iBEX

Test and Risk Analysis Plan

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2017 | 0.1 | Test records are created | Altay Brusan |
| 2018 | 0.1 | Risk analysis is added | Altay Brusan |
| 2018 | 1.0 | Test and Risks are summed under one doc | Altay Brusan |
| 2019 | 1.0 | Test and Risks are updated | Altay Brusan |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Intended Audience 3

1.4 Document Terminology and Acronyms 3

1.5 References 3

1.6 Document Structure 3

2. Evaluation Mission and Test Motivation 3

2.1 Background 3

2.2 Evaluation Mission 3

2.3 Test Motivators 3

3. Target Test Items 3

4. Outline of Planned Tests 3

4.1 Outline of Test Inclusions 3

4.2 Outline of other candidates for potential inclusion 3

4.3 Outline of Test Exclusions 3

5. Test Approach 3

5.1 Initial Test-Idea Catalogs and other reference sources 3

5.2 Testing Techniques and Types 3

5.2.1 Data and Database Integrity Testing 3

5.2.2 Function Testing 3

5.2.3 Business Cycle Testing 3

5.2.4 User Interface Testing 3

5.2.5 Performance Profiling 3

5.2.6 Load Testing 3

5.2.7 Stress Testing 3

5.2.8 Volume Testing 3

5.2.9 Security and Access Control Testing 3

5.2.10 Failover and Recovery Testing 3

5.2.11 Configuration Testing 3

5.2.12 Installation Testing 3

6. Entry and Exit Criteria 3

6.1 Test Plan 3

6.1.1 Test Plan Entry Criteria 3

6.1.2 Test Plan Exit Criteria 3

6.1.3 Suspension and resumption criteria 3

6.2 Test Cycles 3

6.2.1 Test Cycle Entry Criteria 3

6.2.2 Test Cycle Exit Criteria 3

6.2.3 Test Cycle abnormal termination 3

7. Deliverables 3

7.1 Test Evaluation Summaries 3

7.2 Reporting on Test Coverage 3

7.3 Perceived Quality Reports 3

7.4 Incident Logs and Change Requests 3

7.5 Smoke Test Suite and supporting Test Scripts 3

7.6 Additional work products 3

7.6.1 Detailed Test Results 3

7.6.2 Additional automated functional Test Scripts 3

7.6.3 Test Guidelines 3

7.6.4 Traceability Matrices 3

8. Testing Workflow 3

9. Environmental Needs 3

9.1 Base System Hardware 3

9.2 Base Software Elements in the Test Environment 3

9.3 Productivity and Support Tools 3

9.4 Test Environment Configurations 3

10. Responsibilities, Staffing and Training Needs 3

10.1 People and Roles 3

10.2 Staffing and Training Needs 3

11. Iteration Milestones 3

12. Risks, Dependencies, Assumptions and Constraints 3

13. Management Process and Procedures 3

13.1 Measuring and Assessing the Extent of Testing 3

13.2 Assessing the deliverables of this Test Plan 3

13.3 Problem Reporting, Escalation and Issue Resolution 3

13.4 Managing Test Cycles 3

13.5 Traceability Strategies 3

13.6 Approval and Signoff 3

Test and Risk Analysis Plan

# Introduction

## Purpose

The purpose of the Iteration Test Plan is to gather all the information necessary to plan and control the test effort and risks. It describes the approach to testing and analyzing risks of the software, and is the top-level plan generated and used by managers to direct the test effort and risks.

This *Test and risk Plan* for the iBEX supports the following objectives:

• Identifies the items that should be targeted by the tests.

• Identifies the motivation for and ideas behind the test areas to be covered.

• Outlines the testing approach that will be used.

• Identifies the required resources and provides an estimate of the test efforts.

• Lists the deliverable elements of the test project.

## Scope

This document engulfs both system and unit test. The criteria for evaluation are functionality, Reliability, Performance and usability. Risks are analyzed and added to this document to make the comparison easier. This document does not contain the details and the tools used for test and risk implementation. Hardware and device tests are out of the boundaries of this document.

## Intended Audience

This document is prepared for open source community members that want to got involved in iBEX project.

## Document Terminology and Acronyms

**keyword**: one or more words used as a reference to a specific set of actions intended to be performed during the execution

**high-level keyword:** keyword that covers complex activities that may be composed from other keywords and is used by domain experts to assemble keyword test cases.

