cid:image007.png@01CEC010.A7507240

NATIONAL CANCER INSTITUTE

CENTER FOR BIOMEDICAL INFORMATICS AND INFORMATION TECHNOLOGY (CBIIT)

*Greensheets Re-design Release*

SYSTEM TEST PLAN

Version 1.0

Revision History

The table below identifies all changes that have been incorporated into this document. Baseline changes require review and approval.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 6/21/2017 | 1.0 | Initial document | Elizabeth Andreyev |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc425761155)

[1.1 Purpose 5](#_Toc425761156)

[1.2 Objective 5](#_Toc425761157)

[1.3 Testing Scope 5](#_Toc425761158)

[2. Resources 5](#_Toc425761159)

[2.1 Test Team 5](#_Toc425761160)

[3. Test Inclusions 6](#_Toc425761161)

[3.1 Features to be Tested 6](#_Toc425761162)

[4. Test Methodology 6](#_Toc425761163)

[4.1 Test Phases 6](#_Toc425761164)

[4.1.1 System Testing 6](#_Toc425761165)

[4.1.2 Regression Testing 6](#_Toc425761166)

[4.1.3 Issue Severity Levels 6](#_Toc425761167)

[4.2 Test Tasks 7](#_Toc425761168)

[4.3 Test Environment 7](#_Toc425761169)

[4.3.1 Client Hardware 7](#_Toc425761170)

[4.3.2 Servers 8](#_Toc425761171)

[4.3.3 Operating Systems 8](#_Toc425761172)

[4.3.4 Hosted Applications 8](#_Toc425761173)

[4.3.5 Domain 8](#_Toc425761174)

[4.3.6 Software 8](#_Toc425761175)

[4.3.7 Database Management System 8](#_Toc425761176)

[4.3.8 Client Software Configuration 8](#_Toc425761177)

[4.4 Testing Types 8](#_Toc425761178)

[4.4.1 Functionality Test 8](#_Toc425761179)

[4.4.2 Regression Test 8](#_Toc425761180)

[4.4.3 Negative Test 8](#_Toc425761181)

[4.4.4 Ad Hoc Test 8](#_Toc425761182)

[4.4.5 Browser Compatibility Test 9](#_Toc425761183)

[4.4.6 Usability Test 9](#_Toc425761184)

[4.4.7 User Acceptance Test 9](#_Toc425761185)

[4.5 Test Documentation 9](#_Toc425761186)

[4.5.1 Requirements Traceability Matrix 9](#_Toc425761187)

[4.5.2 Test Scripts 9](#_Toc425761188)

[4.5.3 Issues 10](#_Toc425761189)

[4.5.4 Test Results 10](#_Toc425761190)

[4.6 Test Entrance Criteria 10](#_Toc425761191)

[4.7 Test Exit Criteria 10](#_Toc425761192)

[4.8 Suspension Criteria and Resumption Requirements 11](#_Toc425761193)

[4.8.1 Suspension Criteria 11](#_Toc425761194)

[4.8.2 Resumption Requirements 11](#_Toc425761195)

[5. Test Deliverables 11](#_Toc425761196)

[5.1 Test Logs 11](#_Toc425761197)

[5.2 Test Cases 11](#_Toc425761198)

[5.3 Test Plan 11](#_Toc425761199)

[5.4 Test Evaluation Summary 11](#_Toc425761200)

[5.5 Test Data 12](#_Toc425761201)

[6. Responsibilities 12](#_Toc425761202)

[7. Schedule 12](#_Toc425761203)

[8. Risks 12](#_Toc425761204)

[9. Contingences 12](#_Toc425761205)

[Appendix A - Acronyms and Abbreviations 13](#_Toc425761206)

[Appendix B – Requirements Traceability Matrix 14](#_Toc425761207)

# Introduction

## Purpose

The purpose of this Test Plan is to support the testing effort for the Greensheets Re-design Release in accordance with the Standard Software Testing Procedure.

This document represents a clear and concise test strategy for each type of test to be implemented and executed. All the resources needed to successfully implement and execute testing will be identified, including hardware, software, and personnel. In addition, the test plan will identify the artifacts created by the test activities, when the artifacts will be made available, how they will be distributed, their content, and how they should be used.

