**BSM Project**

**Test Plan**

Document Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Contributor** | **Description** |
| V1.0 | April 24, 2012 | Teresa Tang | Initial Draft |
|  |  |  |  |
|  |  |  |  |



**Table of Contents**

1 Introduction 3

1.1 Overview 4

1.2 Quality Objective 4

1.2.1 Primary Objective 4

1.2.2 Secondary Objective 4

1.3 Roles and Responsibilities 5

1.3.1 Project Manager/Product Owner 5

1.3.2 Developer 5

1.3.3 Tester 5

1.3.4 Testing Process Management Team 6

1.3.5 Adopter 6

1.4 Assumptions for Test Execution 6

1.5 Constraints for Test Execution 6

1.6 Definitions 7

2 Test Items 7

2.1 Overview 7

2.1.1 BSM Project Introduction 7

2.2 Features to Be Tested 8

2.2.1 System Configuration & deployment 8

2.2.2 Report Generation 9

2.2.3 NMP Reporting 9

2.2.4 Software PRM Reporting 9

2.3 Features Not to Be Tested List 9

2.3.1 Features not to be tested list 9

3 Test Strategy and Methodology 9

3.1 Purpose 9

3.1.1 Unit Testing 9

3.1.2 Module Integration Testing 10

3.1.2.1 Function Testing 10

3.1.2.2 UI Testing 10

3.1.2.3 Performance Testing 10

3.1.2.4 Reliability Testing 10

3.1.2.5 Localization Testing 10

3.1.2.6 Documentation Testing 10

3.1.3 Iteration/Regression Testing 10

3.1.4 Final release Testing 10

3.1.5 Testing completeness Criteria 11

3.2 Test Levels 11

3.2.1 Build Tests 11

3.2.1.1 Level 1 - Build Acceptance Tests 11

3.2.1.2 Level 2 - Smoke Tests 11

3.2.1.3 Level 2a - Bug Regression Testing 11

3.2.1.4 Level 2a- New Feature Testing 11

3.2.2 Milestone Tests 12

3.2.2.1 Level 3 - Critical Path Tests 12

3.2.3 Release Tests 12

3.2.3.1 Level 4 - Standard Tests 12

3.2.3.2 Level 5 - Suggested Test 12

3.3 Bug Severity and Priority Definition 12

3.4 Bug Regression 12

3.5 Bug Triage 13

3.6 Test Completeness 13

3.6.1 Standard Conditions: 13

4 Test Deliverables 13

4.1 Deliverables Matrix 14

5 Resource & Environment Needs 15

5.1 Testing Tools 15

5.1.1 Tracking Tools 15

5.1.1.1 Testing Cases Management Tool 15

5.1.1.2 Bug Tracker Tool 15

5.1.1.3 Configuration Management 15

5.1.2 Automation Testing Tools 15

5.2 Test Environment 15

5.2.1 Server Side 15

5.2.2 Client Side 15

6 Testing Resource and risk 16

7 Glossary 16

# Introduction

This test plan document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of software projects within BSM.

## Overview

The testing approach is to verify the functionality and features of BSM conform to general specification and requirements.

The purpose of the Test Plan is to achieve the following:

1. Define testing strategies for each area and sub-area to include all the functional and quality requirements.
2. Divide Design Spec into testable areas and sub-areas (do not confuse with more detailed test spec). Be sure to also identify and include areas that are to be not tested.
3. Define testing items and testing methodology.
4. Define testing process and bug-tracking procedures.
5. Identify required resources and related information.
6. Identify testing risks.

## Quality Objective

### Primary Objective

A primary objective of testing application systems is to: assure that the system meets the full requirements, including quality requirements and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

### Secondary Objective

The secondary objective of testing application systems will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

## Roles and Responsibilities

### Project Manager/Product Owner

* Team members: Marc Rochat; Tim Zhang
* Responsible to:
  1. Provide project requirement document to tester.
  2. Provide project function specification document to tester.
  3. Conduct Unit, system, regression and integration testing.

