User Acceptance Test Plan Guide

Guide S 36

Working Draft, Version 1.0

Ministry of Forests

Information Management Group

Table of Contents

TABLE OF CONTENTS 2

1. INTRODUCTION 3

1.1 AUDIENCE 3
1.2 PURPOSE 3
1.3 OTHER STANDARDS 3

2. TEST PLAN DEVELOPMENT 4

2.1 SOURCE MATERIAL 4
2.2 TEST DATA 4
2.3 TEST PLAN SCHEDULES 5
2.4 TEST FACILITIES 5
2.5 TEST SCRIPTS 5
2.6 TEST PLAN OUTPUT 7
2.7 TEST SCRIPT REUSE 7

3. CONCLUSION 8

1. Introduction

This section outlines the audience and purpose of this standards document.

1.1 Audience

This document is directed at those people who are responsible for producing user acceptance test plans for application systems for the British Columbia Ministry of Forests.

1.2 Purpose

The purpose of this document is to provide guidance in the development of user acceptance test plans. These plans should exercise every part of the application to ensure that the business needs of the ministry are met. The plan should not include quality assurance for adherence to Ministry standards. QA is a separate function, which should be addressed before User Acceptance testing starts.

It is not practical to provide a 'universal template' for testing, as the applications delivered to the ministry vary widely in function, application architecture, and the hardware on which they run. This document will provide a list of the kinds of tests that should be included in a test plan to ensure that the delivered application meets the business needs of the users.

1.3 Other Standards

Other related standards can be found on the Information Management Group standards web page http://extranet.for.gov.bc.ca/SysDevGuide/S_Guides

2. Test Plan Development

2.1 Source Material

User Acceptance Test Plans should be based on the Detailed Requirements Document. Test scripts should be developed to cover each requirement. Do not use developer's test plans as the basis for user acceptance testing, this would simply re-run tests the developer should have already carried out, and it reflects the developer's understanding of the requirements, which may be incorrect. Detailed Design documents, process models, system prototypes, etc., are valuable sources for developing the scripts to match the process flow of the application.

Where possible, base test scripts on real source material, such as documents or forms which will be the inputs for the production system. Care must be taken to ensure the scripts completely exercise the function being tested. Incomplete source documents may need to be enhanced with additional information to ensure that all functions are tested. For example, if a screen displays the address of a client, the client number and location may have to be changed to one that requires the maximum number of lines in an address, to ensure that all the address fields are correctly displayed. FOI and security considerations should also be taken into account when using production source material. Client names or other details may have to altered prior to distributing the input material to testers. Business keys may have to the changed to match available test data.

2.2 Test Data

Test data should be prepared in conjunction with the test scripts. Consider the following when determining test data:

- Can live production data be replicated to the test environment?
- Does current production data reflect the needs of the new or updated application?
- Are there any security or FOI considerations in using production data?
- Is there sufficient disk space available in the test environment for replication of production tables, or will a subset of data have to be created?
- Are all the required validation tables created and populated in the test environment?
- If a subset of test data is created, are related tables such as client, opening, etc., populated with sufficient data for the test scripts to be completed?
- Will the test data have to be refreshed for multiple test cycles, for to fully test batch update processes?

Database administration staff should be consulted regarding creation, replication and refreshing test data

2.3 Test Plan Schedules

Scheduling of user acceptance testing should be done in consultation with the application developer, and IMG technical staff to ensure:

- Training and support is available for testers;
- Test equipment (application servers, PCs, printers etc.) and test databases are available;
- There are minimal conflicts with other applications which may be using common test data;
- The results of user acceptance testing are communicated clearly to the developer.

Generally, a minimum of two rounds of user acceptance testing is required, with more for large or complex applications. Sufficient time must be given between the test rounds to collate the test results, communicate them to the developer, fix and test the application, and migrate the new application code to the test environment.

