

FSA Modernization Program
United States Department of Education
Federal Student Aid



eZ-Audit Release 1.01 Test Plan

Version 1.0

Draft

As Of
06/24/2003

Revision History

Date	Version	Description	Author
06/24/2003	1.0	Initial version.	Seth Sinclair

INTRODUCTION	1
ASSUMPTIONS	1
PURPOSE	1
TEST OBJECTIVES.....	1
PROCESS.....	1
<i>Develop Approach.....</i>	<i>1</i>
<i>Plan & Prepare Test</i>	<i>2</i>
<i>Execute Test</i>	<i>2</i>
<i>Manage Test.....</i>	<i>2</i>
<i>Operate Test Environment</i>	<i>2</i>
TEST APPROACH.....	2
TEST PHASES	3
<i>Unit Testing.....</i>	<i>3</i>
<i>R 1.01 System Testing</i>	<i>3</i>
<i>Regression Testing.....</i>	<i>4</i>
TEST CONDITIONS	5
REQUIREMENTS TRACEABILITY	5
BASELINE TEST DATA	5
TEST SCRIPTS	6
<i>Test Script Development</i>	<i>6</i>
<i>Test Scenarios.....</i>	<i>6</i>
<i>Test Script Standards</i>	<i>6</i>
SYSTEM INVESTIGATION REQUESTS (SIRS).....	6
SIR SEVERITY	7
<i>SIR Status.....</i>	<i>7</i>
<i>Re-test SIR Fixes</i>	<i>8</i>
CHANGE CONTROL PROCESS.....	8
DETAILED TEST PLAN.....	8
SYSTEM TEST EXECUTION.....	8
<i>System Test Environment</i>	<i>8</i>
<i>Test Execution.....</i>	<i>8</i>
<i>Test Passes</i>	<i>9</i>
IDENTIFICATION AND RESOLUTION OF ISSUES	9
TESTING SCHEDULE	10
<i>Detailed Test Schedule.....</i>	<i>10</i>

Introduction

Assumptions

- Testing for Release 1.01 (R 1.01) of the e-Z Audit application will not include Performance or Section 508 Compliance tests due to the “minor” nature of the release.
- Testing will not include a formal User Acceptance Testing (UAT). FSA Case users will participate in the second pass of R 1.01 System Test. R 1.01 changes will not present a significant impact for Institution Users.

Purpose

The purpose of this Test Plan is to present the testing approach for R 1.01 of the Federal Student Aid (FSA) eZ-Audit application. The Test Plan defines the business events, test conditions, test cycles, test scripts, and scenarios to be tested. The objective of system testing is to minimize risk by finding and fixing problems during the development process in order to ensure that a quality product is delivered to FSA.

This test plan provides an overall framework for testing R 1.01 of the eZ-Audit application and is not intended to provide step-by-step instructions for every aspect of the testing effort.

Test Objectives

The objectives of the eZ-Audit R 1.01 testing effort are to:

- Verify the application satisfies the business requirements and supports the eZ-Audit business processes as defined by the eZ-Audit R 1.01 requirements and use cases
- Verify that the functionality of the application performs as designed

Process

The process used to structure the test activities is illustrated in the diagram below and explained in the following sections. The eZ-Audit Test team will carry out this process.