**keyword-driven Testing:** testing using test cases composed from keywords.

**Keyword-Driven Testing framework**: test framework covering the functional capabilities of a keyword-driven editor, decomposer, data sequencer, manual test assistant, tool bridge, data and script repositories, a keyword library and the test execution environment

## References

[1] ISO/IEC/IEEE (2015) BSI Standards Publication Software and systems engineering — Software testing Part 4 : Test techniques.

## Document Structure

At the first section the test analysis is added and then risks are analyzed.

# Evaluation Mission and Test Motivation

The aim of this document is to formalize the risk and testing efforts.

## Background

iBEX is required to provide evidence of compliance with medical standards and well-established procedures for medical grade applications. This document formalizes this effort. In order to make a broad view, both tests and risks are brought together in one document.

## Evaluation Mission

Provide a brief statement that defines the mission for the evaluation effort in the current iteration. This statement might incorporate one or more concerns including:

* find as many bugs as possible
* find important problems, assess perceived quality risks
* advise about perceived project risks
* certify to a standard
* verify a specification (requirements, design or claims)
* advise about product quality, satisfy stakeholders
* advise about testing
* fulfill process mandates
* and so forth

## Test Motivators

The main motive is to reduce the bugs and eliminate the risks at the early stage of the project.

# Target Test Items

The listing below identifies those test items⎯software, hardware, and supporting product elements ⎯that have been identified as targets for testing. This list represents what items will be tested.

|  |  |
| --- | --- |
| Workflow | Description |
| register new patient | capture the patient’s demographic information and record it in the local database. |
| view image | load local image file (with tiff or dcm extension) and display it on the screen |
| select imaging task | opens the local task list to select a task. |
| quick scan | starts to acquire a study without registering the patient. It is useful for emergency scans |
| configure devices | updates parameters of components (such as detector binning mode or energization level of power source). |
| apply image filtering | updates the images by applying a filter plug-in (one could also make changes on the filter parameters before its application). |
| manage local database | checks the database for data integrity and controls the capacity. |
| decide on image | after acquiring an image, the user can decide to accept or reject the image. If the image was rejected, then the acquisition is expected to be repeated. |
| log event | Events within the system are recorded in different levels (e.g. warning, information, error, fail). |
| send to PACS | the finished studies are stored in a remote PACS server. The server connection settings should be configurable. |
| configure machine | Peripheral components’ calibration parameters are updated. |
| make DICOM file | converts the raw image into a DICOM file. |
| create account | User credentials are updated. |
| update worklist | Modality worklist is automatically (and manually) updated. |

# Outline of Planned Tests

This section provides a high-level outline of the testing that will be performed. The outline in this section represents a high-level overview of both the tests that will be performed and those that will not.

## Outline of Test Inclusions

* RIS integration.
* PACS integration.
* DCMTK transaction.
* DICOMIZATION.
* Image presentation.
* Plugin integration.
* Log evaluation
* Database transaction.

## Outline of Other Candidates for Potential Inclusion

* Integration with RIS systems provided from different vendors.
* Integration with PACS server provided by different vendors
* Web based PACS server integration

## Outline of Test Exclusions

X-ray scanner device performance test. These tests are device vendor responsibility and out of iBEX boundaries.

# Test Approach

iBEX core is planned to be tested by automatic test tools.

User Interfaces are planned to be tested in manual tools. This include CTK, VTK and Qt user interface widgets.

The DICOM and PACS connection are planned to be semi-automated. The creation side is planned to be automatic, but the outputs evaluation is required to be manual.

## Initial Test-Idea Catalogs and Other Reference Sources

Requirement document.