## Objective

The primary objective of the system test is to provide assurance that the functionality of the Greensheets system is developed and works according to the customer defined requirements, and meets or exceeds user’s expectation. The secondary objective of the system test is to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all high and medium severity issues are addressed in an appropriate matter before release. The System Test Plan supports the following objectives:

* Detail the activities required to prepare for and conduct system test.
* Communicate to all responsible parties the tasks that are to be performed and the schedule to be followed in performing these tasks.
* Define the sources of the information used to prepare this plan.
* Define the test tools and environment necessary to conduct system test.

## Testing Scope

The GREENSHEETS RE-DESIGN system test will be executed under the Windows operating environment with the security group policy objects implemented.

System test will verify that the functionality works according to the specifications. The execution of test scripts will verify requirements and will verify the resolution of the issues.

Tests will be performed at the Functional and Non-Functional levels that include Smoke and Integration tests. The requirements for testing will be derived from the use-cases, business rules document, software requirements specification, design requirements, business cases and software architecture documents.

In addition Regression testing will be conducted at the end of each Sprint to verify that none of the existing functionalities was affected by the new changes made to the system. System testing will be conducted under the Windows 7 environment.

# Resources

The Test Team develops the System Test Plan, executes the Test Scripts, and develops the Test Evaluation Summary Report.

The Test Team is familiar with software testing practices and methodologies, including test planning and documentation, and error reporting.

## Test Team

All test planning and execution will be performed by the Test Engineers and Business Analysts; if necessary, additional testers will be assigned. The Test Team members and their responsibilities are identified in Table 1.

Table 1: Test Team Functions and Responsibilities

|  |  |  |
| --- | --- | --- |
| Position | Name | Function |
| Application Tester | Elizabeth Andreyev | Create and execute test scenarios  Create System Test plan |
| Business Analyst | Gaby Tulchinskaya | Prepare test scenarios for the UAT based on the functional test cases created by the Test Team |
| Application Tester | Sami Rashid | Create and execute test scenarios |

# Test Inclusions

## Features to be Tested

Software Requirements Specification (SRS) document provides a list of the features that are to be included. These items are to be tested during the course of the system test and the user acceptance test.

# Test Methodology

## Test Phases

The Test Team will conduct system testing that includes the execution of test scripts written to verify requirements defined in the Software Requirements Specification (SRS) document for this release. In addition Regression testing will be conducted at the end of each Sprint to verify existing functionality has not been negatively impacted by the new changes. System testing will be conducted under the Windows 7 environment.

### System Testing

The system test will verify the code implementation of requirements defined in the SRS has been met. A set of test scripts will be created and traced to the requirements defined in the SRS. These test scripts will contain detailed steps to verify the requirements included in this release.

Method of Analysis: The detailed test scripts will be executed to ensure that expected results are achieved. All test verification points will be supported by screen prints captured during test execution.

### Regression Testing

The regression test will demonstrate that the changes to the software did not have any negative impact on system functionality.

Method of Analysis: The results of this testing will be analyzed by ensuring that the functions executed at each point of the workflow are performed correctly and the business objective of the GREENSHEETS RE-DESIGN is met. This is accomplished by checking that any data created or entered by a specific functional task is available to other functional areas. In addition, the Oracle database is checked to verify that transactions are recorded correctly in the database.

### Issue Severity Levels

* Critical—System capability significantly degraded or the potential exists for serious degradation, serious impact of immediate concern. This priority is used for problems that have a significant or serious effect on the system and impede system functionality. This priority indicates that the problem is preventing the user from performing functions.
* Major—System capability affected but no serious degradation in performance or usability, moderate impact not of immediate concern. This priority is used for problems that are significant, but have a workaround, or they do not significantly affect the system operations. This priority applies to Issues that do not interfere with current processing or to maintenance procedures that are not time dependent.
* Minor—Inconvenience or annoyance, lack of user friendliness, no significant impact on system operations. This priority is used for “nice-to-have” changes to the system with little significant benefit. This priority applies to optional task requests or system enhancements. This priority is also used for changes deferred for later review, resolution, or incorporation.

## Test Tasks

The task dependencies and the team responsible for each task are listed in Table 2.