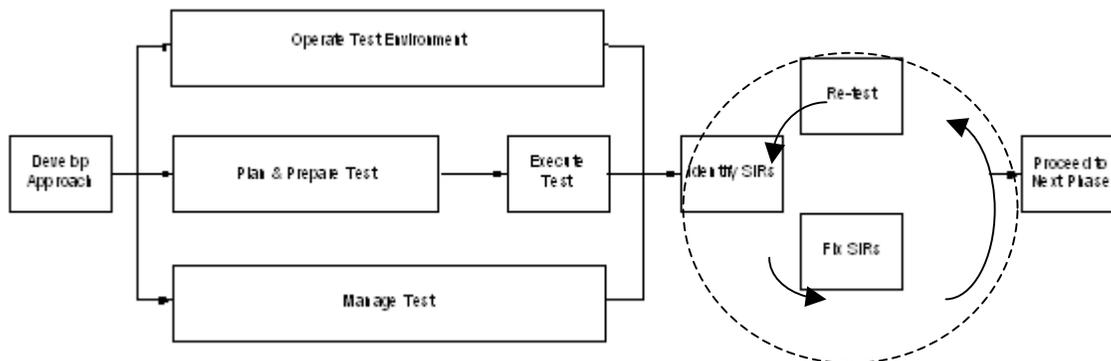


Figure 1, Test Process

Develop Approach

The *Develop Approach* phase consists of the following tasks:

- Define the test approach and detailed test plan
- Identify the test phases

- Define the scope and objectives of the test phases
- Identify the phase containment strategy and entry and exit criteria
- Identify risks associated with the test effort

Plan & Prepare Test

The *Plan & Prepare Test* phase consists of the following tasks:

- Develop the test schedule
- Identify test conditions
- Identify test scenarios
- Define expected results
- Create test scripts
- Define test data
- Verify that the conditions, scenarios, scripts, and data validate functional and technical requirements

Execute Test

The *Execute Test* phase consists of the following tasks:

- Execute test scripts for each phase
- Verify expected results
- Log System Investigation Requests (SIRs)
- Update conditions, scenarios, scripts, and data with discrepancies
- Re-test SIR fixes

Manage Test

The *Manage Test* phase consists of the following tasks:

- Manage issues
- Facilitate/coordinate test activities
- Track progress against the test schedule
- Report status

Operate Test Environment

The *Operate Test Environment* phases consist of the following tasks:

- Allocate environment resources (both human and physical assets)
- Perform configuration management activities

Test Approach

This section defines the types of testing that will take place (unit, R 1.01 system test, and eZ-Audit Regression Test). Several phases of testing will be conducted to ensure that the requirements for Release 1.01 are successfully implemented.

Test Phases

Unit Testing

The eZ-Audit Development Team will conduct unit testing to ensure that each developed code module meets its particular business needs and requirements. Unit testing will be conducted in the development environment.

A unit test is the test of an individual component, or module, of the solution. The objective of a unit test is to verify that the component correctly implements the designed specifications.

Unit testing is based on knowledge of how the logical [code] unit is designed to work. The Team will utilize the test conditions developed for R 1.01. Unit testing will include tests for field ranges, values and lengths, functions, data validation, data dependencies, and any special processing contained in the module.

Modules will be developed and become available for testing independently of other modules. The developer responsible for the module will perform the unit test and identify and document the errors related to the independent operation of the program. There will be no SIRs generated by unit testing. Instead, the exit criteria will be the successful completion of all test conditions.

R 1.01 System Testing

System testing concentrates on requirements and business processes, and ensures that the system requirements have been met. System testing will test the components and interfaces working together as a whole, integrated solution.

The eZ-Audit Test Team will handle the creation and execution of test scripts. The test scripts for this phase will be developed using the requirements and Use Cases developed during the Design phase and the test conditions.

Ideally, system testing cannot begin until all modules have passed the unit test phase because the modules must be fully integrated in the test environment for system testing. However system testing may begin prior to the acceptance of all modules within the prior phase, but the impacted modules must be moved to the end of the test phase.

The Test Team will conduct system testing on the eZ-Audit application to ensure that all developed code modules work together to meet the intended business needs and requirements. System testing will be conducted in the test environment.

R 1.01 System test will consist of three test passes. The first test pass is a full test cycle. The second test pass is a full test cycle, to re-test failed items from first pass and confirm previous successes from first pass after failed items pass successfully. The third test pass (if needed) is a partial test cycle and regression test, to re-test failed items from the second pass. Anything more than 3 passes probably indicates a serious quality problem, and will be reported to the Project Team Lead. The Test Team will continue to add additional test passes until all test conditions are met successfully.

FSA Case users will participate in the second pass of R 1.01 System test. This will replace a formal UAT session.