### Developer

* Team members: Ivan Che; Walter Xu
* Responsible to:
  1. Provide design documents to tester.
  2. Develop BSM system according to system architectural design and requirements
  3. Develop Use cases and requirements in collaboration with the Adopters
  4. Develop and execute Unit testing
  5. Support user acceptance testing

### Tester

* Team members: Teresa Tang;
* Responsible to:
  1. Write test plan, test spec and test report documents.
  2. Review design documents.
  3. Design testing cases in collaboration with the Adopters
  4. Execute testing cycles to assure product meet customer’s requirement.
  5. Develop testing scripts to improve testing efficiency.
  6. Support user acceptance testing

### Testing Process Management Team

* Team members: Jacky Yi
* Responsible to:
  1. Monitor and manage testing integrity and Support testing activities
  2. Coordinate activities across bugs track system.

### Adopter

* Responsible to:  
  1. Contribute to Use case, requirement development through review
  2. Contribute to develop and execution of the development test scripts through review

## Assumptions for Test Execution

* Testing cases have been designed in testing cases management tool and testing cases are reviewed and approved by test lead.
* Unit test is passed and build succeed.
* Planned new features development is completed and testable.
* Test scripts are developed and approved.
* Test Team will support and provide appropriate guidance to Adopters and Developers to conduct testing
* Major dependencies should be reported immediately after the testing kickoff meeting.
* Test results will be recorded and checked in Perforce. Failed scripts and defect list from bug tracking tools with evidence will be sent to Developer directly.

## Constraints for Test Execution

* Testers should clearly understand on test procedures and recording a defect or enhancement in bug track system.
* Developer will receive consolidated list of request for test environment set up, user accounts set up, data set (actual and mock data), defect list, etc. through bug track system after the initial testing kick off meeting.
* Developer will support ongoing testing activities based on priorities
* Test scripts must be approved by Test Lead prior test execution
* Test scripts, test environment and dependencies should be addressed during testing kickoff meeting.
* The Developer cannot execute the End to End test scripts. After debugging, the developer can conduct their internal test, but no results from that test can be recorded / reported.
* Testers are responsible to identify dependencies between test scripts and submit clear request to set up test environment

## Definitions

Bugs:

Any defect that cause software or hardware to malfunction. That is also included in the requirements and does not meet the required workflow, process or function point. Conflict with common sense also will be considered a bug.

Enhancement:

1) Any alteration or modification to the existing system for better workflow and process.

2) An error or defect that causes the software/application or hardware to malfunction.

Where 1) and 2) is NOT included in the requirements can be categorized as an enhancement.

Enhancement can be added as a new requirement after appropriate Change Management process.

# Test Items

## Overview

### BSM Project Introduction

The Business & Security Monitoring is a system allowing to generate business and security reports. The system itself is system agnostic as it can be used to generate reports for any system (e.g. NMP, PRM, CAS …).The system also has to be customized to match the system specific requirements. The following tasks have, among others, o be executed for each supported system:

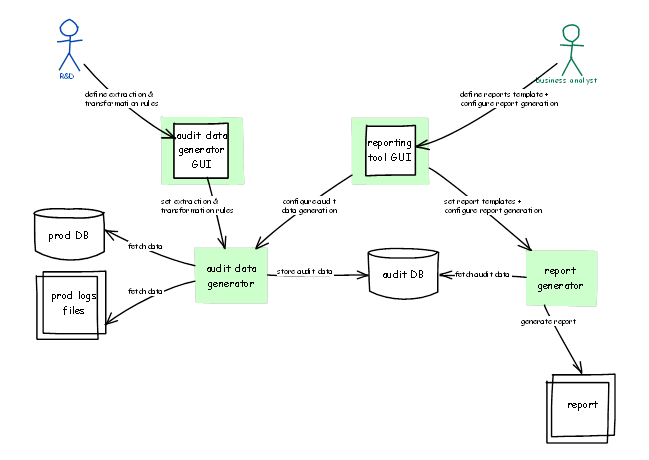
* Define production data that can be used in reports
* Define rules to extract this production date
* Define to rules to convert this production data into business or security relevant information
* Define report templates

The reports can, for example, contain

* The number of new users created over the last month,
* The total number of VOD products sold today
* Or identify devices that might be compromised.

Reports can be generated in several formats like PDF, Excel or web pages. The system allows user to control when and how reports are generated.

The schema below gives an overview of the system architecture.



## Features to Be Tested

The BSM project testing mainly focused on verification that if the system meet the system requirements, including functionality requirements and quality requirements.

### System Configuration & deployment

BSM system is deployed, configured and monitored using Nagra standard methods.

### Report Generation

This feature covers all steps of the reports generation which include (among others):

* Prepare audit data
* Generate audit data
* Prepare reports template
* Generate reports

### NMP Reporting

This feature concerns all requirements that are specific to NMP system monitoring.

