side Testing**

Apart from the Functional testing the Client side and Server side code testing to be performed as required. This section will overlap with the section 4.4. Client side testing typically covers the GUI. Server side testing is to cover the server functionality and database interactions, communication the field devices.

2.6 **Code Coverage**

Since CTMS System is in early development stage the Code Coverage may not be viable at this stage. Code coverage to be performed at later stages. This test will ensure all the code is tested as the part of function testing, Client and Server side testing.

2.7 **Performance Testing**

Performance testing will be part of this system test.

2.8 **Load Testing**

Load testing will be conducted in Iteration II.

2.9 **Test Completion Criteria**

There are two types of test completion: normal test completion and abnormal test completion. Whenever possible, if tests do not complete normally, the test will be attempted again to try and pinpoint the exact anomaly causing abnormal completion. All information for tests that complete abnormally will be documented.

2.9.1 **Normal Test Completion**

A normal completed test termination is any test that completed execution and did not terminate because of failure. The test objectives were completed.

2.9.2 **Abnormal Test Completion**

An abnormal test completion is any test failure that caused the test to terminate. Each test failure will be individually evaluated to determine the severity of the problem. This evaluation will be performed and will include the analysis of the test data. Based on the evaluation, then decide to terminate or re-test specific tests.

2.10 **Test Deliverables**

The following deliverables have been identified for the System Test and Operational Acceptance Test efforts.

- Test Plan (this document).
- Test Procedure document for each phase
- System Test Report for each phase

2.11 **Test Schedule**

The Test schedule is available in the Project plan
2.12 Test Status Reports

Status Reports summarizing test activities; including test progress, pass/fail information and IRs will be generated at a minimum weekly and e-mailed to all involved teams. The information for the reports will be generated using Clear Quest and maintained by Venky Varadarajan. These reports will be located in the Test Process Documentation directory.

2.13 Defect Reporting

Any defects identified during testing of the application that will be submitted as an incident report in Clear Quest with an appropriate classification of importance. The Clear Quest CTMS project database will be used to track defects. Intranet web page at the following link for the IR Reporting procedures: http://toerat/cqweb/logon/default.asp

2.14 Defect Classification

Defects are classified as below
1. Critical
   a. No Workaround
   b. No further testing can be done
   c. Show stopper

2. Major
   a. Possible Workaround
   b. Major system functionality

3. Average
   a. Possible Workaround
   b. Normal system functionality yet important

4. Minor
   a. Possible Workaround
   b. Minor system functionality

Schedule for defect planning and fix

<table>
<thead>
<tr>
<th>Type</th>
<th>Plan</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>1 hour</td>
<td>4-12 hours</td>
</tr>
<tr>
<td>Major</td>
<td>8 hour</td>
<td>1 week</td>
</tr>
<tr>
<td>Average</td>
<td>1 week</td>
<td>3-4 weeks</td>
</tr>
<tr>
<td>Minor</td>
<td>End of Construction</td>
<td>Next Construction</td>
</tr>
</tbody>
</table>
2.14.1 Defect Tracking State model

Defect Tracking State Model

- NEW
  - assign
  - Triage
- ASSIGNED
  - assign
  - Triage
- OPENED
  - open
  - resolve
  - re-assigned
  - TEST MANAGER
  - Tester validates and closes or rejects
- RE-ASSIGNED
  - assign
  - TEST MANAGER
- RESOLVED
  - resolve
  - reject
  - duplicate
- CLOSED
- DUPLICATE

Ability to:
- modify (all)

Test Manager reviews and assigns to Developer
Developer fixes bug & resolves
Tester validates and closes or rejects
### 2.15 Test code build

As per development plan, automated scripts will be set to build the code package every 4 hours. Each build will be uniquely labeled.

There will be separate builds available for the Client side and Server side codes. The server code will be built first; if server fails the client code will fail too. In case of failure the assigned personnel will be posted an email by the system and rectification process will start from there.

The client side code will be installed on the Desktop of the testing machine and server side code will be installed in the Test server.