R 1.01 System Testing will include the following scenarios:

1. Manage Users
2. Query
3. Create Submission
4. Code Findings
5. Assign Submissions
6. Select an Institution
7. Send to Co-Team Leader for Approval
8. Complete DDIF
9. Create Determination for FS
10. Resolve Exemption/Waiver Request
11. Quality Control
12. Unsolicited LOC
13. Upload and Accept Letter of Public Status

R 1.01 System Test Exit Criteria

At the end of the system test, relevant issues should be identified and documented. Level 1 SIRs will be identified as In-Progress prior to progressing to the user acceptance test. A determination on the number of open Level 2 and 3 SIRs will be at the discretion of the System Test manager. Level 4 SIRs are system enhancements and will not count towards the exit criteria for this phase.

Quantity	SIR Level
0	1 - High
3	2 - Medium
No defined limit	3 - Low
NA	4 - Enhancement

Table 1, System Test Exit Criteria

Regression Testing

The eZ-Audit Test Team will conduct end-to-end regression testing to ensure that the components of each system work together as intended after R 1.01 code changes have been made and tested at the system level. This will verify that previously released 1.0 code was not impacted by changes made for R 1.01. Regression testing will be conducted in the test environment. The test will focus on the following 6 scenarios:

1. Non-Profit Annual Submission and Resolution
2. Proprietary Annual Submission and Resolution
3. Public Annual Submission and Resolution
4. Merger/CIO Submission and Resolution
5. A-133 Exemption Submission and Resolution
6. Manage Users - Add/Manage Various User Types

eZ-Audit Regression Test Exit Criteria

At the end of the system test, relevant issues should be identified and documented. Level 1 SIRs will be identified as In-Progress prior to progressing to the regression test. A determination on the number of open Level 2 and 3 SIRs will be at the discretion of the System Test manager. Level 4 SIRs are system enhancements and will not count towards the exit criteria for this phase.

Quantity	SIR Level
0	1 - High
0	2 - Medium
3	3 - Low
N/A	4 - Enhancement

Table 2, Regression Test Exit Criteria

Test Conditions

Test conditions provide the ability to test the system functionality under specific circumstances to be performed in production. The test conditions will be developed so that they are repeatable and reusable. Test conditions must be repeatable so that errors can be reproduced and retested under the same circumstances.

Test conditions will be written using the detailed functional requirements from the Design phase. These conditions will be grouped into scenarios. The scenarios will contain a related set of conditions, with the condition mapped to the Use Cases that trace to the detailed requirements.

Finally, test scripts will be prepared that define the steps, input data, and expected result to test the scenarios. A single scenario may be split across multiple test scripts, or multiple scenarios may be included within a single test script.

Requirements Traceability

R 1.01 Requirements are traced both to R 1.01 Use Cases and to R 1.01 Test Conditions. All R 1.01 requirements are traced to a Use Case. Each requirement also maps to one or more test conditions. Test scripts are then developed based on the Test Conditions and Use Cases.

Baseline Test Data

Baseline data must be defined and loaded into the system prior to testing. This data includes:

- eZ-Audit Test Data – audit and financial data
- User Setup Data – user roles and responsibilities, audit finding codes, etc.

Test data will be created as representative sample data to use when executing the test scripts. Before test execution can begin, baseline test data will be created to enable the testers to execute the test scripts.

Test Scripts

Test Script Development

Test scripts are executed within each test cycle. The scripts test the basic system functionality derived by the system level requirements. Test scripts are documented in terms of a business function or technical requirement, based on the test conditions.

The process for developing system test scripts will be as follows:

1. Identify Test Scenarios from the Use Case documents
2. Identify Test Conditions
3. Write Test Scripts for test execution
4. Map Test Conditions and Scripts to Use Cases [which trace to Requirements]
5. Create baseline Test Data

Test Scenarios

Using the Use Case Specifications, test scenarios will be identified to demonstrate how users perform specific tasks in the eZ-Audit application. Use Case specifications describe functionality in terms of user behaviors to be satisfied by the system. Therefore, the test scenarios will describe the user interaction with the system required to perform a specific business function.