## Testing Techniques and Types

### Data and Database Integrity Testing

|  |  |
| --- | --- |
| Technique Objective: | Exercise database access methods and processes independent of the UI so you can observe and log incorrect functioning target behavior or data corruption. |
| Technique: | • Invoke each database access method and process, seeding each with valid and invalid data or requests for data.  • Inspect the database to ensure the data has been populated as intended and all database events have occurred properly or review the returned data to ensure that the correct data was retrieved for the correct reasons. |
| Oracles: | Human  The correctness of the system under test is determined by manual analysis |
| Required Tools: | The technique requires the following tools:   * base configuration imager and restorer * database SQLite utilities and tools |
| Success Criteria: | Data consistently is deposited with in the database.  Database integrity. |
| Special Considerations: | * Processes should be invoked manually. * Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events. |

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | Exercise target-of-test functionality, including navigation, data entry, processing, and retrieval to observe and log target behavior. |
| Technique: | Execute each use-case scenario’s individual use-case flows or functions and features, using valid and invalid data, to verify that:  • the expected results occur when valid data is used  • the appropriate error or warning messages are displayed when invalid data is used  • each business rule is properly applied |
| Oracles: | Human  consistency oracle |
| Required Tools: | The technique requires the following tools:   * Test Script Automation Tool * base configuration imager and restorer * backup and recovery tools * installation-monitoring tools (registry, hard disk, CPU, memory, and so forth) |
| Success Criteria: | The technique supports the testing of:  • all key use-case scenarios  • all key features |
| Special Considerations: | Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test. |

### User Interface Testing

|  |  |
| --- | --- |
| Technique Objective: | Exercise the following to observe and log standards conformance and target behavior:   * Navigation through the target-of-test reflecting business functions and requirements, including window-to-window, field-to- field, and use of access methods (tab keys, mouse movements, accelerator keys). * Window objects and characteristics can be exercised–such as menus, size, position, state, and focus. |
| Technique: | Create or modify tests for each window to verify proper navigation and object states for each application window and object. |
| Oracles: | a human oracle |
| Required Tools: | Different size screens. |
| Success Criteria: | The text format and order does not crash on different size screens |
| Special Considerations: | Not all properties for custom and third-party objects can be accessed. |

### Performance Profiling

|  |  |
| --- | --- |
| Technique Objective: | To measure the iBEX integrability with different kind of device plugins and filtering plugins |
| Technique: | • Develop different filtering and device plugins |
| Oracles: | * Human * Device API |
| Required Tools: | * Test Script Automation Tool * X-Ray device component and API * installation-monitoring tools 3 |
| Success Criteria: | The technique supports testing:  • plugin not failed for both valid and invalid inputs |
| Special Considerations: | When designing the plugins, they should be unit tested already and include the test results beside the source. No fail due to the plugin is acceptable. |

### Load Testing

|  |  |
| --- | --- |
| Technique Objective: | iBEX is immersed by the imaging tasks |
| Technique: | • A script automatically create load for iBEX. This load include transaction with database and plugins. |
| Oracles: | Script |
| Required Tools: | * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring * Data-generation tools |
| Success Criteria: | System does not crash for a full day operation. |
| Special Considerations: | • Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.  • The databases used for load testing should be either actual size or scaled equally. |

### Stress Testing

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test functions under the following stress conditions to observe and log target behavior that identifies and documents the conditions under which the system **fails** to continue functioning properly  • little or no memory available on the server (RAM and persistent storage space)  • maximum actual or physically capable number of clients connected or simulated  • multiple users performing the same transactions against the same data or accounts  • “overload” transaction volume or mix (see Performance Profiling above)] |
| Technique: | • [Use tests developed for Performance Profiling or Load Testing.  • To test limited resources, tests should be run on a single machine, and RAM and persistent storage space on the server should be reduced or limited.  • For remaining stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix. |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | The technique supports the testing of Stress Emulation. The system can be emulated successfully in one or more conditions defined as stress conditions and an observation of the resulting system state during and after the condition has been emulated can be captured.] |
| Special Considerations: | • [Stressing the network may require network tools to load the network with messages or packets.  • The persistent storage used for the system should temporarily be reduced to restrict the available space for the database to grow.  • Synchronize the simultaneous clients accessing of the same records or data accounts.] |

### Volume Testing

[Volume testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. Volume testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period. For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database, and would check that the software behaved normally and produced the correct report.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test under the following high volume scenarios to observe and log target behavior:  • Maximum (actual or physically-capable) number of clients connected, or simulated, all performing the same, worst case (performance) business function for an extended period.  • Maximum database size has been reached (actual or scaled) and multiple queries or report transactions are executed simultaneously.] |
| Technique: | • [Use tests developed for Performance Profiling or Load Testing.  • Multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix (see Stress Testing) for an extended period.  • Maximum database size is created (actual, scaled, or filled with representative data) and multiple clients are used to run queries and report transactions simultaneously for extended periods.] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | [The technique supports the testing of Volume Emulation. Large quantities of users, data, transactions, or other aspects of the system use under volume can be successfully emulated and an observation of the system state changes over the duration of the volume test can be captured.] |
| Special Considerations: | [What period of time would be considered an acceptable time for high volume conditions, as noted above?] |

### Security and Access Control Testing

[Security and Access Control Testing focuses on two key areas of security:

• Application-level security, including access to the Data or Business Functions

• System-level Security, including logging into or remotely accessing to the system.