Table 2: List of Testing Tasks

|  |  |  |
| --- | --- | --- |
| Task | Predecessor Tasks | Responsibility |
| Complete System Test Plan and Procedures | Requirements completed by Development Team | Test Team |
| Complete Test Scripts | Requirements completed by Development Team | Test Team |
| Conduct Test Readiness Review (TRR) | Test Plan completed | Test Team  Project Team |
| Deliver GREENSHEETS RE-DESIGN to the Test team | Complete development tasks | Software Development group  Database Management group |
| Execute system test scripts | Software delivery completed | Test Team |
| Capture test results during test execution | Software delivery completed, Database setup completed | Software Development group  Database Management group  Test Team |
| Document test summary results | Test execution completed | Test Team |

## Test Environment

The following environments will be maintained and utilized during the development life cycle of the GREENSHEETS RE-DESIGN project:

* Development Team maintains a standard, controlled DEV environment at National Cancer Institute (NCI) facility for use in deploying new functionalities and performing unit testing:

<https://i2e-dev.nci.nih.gov/greensheets/>

* System testing of GREENSHEETS RE-DESIGN will be conducted in the TEST environment, after the Development Team notifies the Test Team that a new software build has been deployed and ready for testing:

<https://i2e-test.nci.nih.gov/greensheets/>

* UAT will be performed in the STAGE environment after all the GREENSHEETS RE-DESIGN Release v1.x requirements have been verified in TEST and all of the high and medium severity defects have been resolved:

<https://i2e-stage.nci.nih.gov/greensheets/>

### Client Hardware

All system tests will be conducted at NCI. The test Personal Computers (PCs) will be provided by NCI, and meet the defined configuration for a production PC. A hardware configuration that satisfies GREENSHEETS RE-DESIGN requirements will be used. GREENSHEETS RE-DESIGN will be tested on a standard Windows 7 desktop client in the National Institute of Health (NIH) domain.

### Servers

Application server

Database Server

### Operating Systems

Linux

### Hosted Applications

Unicode supported Oracle 11g

### Domain

NIH

### Software

N/A

### Database Management System

A new database will be created for the GREENSHEETS RE-DESIGN testing

### Client Software Configuration

The software configuration of each test PC will be the standard PC configuration in production. (e.g., Windows 7, MS Office 2010, Internet Explorer 10.0 and higher, Safari 5.1.x, Chrome v59.x, Firefox v52.1.x).

## Testing Types

During a test cycle several forms of test scripts will be executed. These test types are defined below.

### Functionality Test

Functional testing will focus on any requirements traced directly to use cases or business functions and business rules. Function tests will verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box a technique; that is, verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analysing the output or results. Functional testing will verify functionality, including navigation, data entry, process and retrieval. Each use case will be validated to ensure that the expected results occur when valid and invalid data is used; that appropriate error or warning message are displayed when invalid data is used and that each business rule has been properly applied. Back end testing will be performed where applicable to ensure data integrity.

### Regression Test

Regression testing will verify that pre-existing system functionalities still work as designed. This testing will consist of entering predefined data into the system to ensure that the system behaves as expected. A select set of functionality of the system will be regression-tested using manual testing.

### Negative Test

A negative test is a situation that the system must not allow or one in which an error message will be the expected result.

### Ad Hoc Test

Ad hoc or free play testing is used in situations to check out “what happens if...” circumstances. This testing allows the testers to test using undocumented procedures. Ad hoc testing is usually used to test low risk functionality, such as system components that have not changed. Ad hoc testing is also used to validate software upgrades, which only require a spot check to verify that the upgrade is evident in the test environment.

### Browser Compatibility Test

The GREENSHEETS RE-DESIGN will be tested in Internet Explorer v11 as well as in Firefox 52.1.x, Chrome v58.x and Safari 5.1.x browsers to ensure compatibility.

### Usability Test

Development team will create a non-functioning prototype of the UI components to evaluate the proposed design. Usability testing will be coordinated by testing, but actual testing will be performed by non-testers during the UAT phase. The accessibility testing will ensure that the GREENSHEETS RE-DESIGN is complaint with the 508 Section of the Web Content Accessibility Guidelines (WCAG).

Both usability and accessibility testing will discover how easily people can use the system and feed that information back into improving future designs and implementations.

### User Acceptance Test

During User Acceptance Test (UAT), Office of Grants Administration (OGA) staff will execute workflows and tasks in a STAGE environment. This allows for frequent inspection of the incremental product and provides multiple opportunities for correction.

## Test Documentation

### Requirements Traceability Matrix

The Requirements Traceability Matrix (RTM) is available as a reference document. The RTM is used so that the test requirements are tracked as part of the overall project requirements management process and maps requirements to test cases, and more specifically, to the individual test step that verifies the requirement.

To do this integration, the project maps the test requirements to each item to be tested, identifies the corresponding test verification (qualification) method, and then prioritizes the tests.

