OUM

TE.010 Testing Strategy

AmeriTel Inc

GREXIT Mitigation

Author:	John Needham
Creation Date:	May 8, 2014
Last Updated:	September 14, 2015
Document Ref:	<Document Reference Number>
Version:	DRAFT 1A

Approvals:

<Approver 1> ______________________________

<Approver 2> ______________________________

1 Document Control

1.1 Change Record

Date	Author	Version	Change Reference
8-May-14	John Needham	Draft 1a	No Previous Document

1.2 Reviewers

Name	Position

Contents

2 Introduction

This document provides the Testing Strategy for the Greek Finance System system.

The Greek Finance System Testing Strategy determines the project's approach to testing. The strategy looks at the characteristics of the system to be built, the project time line and budget, and plans the breadth and depth of the testing effort. The Testing Strategy influence tasks related to test planning, test types, test script development, and test execution.

2.1 Background

Over the Easter Weekend in April 2015 Greece declared itself out of the Eurozone project. As a result of the Greek exit (GREXIT) there will be an emergency Oracle project to restructure the Greece Oracle footprint – GREXIT Mitigation. This project started immediately and aims to facilitate normal business trading as soon as possible. The Greek exit has many wide-ranging business and system implications, not all of which are clear right now – however the key near-term objectives have been identified.

2.2 Objectives

The key objectives of the Testing Strategy are as follows:

- Determine the testing approach for the various tests described in the Testing Requirements
- Support the business objectives for the new GREXIT Mitigation project.
- Create a new, integrated testing that includes both existing applications and the new application.
- Build testing that allows for expandability, flexibility, and support of future business requirements.
- Determine the types of tests required by each testing task.

2.3 Work Product Audience

The Greek Finance System Testing Strategy is intended for the following audience:

- AmeriTel Inc and Oracle analysts and designers
- AmeriTel Inc and Oracle testers
- conversion and interface teams
- operations

2.4 Benefits

The Testing Strategy can provide the following benefits:

- availability of human, time, hardware, software, testware resources due to early test planning
- faster development of testing requirements by directly using key project work products
- earlier identification of testing requirements
- adequate test types adjusted to the project needs
- independence of testing from development tasks and resources
- well-defined tests
- progressive and additive test tasks

3 Scope

Below is the testing scope of Greek Finance System detailed through the following items:

- testing tasks
- test types by task
- system interfaces

3.1 Testing Tasks

This project includes the following testing tasks:

- Unit Test
- Use Case Test
- Installation Test
- System Test and Full System Test
- Systems Integration Test
- Acceptance Test

3.2 Test Types by Task

The following list identifies, by testing task, the types of testing that will be conducted:

Unit Test

process step
validation
calculation
error handling
database auditing
security
volume data
help text
checkpoint restart
user interface
report layout
screen layout

Integration Test

use case
service
security
volume data

Installation Test

The following type of information you need to confirm during installation testing will influence the scope:

Installation Routines

(Full) System Test

initial system documentation
manual data load
system process sequence using scripted data

service integration
interface using scripted data
converted data load
converted data inspection
system process sequence using converted data
interface using converted data
parallel legacy reconciliation
job stream
backup and recovery
database auditing
data archival
security
locking
batch response time
online response time

Systems Integration Test

systems integration process sequence using converted data
network stress
security
locking
batch response time
online response time

Acceptance Test

business processes
use cases
batch response time
online response time
parallel running
live data
live environment
final system documentation sign-off

3.3 System Interfaces

The following table identifies key system interfaces that will be integral to the testing of Greek Finance System:

System Interface Name	**Type (input,output, two way)**
Bank Payments	Output
LockBox Receipts	Input
Bank Statements	Input
XML Self-billing Invoices	Output

4 Approach

In the GREXIT Mitigation project, test results should be accepted whenever the system performs in a way that meets the business needs. In order to achieve a business-oriented measure of fitness for purpose, the emphasis in testing is placed on whether the business process, with its supporting application system, meets the objectives stated in Business and System Objectives (RD.001).

Another important principle that will be used for testing is the principle of continuous integration. This means that use case tests will be performed as part of the iterations, and a system test at the end of each iteration.

4.1 Test Scripts and Scenarios

Test Scripts will be used for Unit Tests and Test Scenarios for Integration Tests, System Tests, and the Systems Integration test will be produced and used for each test. The scripts and scenarios will evolve through the project as the system requirements stabilize through the iterations.

The developer will produce test scripts for the unit tests before the actual code is produced and the Integration Test Scenarios will be produced before the components are developed that implement the use case.