Based on the security you want, application-level security ensures that actors are restricted to specific functions or use cases, or they are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that “user type one” can see all customer information, including financial data, however, “user two” only sees the demographic data for the same client.

System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.]

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test under the following conditions to observe and log target behavior:   * Application-level Security: an actor can access only those functions or data for which their user type is provided permissions. * System-level Security: only those actors with access to the system and applications are permitted to access them. |
| Technique: | * [Application-level Security: Identify and list each user type and the functions or data each type has permissions for.]   + Create tests for each user type and verify each permission by creating transactions specific to each user type.   + Modify user type and re-run tests for same users. In each case, verify those additional functions or data are correctly available or denied. * System-level Access: [See Special Considerations below] |
| Oracles: | [Outline one or more strategies that can be used by the technique to accurately observe the outcomes of the test. The oracle combines elements of both the method by which the observation can be made and the characteristics of specific outcome that indicate probable success or failure. Ideally, oracles will be self-verifying, allowing automated tests to make an initial assessment of test pass or failure, however, be careful to mitigate the risks inherent in automated results determination.] |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * “Hacker” security breach and probing tools * OS Security Admin Tools] |
| Success Criteria: | [The technique supports the testing of for each known actor type the appropriate functions or data affected by security settings can be tested.] |
| Special Considerations: | [Access to the system must be reviewed or discussed with the appropriate network or systems administrator. This testing may not be required as it may be a function of network or systems administration.] |

### Failover and Recovery Testing

|  |  |
| --- | --- |
| Technique Objective: | Simulate the failure conditions and exercise the recovery processes (manual and automated) to restore the database, applications, and system to a desired, known, state. The following types of conditions are included in the testing to observe and log target behavior after recovery:  • power interruption to the client  • power interruption to the server  • communication interruption via network servers  • interruption, communication, or power loss to DASD (Dynamic Access Storage Devices) and DASD controllers  • incomplete cycles (data filter processes interrupted, data synchronization processes interrupted)  • invalid database pointers or keys  • invalid or corrupted data elements in database |
| Technique: | The tests already created for Function and Business Cycle testing can be used as a basis for creating a series of transactions to support failover and recovery testing, primarily to define the tests to be run to test that recovery was successful.  • Power interruption to the client: power the PC down.  • Power interruption to the server: simulate or initiate power down procedures for the server.  • Interruption via network servers: simulate or initiate communication loss with the network (physically disconnect communication wires or power down network servers or routers).  • Interruption, communication, or power loss to DASD and DASD controllers: simulate or physically eliminate communication with one or more DASDs or controllers.  Once the above conditions or simulated conditions are achieved, additional transactions should be executed and, upon reaching this second test point state, recovery procedures should be invoked.  Testing for incomplete cycles uses the same technique as described above except that the database processes themselves should be aborted or prematurely terminated.  Testing for the following conditions requires that a known database state be achieved.  Several database fields, pointers, and keys should be corrupted manually and directly within the database (via database tools). Additional transactions should be executed using the tests from Application Function and Business Cycle Testing and full cycles executed.] |
| Oracles: | Program  Human |
| Required Tools: | The technique requires the following tools:   * base configuration imager and restorer * installation monitoring tools * backup and recovery tools |
| Success Criteria: | The technique supports the testing of:   * One or more simulated disasters involving one or more combinations of the application, database, and system. * One or more simulated recoveries involving one or more combinations of the application, database, and system to a known desired state.] |
| Special Considerations: | * Recovery testing is highly intrusive. Procedures to disconnect cabling (simulating power or communication loss) may not be desirable or feasible. Alternative methods, such as diagnostic software tools may be required. * Resources from the Systems (or Computer Operations), Database, and Networking groups are required. * These tests should be run after hours or on an isolated machine. |