Test Script Standards

Each test script will contain the following information:

- Script Name & Number - assigned by script writer
- Description
- Created By
- Tested By - person who is performing the actions for a test condition
- Test Date - written on the printed script by the tester
- Prerequisites - reference data or transactions that must exist prior to script execution
- Use Case(s) Covered

For each step within the script, the following information is shown in table format:

- Step number - auto-assigned by the script template
- Action - detailed instructions for how to accomplish the step
- Expected Results - describes how the system should respond to the action
- Pass/Fail - describes whether the expected and actual results match for a condition. Pass means that the expected and actual results match. Fail indicates the expected and actual results did not pass for a condition.
- Comments - notes about the test condition
- SIR # - related SIR # for a defect logged against the failed test condition. If the condition fails for more than one reason, multiple SIRs will be logged.

System Investigation Requests (SIRs)

Whenever a test condition or test script step does not return the expected result, the tester will log the discrepancy between the expected and actual result as a System Investigation Request, or SIR. SIRs will be logged and tracked in a central repository, or SIR database, using Rational

ClearQuest. The SIR database will include a long and short description of the issue (including the actual and expected results), its severity and status, the affected test area or module, the test phase, test pass, and tester.

SIR Severity

The following table contains the SIR severities and definitions:

ID	Label	Description
1	High	The anomaly results in a failure of the complete software system, a subsystem, or a software module within the system. There is no way to make the failed component(s) work; however there is an acceptable processing alternative that will achieve the desired results (an acceptable work around exists).
2	Medium	The anomaly does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the anomaly impairs system usability.
3	Low	The anomaly does not cause a failure, does not impair usability, and the desired processing results are easily obtained by working around the anomaly. The anomaly may also be the result of non-conformance to a standard or related to the aesthetics of the system.
4	Enhancement	The anomaly is a request for an enhancement. Anomalies at this level may be deferred to a future release or will be treated as an addition to the defined project scope if the Modernization Partner and FSA agree upon its inclusion.

Table 3, SIR Severity Levels

SIR Status

The following table contains the SIR Status levels that will be used during this effort and a definition of each:

SIR Status	Definition
Opened	The SIR was created and is awaiting assignment or review
In-Progress	The SIR was assigned and that resource is currently working on its resolution
Resolved	Corrective action was completed and the SIR is awaiting validation
Closed	Corrective action has been completed and validated
Postponed	The CCB agreed to postpone the corrective action on this SIR to a later release
Duplicate	The issue identified in this SIR is identical to another open SIR, and the resolution to the duplicate SIR will be tracked through the original open SIR

Table 4, SIR Statuses

Re-test SIR Fixes

As SIR fixes are completed, the script in which the SIR was encountered will be re-tested. These SIRs may have originated from an earlier pass, the same pass, or from another script that has one or more components in common. The Test Team will coordinate with the Development Team to group related issues together, thus minimizing the delay of executing the next pass of the test script.

SIR fixes will be re-tested using local copies of the original input files, data, etc. After conducting a re-test, SIRs that fail are reopened and newly identified SIRs are logged in the SIR database. Additional passes will continue to be conducted, as appropriate, until the exit criteria are achieved for each phase.

Change Control Process

During the solution lifecycle process, the controlled baseline requirements may require updates and the Change Control Process is designed to manage these modifications. Changes may be required for a variety of reasons, including the addition of new technology or functionality, the resolution of issues, and in response to technical and operational tests and evaluations.

Identified changes will be logged as System Investigation Requests (SIRs) and discussed by the Change Control Board (CCB). Changes that are related to existing requirements will be assigned for remediation. Changes that result in a new requirement will be noted as an enhancement and will either be postponed to a future release or will be negotiated with the FSA Project Sponsor for inclusion into the current release.

Detailed Test Plan

System Test Execution

System Test Environment

A separate test instance will be set up for the test team. All successfully unit-tested code will be migrated to the test environment prior to test execution. Two test instances will be created. Having a second test instance will allow the test team to create a new baseline each time a test pass has been completed.