The number of users required for testing, the length of each round of testing, and the number test cycles should be estimated in advance. Try to secure a commitment from users to complete the full test cycle, to cut down on time spent training new users.

2.4 Test Facilities

Wherever possible, testing should be done at a dedicated facility, away from testers normal work environment. This allows testers to perform the scripts uninterrupted by normal work, and provides an environment where training and support can be provided. Ensure that:

- Any boardrooms, etc., that are to be used for training have sufficient network access to support the number of testers involved;
- Any PCs, projectors, or other equipment required has been booked;
- Systems support people are available to move and configure PCs, etc.

2.5 Test Scripts

Test scripts should be developed to fully exercise each business function of the application. The script should be designed to test that all valid data required by the application can be entered and saved, and to test that invalid data is rejected. Data captured in tests to ensure that all valid data is accepted and retained can form the basis for further tests of view, update, delete or reporting functions.

Tests of large applications can be broken down so that individual testers or groups test specific functions, each with a unique test script. The tests can be run in parallel to reduce the elapsed time required for each round of testing.

Scripts should be sufficiently detailed that new users can execute them with minimal training. A listing of valid data should be provided so the tester can proceed without delays caused by validation errors. This can be done by providing customized scripts for each tester, or by providing test data in an appendix to the test plan. The test data should include such things as clients, tenures, opening numbers, etc., which are valid for the test database being used. Any codes that are not supported by list boxes or prompt functions should also be provided.

Scripts should include tests for the following:

- Page header and default data is complete and provides useful information.
- Information is presented in a logical way that is consistent with the users' business flow.
- The cursors tab between fields in a logical sequence.
- List boxes display all code options clearly, and function correctly.
- Field lengths are long enough to display the maximum valid data values.
- Date edits make business sense. E.g. completed_date can't be in the future.
- Edits function correctly, allowing all valid data and preventing invalid data for being entered. Include tests for cross-field edits (e.g. a status of 'completed' cannot be entered if a 'completed_date' field is blank).
- Data that is saved to the database is displayed as entered – no truncation, etc.
- Special Characters are handled correctly, and don't cause printing or processing errors. Single and double quotes, asterisks, pound signs etc., should be tested.
- Sequence numbers are generated properly, and any 'child' records are properly related back the parent record.
- Deletions are handled properly and don't create orphan records. Any rules preventing deletions are enforced.
- Prompt (including mouse-over messages) and Error messages are clear and understandable to users.
- Help text is available and provides useful information to the users.
- All Navigation options are available and function correctly.
- Screen layout and navigation through use of buttons or other controls is consistent throughout the application.
- Fonts and form sizes display clearly at normal default display settings.
- Any function keys or other special keys or controls function correctly.
- Security groups function as required. Testers should be added to each available security group, and should test all functions in the application, to ensure that groups with restricted privileges cannot use functions from which they should be restricted.
- Reports have appropriate selection criteria, and the report output matches the selection criteria used.
- Reports meet ministry standards for headings and display format.
- Batch jobs or stored procedures execute as expected and produce the desired results.

2.6 Test Plan Output

Test plans scripts should provide space for testers to make notes, and to check off the portion of tests completed. A standard error report form should be included, preferably as an electronic form. The error report should include fields for:

- Tester's name.
- Test date.
- Id of the test data (business keys used, etc., so the test is reproducible).
- Function being performed.
- Problem type (system error, business function not met, suggested enhancement).
- Expected result.
- Actual result.
- Description of the problem. Include Screen shots any error messages received if possible.

The test script should include instructions for problem reports to be saved into a specific directory so that the reports can be more easily collated.

2.7 Test Script Reuse

The test scripts created for a new system should be re-cycled throughout the application's life. Editing existing test scripts can save time and money, and maintain quality, as key requirement information will be re-used.

3. Conclusion

This document provides the guidelines that will facilitate production of user acceptance test plans .

It is intended to provide a checklist of the most common items that should be considered when designing a user acceptance test plan.