The code will be packaged with the property files to target the intended environment.

Once the code at Test environment is tested and approved for migration, the same build will be moved to the Staging for Operational testing.

At staging the code will be tested for operational readiness.

On approval the same code build will be moved to production.

Every code build that to be tested will accompany a release document, which will details the defects, CR, ER fixed and this document will relate the Release number and the Build label.

Every migration will accompany a release document with the details of the migration.

### 2.16 Requirements Traceability

Each Test Case will be designed to represent a Use Case and will include a reference to that Use Case. All Use Cases will be represented by correlating Test Cases to ensure test coverage.

Please see Appendix A for Use Case traceability and System Test Coverage.

### 2.17 Testing Methodology and Tools

The Test plan will obtain the requirements from Rational RequisitePro, Rational Rose models and use cases. The Test planning will be done through the Rational Test Manager.

Rational Test manager test cases will be integrated with the requirements in Requisite pro and Rational Rose. This provides the requirement tracking and ensures the application and testing covers all the requirements and also the changes to the requirement are tracked.

Functional testing will be done in two modes – Manual and Automated.

For the Manual testing, scripts will be written through the Rational Manual test and the automated testing the scripts will be recorded through the Rational Robot. These tools are integrated with Test Manager.

Test plan will be divided based on the type of testing and the test cases are grouped by the UseCases and stored in the folders with meaning full names.

Rational Test Manger will be used for the test execution, test logs and the test reports and test metrics.

The defects that may be found during the testing will be added to Clear Quest through the TestManager.
Rational PureCoverage will be used for the Code Coverage check.

For the full list of tools used in this project please refer to the Tools.doc in the CTMS Project Repository.

3 TEST DATA SELECTION CRITERIA

3.1 CTMS Data Selection

The CTMS Test Cases will be initiated by simulation of the user function to cover the various scenarios that affect CTMS functionality in production, and will reflect the data requirements specified in the test cases.

4 TESTING ENTRANCE, SUSPENSION AND EXIT CRITERIA

In order to perform system test in an efficient and timely manner, certain systems development activities and deliverables are required before entering each of the various phases of system test. Particular criteria have been defined to ensure that an acceptable level of system integrity has been achieved. Listed below are the necessary deliverables required for the entrance into the test-execution test phase:

4.1 Entrance Criteria

The following requirements must be met before the System Test Execution Phase can begin:

<table>
<thead>
<tr>
<th>Entrance Criteria</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>The completed Detailed Requirements Specifications and Architecture Design Design documents have been reviewed and signed off.</td>
<td>CTMS Business Analyst, CTMS Development</td>
</tr>
<tr>
<td>Unit testing have been completed successfully by development and issues if any resolved.</td>
<td>CTMS Development</td>
</tr>
<tr>
<td>An appropriate and separate test environment has been set up; all hardware and software required for the system test effort, as specified in Section above, has been installed and verified.</td>
<td>CTMS System Tester, CTMS Development</td>
</tr>
<tr>
<td>Acceptance testing of the test environment configuration for the application and database has been completed.</td>
<td>CTMS System Tester</td>
</tr>
<tr>
<td>Database structure/schema are approved and tested.</td>
<td>Database Analyst</td>
</tr>
<tr>
<td>A listing of the delivered functions has been delivered as Release Notes (per iteration).</td>
<td>CTMS Development</td>
</tr>
<tr>
<td>No outstanding ERs are pending for this delivery, in this release.</td>
<td>CTMS Development</td>
</tr>
</tbody>
</table>

4.2 Suspension Criteria

This section contains the conditions associated with the CTMS applications that would cause a suspension of testing. They include:

- Inability to access the applications.
- Inability to run primary functions of the applications.
- Inability to connect with required interfaces.
The identification of showstopper defects that will require code, data-content or configuration changes and retest.

The existence of open, un-repaired Severity Critical IRs.