### Test Scripts

Test scripts are written with the steps necessary to properly test that the system meets the requirements. Test scripts are written so that a “non-technical” individual could run the test scripts with only basic instructions. This means that the individual steps and expected results are in plain language.

Individual test scripts are developed to test specific functions or functional areas of a system. A test script can also be developed for each Issue that is assigned to the release when it is more effective to do so than to update an existing test script to test the Issue. A comprehensive test script that includes all the procedures to fully test the Issue often requires extensive research. Meetings with the developers are often necessary to achieve a complete understanding of the system changes that were made and which system components are affected by the changes.

Once written, each script is informally peer reviewed. If omissions are detected, scenarios are revised and reviewed again as necessary. Scripts that are updated subsequent to the peer review are peer reviewed again. Often, more information is available during and upon completion of software integration testing, which requires the scenarios to be altered or enhanced.

The basic structure of the test scripts consists of a header page that contains the name of the individual test script, along with the specific purpose of the test script, test engineer contact information, basic system information (if needed), a large section for notes regarding the test script, and most importantly, a section to list any prerequisites and requirements necessary to properly run the test case. The remainder of the test script is set up to include a step number, the inputs or instructions for the step, the expected results, and a Pass\Fail column. The last column can be used for comments, or if the step corresponds to a specific requirement, the requirement number. This supplements the RTM to ensure all documented requirements are tested.

### Issues

All problems discovered during the system test phase will be recorded by creating a Bug in the JIRA Issue tracking tool. All Bugs will be tracked. The Bugs will be re-tested by repeating the test procedures that caused the original failure. Depending on the Bug, it may be necessary to execute additional test scripts to ensure that the problem has been resolved. The severity of the problem, as defined in Table 3, will determine whether or not the fix and the release will pass system testing.

Table 3: Bug Severity

|  |  |  |
| --- | --- | --- |
| Severity | Description | Note |
| Critical | System capability significantly degraded or the potential exists for serious degradation, serious impact of immediate concern. This priority is used for problems that have a significant or serious effect on the system and impede system functionality. This priority indicates that the problem is preventing the user from performing functions. | Fail |
| Major | System capability affected but no serious degradation in performance or usability, moderate impact not of immediate concern. This priority is used for problems that are significant, but have a workaround, or they do not significantly affect the system operations. This priority applies to Issues that do not interfere with current processing or to maintenance procedures that are not time dependent. | Pass with acceptable workaround |
| Minor | Inconvenience or annoyance, lack of user friendliness, no significant impact on system operations. This priority is used for “nice-to-have” changes to the system with little significant benefit. This priority applies to optional task requests or system enhancements. This priority is also used for changes deferred for later review, resolution, or incorporation. | Pass with deferment to later release |

### Test Results

Once the testing phase has been completed, the Test Engineer will draft a document listing the test scripts that were performed, any ad hoc testing that took place, and any problems (Issues) that were encountered and/or fixed during the testing phase. This document will be forwarded to management with the finished test packets for review.

## Test Entrance Criteria

Before a release can be system tested, certain criteria must be met. The following lists the entrance criteria for each deliverable:

* All software unit testing and software integration testing has been successfully completed for the particular deliverable being tested.
* A TRR was conducted to validate that the system is ready to be released into the test environment.
* Peer review of the individual test scripts have been performed to ensure adequate testing.
* The software coding for the release has been frozen and all software has been checked into Subversion.
* The system administrators have moved the release into the test environment and started all necessary executables and scripts.
* The test environment has been fully built out/updated as needed based upon the release notes.

## Test Exit Criteria

The following is the list of exit criteria that must be met prior to a release deliverable being completed:

* The Pass criterion for each test script is that it has been successfully completed and no open high-severity Issues are logged against it.
* The Fail criterion for each test script is that it has not been successfully completed, and either an open high-severity Issue or a medium-severity Issue without an acceptable workaround is logged against the script.
* All testing as outlined in the System Test Plan and in the Test Scripts has been completed.
* The specific test scripts for the release are completed and/or updated where necessary.
* The test scripts are updated with any changes made during testing.
* The test results document is updated listing any failed test steps and any inconsistencies that were discovered during testing. The results will also be updated with any Issues that were created during the test cycle, as well as with any that were tested and closed.
* The JIRA Issue tracking tool is updated with new Bugs, and any Bugs that passed testing have been closed.

## Suspension Criteria and Resumption Requirements

### Suspension Criteria

The existence of a high severity Issue that will not allow the test engineer to complete the execution of a detailed test scripts will cause the suspension of all or a portion of the test activity.