### Configuration Testing

|  |  |
| --- | --- |
| Technique Objective: | [Exercise the target-of-test on the required hardware and software configurations to observe and log target behavior under different configurations and identify changes in configuration state.] |
| Technique: | * Use Function Test scripts. * Open and close various non-target-of-test related software, such as Microsoft Excel and Word applications, either as part of the test or prior to the start of the test. * Execute selected transactions to simulate actors interacting with the target-of-test and the non-target-of-test software. * Repeat the above process, minimizing the available conventional memory on the client workstation. |
| Oracles: | statistical oracle  specifications and documentation |
| Required Tools: | The technique requires the following tools:   * base configuration imager and restore * installation monitoring tools |
| Success Criteria: | The technique supports the testing of one or more combinations of the target test items running in expected, supported deployment environments. |
| Special Considerations: | * What non-target-of-test software is needed, is available, and what is accessible on the desktop? * What applications are typically used? * What data are the applications running; for example, a large spreadsheet opened in Excel or a 100-page document in Word? * The entire system’s netware, network servers, databases, and so on, also needs to be documented as part of this test. |

### Installation Testing

|  |  |
| --- | --- |
| Technique Objective: | Exercise the installation of the target-of-test onto each required hardware configuration under the following conditions to observe and log installation behavior and configuration state changes:   * new installation: a new machine, never installed previously with iBEX * update: a machine previously installed iBEX, same version * update: a machine previously installed iBEX, older version] |
| Technique: | * Develop manual scripts to validate the condition of the target machine.   + new: never installed   + same or older version already installed * Launch or perform installation. |
| Oracles: | Human |
| Required Tools: | The technique requires the following tools:   * base configuration imager and restorer |
| Success Criteria: | The technique supports the testing of the installation of the developed product in one or more installation configurations. |
| Special Considerations: | iBEX application has been successfully installed and no major software components are missing? |

# Entry and Exit Criteria

## Test Plan

### Test Plan Entry Criteria

The test plan starts with project

### Test Plan Exit Criteria

When project is finished.

### Suspension and Resumption Criteria

Project is suspended.

## Test Cycles

### Test Cycle Entry Criteria

New requirements are added, or specific design criteria is added to the

### Test Cycle Exit Criteria

Project finished.

### Test Cycle Abnormal Termination

Project is halted.

# Deliverables

Keywords test results

Risk results

Coverage tests

Runtime resource monitor

## Test Evaluation Summaries

The code coverage should be more than 80 percent of the code. Risks should be rationally controller and runtime resources should meet the target system boundaries.

## Reporting on Test Coverage

At the end of each iteration and after each device integration phase:

* Function coverage
* Condition coverage

Should be evaluated. These tests need Qt compatible coverage tools (i.e. Cococ Squish)

## Perceived Quality Reports

[Provide a brief outline of both the form and content of the reports used to measure the perceived quality of the product, and indicate how frequently they will be produced. Give an indication about to the method and tools used to record, measure, and report on the perceived product quality. You might include some analysis of Incidents and Change Request over Test Coverage.]

## Incident Logs and Change Requests

All risks and test must be recorded in this and risk analysis document.

## Smoke Test Suite and Supporting Test Scripts

## Additional Work Products

In this section, identify the work products that are optional deliverables or those that should not be used to measure or assess the successful execution of the **Test Plan**.

### Detailed Test Results

This denotes either a collection of Microsoft Excel spreadsheets listing the results determined for each test case, or the repository of both test logs and determined results maintained by a specialized test product.

### Additional Automated Functional Test Scripts

These will be either a collection of the source code files for automated test scripts, or the repository of both source code and compiled executables for test scripts maintained by the test automation product.

### Test Guidelines

During the test design critical faults and failure should be recorded in a risk analysis pattern. This document can be used to monitor in between tests and risks.

### Traceability Matrices

Using a tool such as MS Excel, opensource community contributors can provide one or more matrices of traceability relationships between traced items.

# Testing Workflow

The keywords should identify during the requirement analysis. Each word corresponds a high level usecase. During the system refinement and progressing with source code, each keyword is broken up into smaller unit test. These tests are going to be done by the programmer and the testers.

Identify usecase keywords

Programming

Testing

Unit tests

# Environmental Needs

This section presents the non-human resources required for the **Test Plan**.