4.2 Approach per Test Task

Unit Testing

The following types of information you need to confirm during unit testing will influence the scope:

- System Process Step
- Validation
- Calculation
- Error Handling
- Database Auditing
- Security
- Volume Data
- Help Text
- Check-point Restart
- User Interface
- Report Layout
- Screen Layout

System and Integration Testing

The following types of information you need to confirm during system testing will influence the scope:

- Initial System Documentation
- System Process Sequence using Scripted Data
- Interface using Scripted Data
- Interface to Application Extensions
- Parallel Legacy Reconciliation
- Job Stream
- Backup and Recovery
- Database Auditing

- Data Archival
- Security
- Locking
- Batch Response Time
- Online Response Time
- Regression Testing
- Network stress
- Security
- Locking
- Batch response time
- Online response time
- Interfaces to third-party and legacy systems

Data Conversion Testing

- Manual Data Load
- Converted Data Load
- Converted Data Inspection
- Converted Data Reconciliation
- System Process Sequence using Converted Data
- Interface Using Converted Data
- Systems integration process sequence using converted data

Acceptance Testing

- Business Processes
- Batch Response Time
- Online Response Time
- Live Data
- Live Environment
- Final System Documentation
- Sign-off

5 Test Data

This section describes what kind of test data will be needed as part of the tests, how it will be obtained, and for which tests it will be required and in which testing environments.

5.1 Converted Data Sources

The following table identifies legacy or other sources of converted data that will be used for testing:

Legacy System or Source Name	Description of Converted Data	Needed for (when)	Load in Test Environments
<ABC System>	<Accounting information>	<Construction phase, 1st iteration>	<Use Case Test Environment, System Test Environment>

5.2 Other Test Data

In addition to the converted data sources above, the following test data will be needed. Some of these are needed only until the converted data has become available:

Type of data	Description	When	Environment	How to produce	Volume	Con
<PERSON>	<ID,NAME,GENDER,AGE,…>	<Elaboration phase, 2nd iteration>	<Use Case Test Environment >	<Select from IDP system, manually adjust>	<100>	<Yes>

6 Constraints

The project must operate within the following limits:

6.1 Time

Testing tasks will be constrained by time, affecting the following activities:

- the ability to setup and maintain test data as required by test scripts and sequences
- the ability to deliver stable manual data
- the turnaround time necessary for application software fixes
- the turnaround time necessary for converted data fixes
- the time required to run batch modules

6.2 Required System Resources

The Greek Finance System testing effort is restricted by the availability of the following system resources:

- Network capacity
- Available testing environments
- Available test data

6.3 Business

The Greek Finance System testing effort is restricted by the following business policies:

- Day-to-day workload of testers
- Overtime restrictions
- Budgetary constraints

6.4 Technical

The Greek Finance System testing effort is restricted by the following technical constraints:

- Adequate testing environments
- Systems integration
- Hardware compatibility

7 Testing Environments and Testing Tools

- Fusion Test Instance
- Test Director – test platform

7.1 Testing Environments

The following table documents the testing environment criteria for each testing task:

Testing Task	Platform	Software	Server Environment
Unit Test	Development	Oracle Fusion Financials	Cloud
Use Case Test	Test	Oracle Fusion Financials	Oracle Fusion Financials
System Test	Test	Oracle Fusion Financials	Oracle Fusion Financials
Full System Test	Conversion	Oracle Fusion Financials	Oracle Fusion Financials
Systems Integration Test	Test	Oracle Fusion Financials	Oracle Fusion Financials
Acceptance Test	Conversion	Oracle Fusion Financials	Oracle Fusion Financials

7.2 Testing Tools

The following testing tools will be made available:

Testing Tool	Software vendor	Purpose
Test Director	HP Quality Centre	Manage the test scripts and provide status / progress reporting.

8 Acceptance Criteria

Predefined acceptance criteria should be established prior to testing. These criteria are used as a standard to measure the success of each test. Once these predefined acceptance criteria are met, the process owners can and should feel confident in signing an acceptance certificate for each deliverable.

The overall exit criteria for the testing phase are:

- Major business process has been validated
- Major outputs/reports have been produced
- No Severity 1 or 2 issues remain
- For all Severity 3 and 4 issues, workarounds have been documented and signed off
- Test completion/success above 95% (volume)
- All known defects have been documented in the defect tracking system
- All Stakeholders have signed off on the test results

8.1.1 Defect Severity Definition

8.1.1.1 Severity 1 (Critical)

Profile:

- Produces fatal errors
- No workaround exists
- User will not be able to perform primary functions
- System cannot be deployed to production environment

Testing Impact:

- Unable to perform basic functions
- Scripts / Test cases for a particular High or Medium-frequency business scenario cannot be completed
- Testing progress is prevented

Response time: 4 hour timeframe to provide an initial response and provide a resolution plan.

