|  |  |  |
| --- | --- | --- |
| Document ID  **ITSW107** | Title  **SOFTWARE TESTING** | Print Date  **mm/dd/yyyy** |
| Revision  **0.0** | Prepared By  **Preparer’s Name / Title** | Date Prepared  **mm/dd/yyyy** |
| Effective Date  **mm/dd/yyyy** | Reviewed By  **Reviewer’s Name / Title** | Date Reviewed  **mm/dd/yyyy** |
|  | Approved By  **Final Approver’s Name / Title** | Date Approved  **mm/dd/yyyy** |

**Policy:** All software products developed by the Company shall pass acceptance tests, beta tests, and software release tests before they are released to the customer or user community.

**Purpose:** To ensure that Company-developed software is error-free and is capable of consistently performing the tasks for which it was designed.

**Scope:** All software products and updates released by the company.

**Responsibilities:**

The Beta Test Coordinator is responsible for implementing, coordinating, and managing the beta testing with the beta test users.

The Quality Assurance Manager is responsible for administering and coordinating the tests indicated in accordance with the procedure found in the acceptance test plan.

The Software Designer is responsible for reviewing test problem reports and assigning problem reports to software programmers for resolution.

The Systems Analyst is responsible for creating test plans for completed software products.

**Procedure:**

### 1.0 SOFTWARE TESTING OVERVIEW

* 1. During software development, the Systems Analyst creates plans for testing the completed software product. For information on creating test plans, see ITSW103 SYSTEMS ANALYSIS.
  2. After programming has been completed, the Quality Assurance Manager administers and coordinates the tests indicated in the test plan.
  3. Each test conducted should include a test script or scripts with detailed instructions on what exactly to test, using ITSW107-1 PROJECT TEST SCRIPT as a guide. The tester should follow each procedure and note any irregularities discovered on the form.
  4. ITSW107-2 PROJECT TEST CHECKLIST should be used to highlight specific test criteria.
  5. The Quality Assurance Manager should maintain a central database of test problem reports, using ITSW107-3 PROJECT TEST PROBLEM REPORT as a guide.

### 2.0 SOFTWARE ACCEPTANCE TESTING

2.1 The Quality Assurance Manager tests the software in accordance with the procedure found in the acceptance test plan, in accordance with ITSW103 SYSTEMS ANALYSIS. The test must be completed error-free before the software is released to beta test.

2.2 The Quality Assurance Manager completes a problem report for each error encountered. If a problem prevents one or more components from being thoroughly tested, the Quality Assurance Manager:

* Discontinues testing the affected components.
* Immediately submits the problem report to the Systems Analyst for resolution.

2.3 The Systems Analyst and Software Designer review the test problem reports. They separate the reports into two groups, one for programming errors, and the other for design errors.

* To resolve a design error, the Project Manager, Systems Analyst and Software Designer perform the steps outlined in ITSW108 DESIGN CHANGES DURING DEVELOPMENT.
* To resolve a programming error, the Software Designer assigns the problem report to the responsible programmer. The programmer corrects the error and returns the updated software components to the Quality Assurance Manager for retesting.

2.4 After the acceptance test is complete, the Quality Assurance Manager releases the software to beta test.

### 3.0 SOFTWARE BETA TESTING

* 1. The Quality Assurance Manager appoints a Beta Test Coordinator responsible for implementing the beta test, in accordance with ITSW103 SYSTEMS ANALYSIS.
  2. The Beta Test Coordinator contacts and schedules the beta test sites as early in the software development project as possible. The coordinator identifies the key contact at each site and promptly informs them of any changes in the project schedule.
  3. The Beta Test Coordinator schedules training for the beta testers. The coordinator schedules this training as close as possible to the date on which the beta release will be installed. Properly coordinating training helps ensure that the information will be fresh in the minds of the users during the test.
  4. The Beta Test Coordinator ensures that the testers get any additional help they might need to get ready to use the software. If necessary, an internal software consultant helps the testers install and implement the software.
  5. To ensure that the testers continue to move forward with their testing, the Beta Test Coordinator contacts them weekly. The coordinator must be prepared to answer the testers’ questions and address any problems they are having.
  6. As the testers identify problems, The Beta Test Coordinator completes an ITSW107-3 PROJECT TEST PROBLEM REPORT for each error encountered.
  7. The Systems Analyst and Software Designer review the beta test problem reports. They separate the reports into two groups, one for programming errors, and the other for design errors.
* To resolve a design error, the Project Manager, Systems Analyst, and Software Designer perform the steps outlined in ITSW108 DESIGN CHANGES DURING DEVELOPMENT.
* To resolve a programming error, the Software Designer assigns the problem report to the responsible programmer. Programmers should respond immediately to problems that prevent testers from using the system. Less severe problems can be fixed periodically, such as biweekly or monthly.
* The programmer corrects the error, completes the resolution section of ITSW107-3 PROJECT TEST PROBLEM REPORT and returns the updated software components to the Beta Test Coordinator for retesting.
  1. The Beta Test Coordinator sends updates to all beta sites. Beta testing continues as defined in the test plan until the Project Manager, Systems Analyst, and Quality Assurance Manager determine that sufficient beta testing has occurred.
  2. After the beta test is complete, the Quality Assurance Manager releases the software for final release testing.