### Software PRM Reporting

This feature concerns all requirements that are specific to software PRM system monitoring.

## Features Not to Be Tested List

### Features not to be tested list

TBD

# Test Strategy and Methodology

## Purpose

### Unit Testing

Unit testing is conducted by the Developer during code development process to ensure that proper functionality and code coverage have been achieved by each developer both during coding and in preparation for acceptance into iterations testing.

The following are the example areas of the project must be unit-tested and signed-off before being passed on to regression Testing:

* Databases, Stored Procedures, Triggers, Tables, and Indexes
* .DLL, .EXE and other binary formatted executable

### Module Integration Testing

#### Function Testing

Verify BSM function can work normally as requirement.

Verify the installer of BSM can be installed and configured normally.

#### UI Testing

Verify the BSM UI meet requirement.

#### Performance Testing

Verify BSM meet the basic performance goal (currently we don’t have a definition of performance goal), the performance metrics should include CPU, Memory, Disk IO and Network IO.

Verify reports generation doesn’t impact performances of the system for which the reports are generated.

#### Reliability Testing

We need to make sure BSM is robust and stable, so we need to do MTBF(Mean Time Between Failure) Testing for BSM.

#### Localization Testing

BSM is planned to have English and French version, so we need to do Localization Testing before French version release.

#### Documentation Testing

Some documents will be released, for example: Installation Guide, User Guide. It’s necessary to confirm these documents are precise and up to date.

### Iteration/Regression Testing

In each iteration cycle, a debriefing should be held. Specifically, the report must show that to the best degree achievable during the iteration testing phase, all identified severity 1 and severity 2 bugs have been communicated and addressed. At a minimum, all priority 1 and priority 2 bugs should be resolved prior to entering the next testing phase.

### Final release Testing

In this milestone process as well by providing confirmation feedback on new issues uncovered, and input based on identical or similar issues detected earlier. The intention is to verify that the product is ready for distribution, acceptable to the customer and iron out potential operational issues.

Assuming critical bugs are resolved during previous iterations testing- Throughout the Final Release test cycle, bug fixes will be focused on minor and trivial bugs (severity 3 and 4). Testing will continue its process of verifying the stability of the application through regression testing (existing known bugs, as well as existing test cases).

The milestone target of this phase is to establish that the application under test has reached a level of stability, appropriate for its usage that it can be released to the end users.

### Testing completeness Criteria

The milestone target is to place the release/app (build) into production after it has been shown that the app has reached a level of stability that meets or exceeds the client expectations as defined in the Requirements, Functional Spec., and BSM Production Standards.

## Test Levels

The test categories and test levels are defined below:

### Build Tests

#### Level 1 - Build Acceptance Tests

Other related test cases ensure that adopters received the proper Development Release Document plus other build related information (drop point, etc.). The objective is to determine if further testing is possible. If any Level 1 test case fails, the build is returned to developers un-tested.

#### Level 2 - Smoke Tests

The objective is to determine if further testing is possible. These test cases should emphasize breadth more than depth. All components should be touched, and every major feature should be tested briefly by the Smoke Test. If any Level 2 test case fails, the build is returned to developers un-tested.

#### Level 2a - Bug Regression Testing

Every bug that was “Open” during the previous build, but marked as “Fixed, Not reproduced” for the current build under test, will need to be regressed, or re-tested. Once the smoke test is completed, all resolved bugs need to be regressed.

#### Level 2a- New Feature Testing

The objective is to determine if the new feature of the iteration works normally as requirement, expose bugs as much as possible and raise improvements. New test cases will be designed and executed. These test cases should emphasize both breadth and depth. Bugs and improvements will be reviewed among testers, developers and project managers to decide if will be fixed and when will be fixed.

### Milestone Tests

#### Level 3 - Critical Path Tests

Critical Path test cases must pass by the end of every 2-3 Build Test Cycles. They do not need to be tested every drop, but must be tested at least once per milestone. Thus, the Critical Path test cases must all be executed at least once during the Iteration cycle, and once during the Final Release cycle.

### Release Tests

#### Level 4 - Standard Tests

Test Cases that need to be run at least once during the entire test cycle for this release. These cases are run once, not repeated as are the test cases in previous levels. Functional Testing cases can be tested multiple times for each Milestone Test Cycle (Iteration, Final Release, etc.).