4.3 Exit Criteria

The following criteria must be met in order to determine successful completion of System Testing:

<table>
<thead>
<tr>
<th>Exit Criteria</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Test Scenarios identified have been executed successfully.</td>
<td>CTMS System Tester:</td>
</tr>
<tr>
<td>All opened Defect reports (IR’s) for the application have been classified for correction or acceptance, including acceptance sign-off by the test team and sign-off for all work-arounds.</td>
<td>CTMS System Tester:</td>
</tr>
<tr>
<td>All required test scripts, test results, Defect reports, application configuration and database information for test scenarios have been backed up for archival purposes.</td>
<td>CTMS System Tester:</td>
</tr>
<tr>
<td>Delivery &amp; Release Notes document written and delivered.</td>
<td>CTMS System Tester:</td>
</tr>
<tr>
<td>Sign-off Delivery &amp; Release Notes</td>
<td>CTMS System Tester:</td>
</tr>
</tbody>
</table>

5 Testing Environment

5.1 Testing Environment

The testing of the CTMS will be conducted in the CTMS Testing Environment. Configuration changes will be made to this environment, as needed, in order to incorporate the CTMS application into the environment. For the detailed operating environment details please refer to the document “CTMS/CTIS Technical Operating Environment”

A high-level, physical depiction of the CTMS, as a fully integrated, end-to-end environment operating in conjunction with all of the major applications supporting the CTMS Integration, is provided in Figure CTMS Test Environment, located on the last page of this document.

5.2 Required Hardware and Software

<table>
<thead>
<tr>
<th>Server</th>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOCRHTST01 10.82.60.68</td>
<td>JBOSS</td>
<td>3.2. Service Pack 1</td>
</tr>
<tr>
<td>TOCUXTST01 10.82.60.69</td>
<td>Linux</td>
<td>Red Hat Linux 2.6.8-1.521</td>
</tr>
<tr>
<td></td>
<td>Ant</td>
<td>1.6.1</td>
</tr>
<tr>
<td></td>
<td>Xdoclet</td>
<td>Most Recent Version</td>
</tr>
<tr>
<td></td>
<td>Java JDK 1.4.2.03</td>
<td>1.4.2.03</td>
</tr>
<tr>
<td></td>
<td>Quartz</td>
<td>Most Recent Version</td>
</tr>
<tr>
<td></td>
<td>Java COMM API</td>
<td>Most Recent Version</td>
</tr>
<tr>
<td></td>
<td>ClearCase Client</td>
<td>Most Recent Version</td>
</tr>
<tr>
<td></td>
<td>JMQ, JMX, JCA</td>
<td>Most Recent Version</td>
</tr>
<tr>
<td></td>
<td>Digiport drivers</td>
<td>Most Recent Version</td>
</tr>
</tbody>
</table>
### 5.3 Test Database Instances

The following test database instances will need to be configured for CTMS testing (will be updated):

- **Server** = TOCDBDEV-10.82.70.10
- **SID** = xxxx
- **Schema** = CTMS1.0_dba
- **HP-UX 11.11**
- **ArcSDE Spatial server**
- **Oracle9i**
6 TEST PLAN SIGNOFF

Signature of this test plan document confirms that you have reviewed this Test Plan for CTMS and accept the testing approach described in the test plan as the appropriate level of systems testing for this application. Signature does not indicate approval of the test cases in the Test Case Document.

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert Wycoff</td>
<td>Network analyst</td>
<td>ITS</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carry Weiss</td>
<td>Project Management</td>
<td>ITS</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>John Williams</td>
<td>Project Management</td>
<td>ITS</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pawan Kharbanda</td>
<td>Project Lead Software, Design</td>
<td>EnRoute TS Inc</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Venky Varadarajan</td>
<td>Systems Tester</td>
<td>EnRoute TS Inc</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 CTMS Test Environment
8 Appendix A

This Appendix will be a live document. This Section list the data related to the Elaboration Phase 1. As the project progress the Appendix will be updated accordingly

"CTMS1.0 Test Plan-Appendix A.xls"