## Base System Hardware

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Database Server |  |  |
| —Network or Subnet |  |  |
| —Server Name |  |  |
| —Database Name |  |  |
| Client Test PCs |  |  |
| —Include special configuration requirements |  |  |
| Test Repository |  |  |
| —Network or Subnet |  |  |
| —Server Name |  |  |
| Test Development PCs | 1 | Windows or Linux supported configuration |

## Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| Linux/Windows | 6/10 | Operating System |
| DCMTK licence | 4+ | Software package |
| Qt | 5+ | Software package |
| LCD screen | 15,20,22 inches | Monitor (medical) |
| DvTK | 3+ | Simulator |
| SQLite | 3 | DBMS |

## Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| Test Management |  |  |  |
| Defect Tracking |  |  |  |
| ASQ Tool for functional testing |  |  |  |
| ASQ Tool for performance testing |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |
| DBMS tools |  |  |  |

## Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration |  |  |
| Minimal configuration supported |  |  |
| Visually and mobility challenged |  |  |
| International Double Byte OS |  |  |
| Network installation (not client) |  |  |

# Responsibilities, Staffing, and Training Needs

This section presents the required resources to address the test effort outlined in the **Test Plan**—the main responsibilities, and the knowledge or skill sets required of those resources.]

## People and Roles

This table shows the staffing assumptions for the test effort.

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager |  | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst |  | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer |  | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester |  | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator |  | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager |  | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer |  | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer |  | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

## Staffing and Training Needs

This section depends on the organization strategy.

# Iteration Milestones

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed | - | - | - | - |
| Iteration starts | - | - | - | - |
| Requirements baselined | - | - | - | - |
| Architecture baselined | - | - | - | - |
| User Interface baselined | - | - | - | - |
| First Build delivered to test | - | - | - | - |
| First Build accepted into test | - | - | - | - |
| First Build test cycle finishes | - | - | - | - |
| [Build Two will not be tested] | - | - | - | - |
| Third Build delivered to test | - | - | - | - |
| Third Build accepted into test | - | - | - | - |
| Third Build test cycle finishes | - | - | - | - |
| Fourth Build delivered to test | - | - | - | - |
| Fourth Build accepted into test | - | - | - | - |
| Iteration Assessment review | - | - | - | - |
| Iteration ends | - | - | - | - |

# Risks, Dependencies, Assumptions, and Constraints

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Prerequisite entry criteria is not met. | Designer will define the prerequisites that must be met before Load Testing can start. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | Tester and designer will ensure a full set of suitable and protected test data is available. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | Tester will endeavour to ensure the Database is regularly refreshed. | * Restore data and restart * Clear Database |

| **Dependency between** | **Potential Impact of Dependency** | **Owners** |
| --- | --- | --- |
| Test plan success dependents on design team decisions. | High | Manager |
|  |  |  |
|  |  |  |

| **Assumption to be proven** | **Impact of Assumption being incorrect** | **Owners** |
| --- | --- | --- |
| The PACS and RIS emulator works on the same workstation machine | In real word these machines are separated | Tester, designer |
|  |  |  |
|  |  |  |

| **Constraint on** | **Impact Constraint has on test effort** | **Owners** |
| --- | --- | --- |
| Time. Test result should not take much time | high | Manager, tester |
|  |  |  |
|  |  |  |

# Management Process and Procedures

If the test was not successful, then the project design should be revisited to eliminate risks.

## Measuring and Assessing the Extent of Testing

All keywords should be tested.

## Assessing the Deliverables of this Test Plan

Test results and keyword list

## Problem Reporting, Escalation, and Issue Resolution

This document should be updated

## Managing Test Cycles

[Outline the management control process for a test cycle.]

Each new requirement should be reflected on this document to analyze the risks and test.

## Traceability Strategies

* Coverage of Testing against Specifications — enables measurement the extent of testing
* Motivations for Testing — enables assessment of relevance of tests to help determine whether to maintain or retire tests
* Software Design Elements — enables tracking of subsequent design changes that would necessitate rerunning tests or retiring them
* Resulting Change Requests — enables the tests that discovered the need for the change to be identified and re-run to verify the change request has been completed successfully

## Approval and Signoff

The technical members should sign the test results