All causes of suspension that are related to a JIRA ticket will require the delivery of a new GREENSHEETS RE-DESIGN software executable.

### Resumption Requirements

The resumption of test activities will require the successful execution of the detailed test scripts that caused the suspension. This will enable the Test Team to determine with some accuracy if the error that caused suspension has been fixed correctly.

The delivery of a new GREENSHEETS RE-DESIGN executable will require that all impacted system test scripts for that executable will be rerun.

# Test Deliverables

## Test Logs

Defects, issues and potential enhancements identified during testing are tracked and managed using JIRA. JIRA will be used to track and continually update problems from initial creation through final resolution.

## Test Cases

A single test case is a set of test inputs, execution conditions, and expected results developed to verify compliance with a specific Requirement. The test cases created for this project conform exactly to the requirements gathered in the use cases. Revisions to the test cases will occur only if the uses cases are revised; or changes to the software are generated during User Acceptance Testing. The finalized Test Case documents are stored in Subversion at https://ncisvn.nci.nih.gov/svn/iscs/greensheets/Test/Test Artifacts/GS Re-design Test Cases.

## Test Plan

The test plan contains information about the purpose and goals of testing within the project. It also serves to identify strategies to be used to implement and execute testing and resources needed. The finalized Test Plan document is stored in Subversion at https://ncisvn.nci.nih.gov/svn/iscs/greensheets/Test/Test Artifacts.

## Test Evaluation Summary

Test evaluation summary report presents a summary of the test results and key measures of test for objective evaluation and assessment: number of requirements/test cases executed during each test cycle, number of passed/failed test cases and corresponding defect numbers, number of defects found in each Sprint. Test Evaluation Summary document is stored in Subversion at https://ncisvn.nci.nih.gov/svn/iscs/greensheets/Test/Test Artifacts/.

## Test Data

Test data consist of sample MSWord documents, HTML, Bitmaps (\*bmp), JPEGs (\*.jpg), Acrobat (\*.pdf) files and general free flowing text files that the Test team created to assist with their System Level testing.

# Responsibilities

The Salient Test Team is responsible for test planning activities and performs all of the system test tasks, including generating test planning documents, test execution, and generating test results artifacts.

The following groups provide the support activities required to plan and execute a successful system test phase:

* Development – Responsible for providing the software for the release to be tested. Also responsible for providing fixes to the software anomalies found and documented during the test phase.
* Database (DB) Support – Responsible for ensuring that the proper database instances are available to the test team.

# Schedule

The system test schedule incorporates all of the tasks for the project and includes the test planning, scripts, execution, and test results. The master test schedule maintained by the project manager includes time for output review, results preparation, and system deployment. The schedule is updated and used to track progress throughout the testing cycle.

To obtain the current schedule dates, please refer to the overall project schedule maintained by the Project Manager.

# Risks

Risks and assumptions concerning the system test effort are listed below and, if inherited, will delay the testing effort:

If the GREENSHEETS RE-DESIGN software does not meet the normal quality standards, and then discovery of a large number of errors during system test is likely. Each error will require resolution by the development team, CM delivery of the proper baseline for testing, and verification by the test team. A large number of errors in an executable will require more passes through this cycle and additional effort on the part of the test team.

If development is late with delivering the GREENSHEETS RE-DESIGN as scheduled, the test schedule will be significantly impacted.

# Contingences

The test management staff will closely monitor the progress of test execution and the failure rate for schedule impact. Findings and observations will be addressed with project management as appropriate.

# Appendix A - Acronyms and Abbreviations

| Acronym / Abbreviation | Term |
| --- | --- |
| CM | Configuration Management |
| DB | Database |
| GUI | Graphic User Interface |
| MS | Microsoft |
| OGA | Office of Grants Administration |
| NCI | National Cancer Institute |
| PCs | Personal Computers |
| NIH | National Institute of Health |
| QA | Quality Assurance |
| RTM | Requirements Traceability Matrix |
| TRR | Test Readiness Review |
| TSR | Test Summary Report |
| UAT | User Acceptance Test |
| SRS | Software Requirements Specification |
|  |  |

# Appendix B – Requirements Traceability Matrix

| **Use Case#** | **Use Case Description** | **Test Case#** | **Use Case Status** | **Comment** |
| --- | --- | --- | --- | --- |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |
|  |
|  |
|  |
|  |  |  | Pass |  |
|  |
|  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |
|  |  |  | Pass |  |