8.1.1.2 Severity 2 (High)

Profile:

- Produces fatal errors in a major function
- No workaround exists for that function
- User will not be able to perform that function
- System cannot be deployed to production environment

Testing Impact

- Testing can continue, but a major portion cannot be tested
- Scripts / Test cases for a particular High or Medium-frequency business scenario cannot be completed

Response time: 8 hours

8.1.1.3 Severity 3 (Medium)

Profile:

- System produces misleading/incorrect information
- A workaround exists, but the solution is undesirable
- System deployment to production environment is questionable.

Testing Impact

- Testing can continue however other tests may be impacted.

Response time: feedback within 2 days

8.1.1.4 Severity 4 (Low)

Profile:

- Workaround exists
- Users may have difficulty using features as a result of the workaround

Testing Impact

- Scripts can be completed by completing the objective with different steps
- Results from the scripts do not exactly match the expected result

Response time: feedback within 3 days

8.1.2 Defect Status

New Initial status. Defect has been logged but not reviewed.

Assigned Defect has been reviewed in defect meeting and assigned for resolutions to an owner.

Closed No issues outstanding, defect has been resolved satisfactorily.

9 Problem Management

The assessment and prioritization of defects found during testing will be strictly controlled using the PJM Issue and Problem Management process. A detailed discussion of problem management is found in the PJM documentation. Problem management should be adopted at the overall GREXIT Mitigation project level. Therefore, the testing process should use the same problem management process that each of the other OUM processes use.

10 Critical Success Factors

In addition to the GREXIT Mitigation overall critical success factors, the following critical success factors are specific to the Testing process:

- Testing considerations must begin in the early phases of the project, and be an integrated part of the project process.
- Test script development must be based on system requirements and future business processes and business use cases.
- Testing must be objective and must be performed by an independent test team (other than the programmers responsible for the application software).
- The problem management process must be functional as soon as testing begins
- Check that only valid and non-duplicated defects are processed.
- Multiple iterations for each testing task should be planned to allow for a higher density of testing for the current test iteration and scheduled fixes for the next iteration.
- Planning for the systems integration test should start early, as it will involve multiple projects, systems, and organizations.
- Use an automated tool should be used to perform business system testing.
- Locking, response time, and stress testing should use process-based testing scripts.
- Components should be categorized by their relative importance to the business for defect prioritization and performance testing.
- Strong executive sponsorship and management support of the project mission and project team is needed.
- Adequate project staffing for the expected goals and timeline are to be met
- Clear roles and responsibilities must be defined for the project in order to assure accountability, ownership, and quality.
- A committed and well-informed testing project manager and project team should have a thorough understanding of the project goals, milestones, and any relationships with other implementation projects.
- A comprehensive project workplan and quality management approach.
- A defined and maintained project infrastructure throughout the project duration.
- A thorough understanding of known project risks and assumptions by business and project management.
- Early testing, planning, and coordination across all AmeriTel Inc division, subsidiaries and associated corporations to maximize timely and cost-effective enterprise wide testing and migration development investments
- Representatives from each testing group are part of the project team to facilitate consideration of enterprise-wide performance quality management objectives.

11 Risks and Contingency Plans

Identified testing risks and their associated contingency plans include the following:

Software Release Versions and Availability

The combination of Third Party Systems and Oracle will need to be carefully coordinated to ensure that the production environment is faithfully mirrored for testing.

Strategic Testing Decisions

There is a risk that if certain process areas are left out of testing the final configuration may not be fully validated.

Resource Availability

Due to the distributed nature of some of the processes and conflicting demands from the day-to-day running of the business there is a risk that ISAC and local business resources may not be available for testing.

Converted Data

Programmatic Data conversions are being developed off-site and there is a risk that communications may be more complex and defect turnaround times elongated.

12 Testing Metrics and Measures

One of the outputs from the testing process is metrics and measures related to testing. For each test, a number of metrics and measures will be collected.

The following metrics and measures will be collected:

Example testing metrics are as follows:

- number of test iterations planned for each test task
- number of developers test programs
- number of developed classes
- percentage of total number of classes unit tested
- percentage of total use case scenarios tested in use case test
- relative importance of the application system to the business
- complexity of the application system under test
- number of functional areas involved in the unit and use case test
- percentage of completed unit and use case tests
- number of business processes
- number of scenarios per business process
- number of test steps per scenario
- complexity of the component under test
- complexity of the scenario under test
- number of other application systems involved in the systems integration test
- required online response time for critical components
- batch nightly window response time
- number of end users required for the stress test
- number of defects per test, period, iteration,…