## 4.0 SOFTWARE FINAL RELEASE TESTING

* 1. When the software is fully tested and ready for release to sales and shipping, the Quality Assurance Manager tests to ensure that the software installation process functions without error.
  2. If the software is an update to an existing software product, the quality assurance analyst must test the installation process for all versions of the software that the company supports.
  3. After the final release test is complete, the Quality Assurance Manager releases the software for distribution, in accordance with ITSW109 SOFTWARE RELEASES AND UPDATES procedure.

**Forms:**

* ITSW107-1 SOFTWARE PROJECT TEST SCRIPT
* ITSW107-2 SOFTWARE PROJECT TEST CHECKLIST
* ITSW107-3 SOFTWARE PROJECT TEST PROBLEM REPORT

**References:**

1. **IEEE STANDARD 829-2008 – STANDARD FOR SOFTWARE AND SYSTEM TEST DOCUMENTATION**

This is the IEEE standard for documenting software testing, which specifies eight stages in the documentation process, each stage producing its own separate document:

* **Test Plan**: A detail of how the test will proceed, who will do the testing, what will be tested, in how much time the test will take place, and to what quality level the test will be performed.
* **Test Design Specification**: A detail of the test conditions and the expected outcome. This document also includes details of how a successful test will be recognized.
* **Test Case Specification**: A detail of the specific data that is necessary to run tests based on the conditions identified in the previous stage.
* **Test Procedure Specification**: A detail of how the tester will physically run the test, the physical set-up required, and the procedure steps that need to be followed.
* **Test Item Transmittal Report**: A detail of when specific tested items have been passed from one stage of testing to another.
* **Test Log**: A detail of what tests cases were run, who ran the tests, in what order they were run, and whether or not individual tests were passed or failed.
* **Test Incident Report**: A detail of the actual versus expected results of a test, when a test has failed, and anything indicating why the test failed.
* **Test Summary Report**: A detail of all the important information to come out of the testing procedure, including an assessment of how well the testing was performed, an assessment of the quality of the system, any incidents that occurred, and a record of what testing was done and how long it took to be used in future test planning. This final document is used to determine if the software being tested is viable enough to proceed to the next stage of development.

1. **ISO/IEC 12207:2008 – SYSTEMS AND SOFTWARE ENGINEERING – SOFTWARE LIFE CYCLE PROCESSES**

For detailed information, see <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4578383>.

1. **IEEE 12207-2008 – SYSTEMS AND SOFTWARE ENGINEERING – SOFTWARE LIFE CYCLE PROCESSES**

This ISO standard describes the major component processes of a complete software life cycle and the high-level relations that govern their interaction. It establishes a software life cycle architecture based on two principles, modularity of processes and responsibility for processes. There are three process classes in the ISO software life cycle: primary (such as acquisition and operations); supporting (such as documentation and configuration management); and organizational (such as infrastructure and training). Each life cycle process is made up of activities, and each activity is further subdivided into tasks. The standard is based on ISO quality management principles.

The IEEE version of 12207 is more closely aligned with the ISO standard than it was in previous versions.

For more information, visit the ISO web site at <http://www.iso.org/iso/catalogue_detail.htm?csnumber=43447> or the IEEE web site at <http://standards.ieee.org/findstds/standard/12207-2008.html>.

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Description of Changes** | **Requested By** |
| 0 | mm/dd/yyyy | Initial Release |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[This page intentionally left blank]

**ITSW107-1 SOFTWARE PROJECT TEST SCRIPT**

Test Case No. \_\_\_\_\_\_\_\_\_\_\_\_ Page \_\_\_ of \_\_\_

|  |  |
| --- | --- |
| **PROCEDURES** | **NOTES** |
| LOGIN:   * + 1. At Login Prompt, enter user name “Test” and press the “Enter” key     2. At Password Prompt, enter user password “Test1” and press the “Enter” key | [ Sample text provided as an example] |

[This page intentionally left blank]

**ITSW107-2 SOFTWARE PROJECT TEST CHECKLIST**

Test Case No. \_\_\_\_\_\_\_\_\_\_\_\_ Checklist No. \_\_\_\_\_\_\_\_\_\_\_\_ Page \_\_\_ of \_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **ITEM** | **YES** | **NO** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[This page intentionally left blank]

**ITSW107-3 SOFTWARE PROJECT TEST PROBLEM REPORT**

Test Case No. \_\_\_\_\_\_\_\_\_\_\_\_ Test Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Test Phase (circle one): Acceptance Beta Software Version: \_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **PROBLEM DESCRIPTION** | **PROBLEM RESOLUTION** |
|  |  |

**Problem Reported By**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Date**: \_\_\_\_\_\_\_\_\_\_

**Problem Status** (circle one): Open Closed On-Hold

**Problem Reviewed By** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Date**: \_\_\_\_\_\_\_\_\_\_

[This page intentionally left blank]