Standard test cases usually include Installation, Data, GUI, and other test areas.

#### Level 5 - Suggested Test

Most Performance and Stress Test Cases are classic examples of Suggested test cases (although some should be considered standard test cases).Also include some special test required by developer or Management team or customer.

## Bug Severity and Priority Definition

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined. Tester will assign a severity and priority level to all bugs. The Test Lead will be responsible to see whether that a correct severity and priority level is assigned to each bug.

## Bug Regression

All bugs that are resolved as “Fixed, Not Reproduced” will be regressed when testing team is notified of the new drop containing the fixes. When a bug passes regression it will be considered “Closed, Fixed”. If a bug fails regression, adopters testing team will notify development team by entering notes into bug track system. The Test Lead will be responsible for tracking and reporting to development and product management the status of regression testing.

## Bug Triage

The Test Lead, Product Manager, and Development Lead should all be involved in these triage meetings. The Test Lead will provide required documentation and reports on bugs for all attendees. The purpose of the triage is to determine the type of resolution for each bug and to prioritize and determine a schedule for all “To Be Fixed Bugs’. Development will then assign the bugs to the appropriate person for fixing and report the resolution of each bug back into the bug tracker system. The Test Lead will be responsible for tracking and reporting on the status of all bug resolutions.

## Test Completeness

Testing will be considered complete when the following conditions have been met:

### Standard Conditions:

1. When Adopters and Developers, agree that testing is complete, the app is stable, and agree that the application meets functional requirements.
2. Script executions of all test cases in all areas have completed.
3. Automated test cases have in all areas have completed.
4. All priority 1 and 2 bugs have been resolved and closed.
5. Ad hoc testing in all areas has been completed.
6. The product management team approves the test completion

# Test Deliverables

Testing will provide specific deliverables during the project. These deliverables fall into three basic categories: Documents, Test Cases / Bug Write-ups, and Reports. Here is a diagram indicating the dependencies of the various deliverables:



As the diagram above shows, there is a progression from one deliverable to the next. Each deliverable has its own dependencies, without which it is not possible to fully complete the deliverable.

The following page contains a matrix depicting all of the deliverables that Testing will use.

## Deliverables Matrix

Below is the list of artifacts that are process driven and should be produced during the testing lifecycle. This matrix should be updated routinely throughout the project development cycle in project specific Test Plan.

|  |
| --- |
| **Deliverable** |
| **Documents** |
| Test Approach |
| 🡪 Test Plan |
| 🡪 Test Schedule |
| 🡪 Test Specifications |
| **Test Case / Bug Write-Ups** |
| Test Cases / Results |
| Test Coverage Reports |
| Bug tracker for bug reporting |
| **Reports** |
| Test results report |
| Test Final Report - Sign-Off |

# Resource & Environment Needs

## Testing Tools

### Tracking Tools

#### Testing Cases Management Tool

* Currently we use Fitnesse to manage testing cases.

#### Bug Tracker Tool

Currently we choose Jira to record and track bugs.

#### Configuration Management

Currently we use Perforce to do configuration management, use Jenkins to do continuous integration.

### Automation Testing Tools

* Fitnesse
* Design a Testing Framework for BSM

## Test Environment

### Server Side

1. Hardware: Virtual Machine (Oracle VirtualBox), memory size 1024M.
2. Software: RedHat5.5 64bit

JBoss430.6

jdk1.6.0\_20 64bit

### Client Side

1. Hardware: Dell Desktop
2. Software: Windows XP

IE7

Perforce

.

# Testing Resource and risk

1. This plan is based on the requirement and architecture described in SYSAD Reporting Tool.doc V0.1.Some features will be described better in next release of the document. Some features or changes may be added during development cycle, so testing activities will need to adjust for changes.
2. Current the automation cases rate is very low, regression testing take too long to finish.
3. Some features rely on other system, for example: NMP, PRM. Tester doesn’t have knowledge on these systems.
4. Currently the testing resource is not enough to meet the goal of this test plan.

# Glossary

|  |  |
| --- | --- |
| **Acronym Abbreviation** | **Definition** |
| BSM | Business & Security Monitoring |
| VOD | Video On Demand; umbrella term for a wide set of technologies whose common goal is to enable individuals to select video streams from a central server for viewing on a television or computer screen |
| NMP | Nagra Media Player |
| PRM |  |
|  |  |
|  |  |