The first test instance will be the current testing instance used to execute each System Test Pass. The second instance will be used solely for inputting baseline test data. At the beginning of each test pass, the Test Team will coordinate with the Development Team to refresh both instances and apply the baseline data to each instance.

Test Execution

The Test team will execute the scripts according to the step-by-step directions in each script. Scripts that pass will be signed off and stored in a binder.

Test Passes

A test pass is defined as one full test execution cycle. Development remediation activities are conducted between passes within the same test phase. Additional passes will be run within a test phase until the test meets the defined exit criteria.

Test passes will be performed in steps. A pass contains the following steps:

1. The tester executes the test scripts
2. The tester enters a SIR when results from the test deviate from what is expected
3. The Test Team notifies the Development Team when the test pass is complete
4. The Development Team runs the SIR report at the end of the pass
5. The Development Team performs an impact analysis on the SIR list
6. SIR fixes are completed and moved into the testing environment according to Configuration Management guidelines
7. The Development Team updates the SIR database with the SIR status information
8. The Development Team notifies the Test Team that testing may resume
9. The Test Team begins a new pass and continues this process until the exit criteria have been achieved for the phase

If a critical error occurs within a test pass that prevents the entire set of scripts from being executed and the fix requires lengthy remediation, the pass will be considered complete and the SIR resolution phase will begin. Once the SIR fixes are migrated into the test environment, a new pass will begin. The beginning of a new pass will be at the discretion of the Test Team, and a new pass may begin with unresolved SIRs outstanding. Test passes will continue until the scripts for that pass have been executed and the exit criteria have been satisfied. The amount of time necessary to complete a test pass will vary depending on the particular area of testing being performed.

Identification and Resolution of Issues

The following process will be followed for identifying and resolving problems encountered during testing:

1. The Tester discovers a discrepancy with a script and works with another tester to determine if the problem is with the software, the test script, or some other source. This will serve as a form of triage to identify true issues, and to not bother the programmers with issues not related to the code.
2. If the Test Team determines that the problem is with the software, the tester will log the issue in the SIR database with supporting detail including, screen prints or sections of the requirements document. The tester will document the problem by entering "Fail" in the Pass/Fail column of the script, at the step where the script failed. The tester should also capture the date and time the failure was noted.
3. The SIR database should be updated to reflect the script failure. In addition to tracking software failures, the team will also track failures related to undefined requirements, faulty scripts, Oracle errors, and "other."
4. The Test Team will notify the Development Team of any failed test scripts in order to assign the problem to a developer to correct it.
5. The developer should work with the tester in the development environment to ensure that the software change fixed the problem.

6. The Implementation Team should notify the Test Team when the changed software has been migrated to the test environment. This should be done following the FSA-defined process.
7. The tester will confirm, in the test environment, that the software change fixed the problem.
8. The tester will note, in the Pass/Fail column of the script, the date and time that the step of the script was successfully executed, and continue with the script.
9. During this process, if a problem is identified in testing, the System Test and Implementation Teams will make a determination on the need for and scope of regression testing.

The key to quickly resolving software defects is direct communication between the tester and the developer who is fixing the problem.

Testing Schedule

The test schedule is designed to meet the Release 1.01 go-live date of August 4th, 2003.

Detailed Test Schedule

The following table depicts the activities and events to be conducted for R 1.01 Testing.

Test Phase	Planned Dates
Develop Test Conditions	6/06 - 6/17
Document Test Data Needs	6/26
Develop R 1.01 System Test Scripts	6/13 - 7/03
Develop eZ-Audit Regression Test Scripts	7/7 - 7/10
Execute System Test:	7/14 - 7/24
Pass 1	7/14 - 7/15
Fix Cycle	7/16 - 7/17
Pass 2	7/18 - 7/21
Fix Cycle	7/22
Pass 3	7/23 - 7/24
Fix Cycle	7/25
Execute eZ-Audit Regression Test	7/28 - 7/30
Final Code Fixes	7/30 - 7/31
Code Complete	7/31
R 1.01 Code Available in Production	8/3

Table 5, Test Schedule