|  |
| --- |
| **Test Strategy Document**  **(TSD)** |

Issue Date: 06/05/2019

**SAMProject**

Contents

[1. Test Strategy Document Purpose and Objectives 2](#_Toc10640785)

[2. Project Scope 3](#_Toc10640786)

[3. Test Objectives & Approach 6](#_Toc10640787)

[Test Objectives 6](#_Toc10640788)

[Test Approach 6](#_Toc10640789)

[Test Tools 8](#_Toc10640790)

[Test Products 8](#_Toc10640791)

[4. Roles & Responsibilities 8](#_Toc10640792)

[5. Next Phase Test Milestones 9](#_Toc10640793)

[6. Test Environment and Release Management 9](#_Toc10640794)

[Test Environment 9](#_Toc10640795)

[Code Merge Process 9](#_Toc10640796)

[Test Environment 10](#_Toc10640797)

[7. Test Deliverables 10](#_Toc10640798)

[8. Acceptance Process and Defect Management 10](#_Toc10640799)

[Defect Prioritisation 10](#_Toc10640800)

[8.1.1 Defect Severity 11](#_Toc10640801)

[8.1.2 Defect Impact 12](#_Toc10640802)

[9. Entry and Exit Criteria 12](#_Toc10640803)

[Entry Criteria 12](#_Toc10640804)

[Exit Criteria 12](#_Toc10640805)

[10. Dependencies 12](#_Toc10640806)

[11. Risks & Contingencies 13](#_Toc10640807)

[12. Assumptions & Constraints 14](#_Toc10640808)

[Assumptions 14](#_Toc10640809)

[13. Communication Plan 14](#_Toc10640810)

1. Test Strategy Document Purpose and Objectives

The purpose of the Test Strategy Document (TSD) is to define and describe the testing approach for the SAM Software project phase roll out.

The main objectives of the TSD are to:

* Establish an overarching and comprehensive framework and approach that covers all test planning, test design and test execution across all delivery test streams within the project.
* Describe the project test activities and deliverables.
* Present the planned test milestones, any assumptions and/or test dependencies to key project stakeholders, both internal and external.
* Highlight the entry and exit criteria for each of the testing phases.
* Define the test coverage management, defect management and other testing-related processes.
* Specify the test tools, test environments and test data requirements.
* Identify risks, impacts and mitigation actions.
* Establish the test team structure, roles and responsibilities required to complete the testing within the planned timeframes.

1. Project Scope

SAM will provide the agreed Capabilities and features, as listed in the below table.

|  |  |  |
| --- | --- | --- |
| **Capabilities** | **Ribbon (Features)** | **Phase** |
|  |
| **File Tab** | New |  |
| Open |  |
| Save |  |
| Saves as |  |
| Export |  |
| Back button |  |
| Window Minimize / Restore / Maximized |  |
| Window Title |  |
|  | |  |
| **Home Tab** | Dividers (4X) |  |
| Paste |  |
| Copy |  |
| Delete |  |
| Undo |  |
| Redo |  |
| Cut |  |
| Rename (Edit??) |  |
| Auto Recalculate |  |
| Insert |  |
| Find |  |
| Properties |  |
| Tabular View |  |
| Object Setup |  |
| Move Up |  |
| Move Down |  |
| Force Recalculation |  |
| Protect |  |
| Units |  |
| Show Ancestry |  |
| Level |  |
| Run |  |
| Record Macro |  |
|  | |  |
| **Layout Tab** | Design Explorer |  |
| Graphics |  |
| Messages pane |  |
| KCL Panel |  |
| Help |  |
|  | |  |
| **Temp Location Tab** | Link to Paramarine |  |
| Link to Excel |  |
| Input Valve |  |
| Output Valve |  |
| Trace Anscestor (Spelling mistake - Ancestor) |  |
| Trace Descendents |  |
| Remove Desc Trace |  |
| Remove Ancs Trace |  |
| Remove Traces |  |
|  | |  |
| **SAM Tab** | All Scenarios |  |
| Additional Scenarios |  |
| Random Scenario |  |
| Update Results |  |
| Compare |  |
| Optimise |  |
|  |  |  |
| **Quick Access Toolbar** | Save |  |
|  | Undo |  |
|  | Redo |  |
|  | Pause |  |
|  | Auto Recalculate |  |
|  | No Name (Pause) (BUG??) |  |
|  | Play (No name??? Bug ???) |  |
| **Quick Access Toolbar** | Save |  |
|  | Undo |  |
|  | Redo |  |
|  | Pause |  |
|  | Auto Recalculate |  |
|  | No Name (Pause) |  |
|  | Play (No name) |  |
|  | |  |
| **Settings Tab** | Decimal Places |  |
| Text  Text color  Protected  Constrained Input  Constrained Output  Overridden Output |  |
| Graphs  Probability Distribution  Histogram |  |
| Excel  Link In  Link Out |  |
| Max. number of columns [20,……20000] |  |
| Prompt user if rows exceed [9,999,999] |  |
| Auto remove previous messages (Automatically removes previous message) |  |
| Max. number of messages [10,000 limit] |  |
| Messages Plane  Auto remove previous messages  Max. number of messages [10,000] |  |
| Auto adjust the width of the design |  |
| Min. width |  |
| Max. width |  |
| Global Graph Series Settings |  |
| Zero |  |
| Hide All  Minor  Major |  |
| Legend |  |
| Legend Position  Top  Right  Bottom |  |
| Advanced  Edit/delete solid body split operations |  |
|  |  |
|  | |  |

Please refer to the Requirement estimation worksheet (Requirement Estimations.xlsx) for detailed scope.

1. Test Objectives & Approach

## Test Objectives

The primary objective for testing is to validate that the developed SAM solution conforms to the acceptance criteria agreed by QinetiQ as well as ensuring that:

* Functionality – SAM satisfies all mandatory requirements.
* Efficiency and Performance – SAM performs within acceptable response times and can support anticipated volumes in the future.
* Security and Privacy – SAM facilitates appropriate management of user data that complies with legal and regulatory requirements.
* Reliability and Availability – SAM is fault tolerant, available for use when required and has capabilities to recover from failovers.
* Usability – SAM facilitates friendly and easy user navigation and operation.

## Test Approach

A phased approach for testing will be adopted by Test Team, facilitating stage containment and a level of control over software quality. This quality control is achieved through the explicit use of Entry and Exit Criteria per test phase. The test phases are organised such that optimal quality is attained in the most optimum manner.

|  |  |  |
| --- | --- | --- |
|  | **PROJECTED MILESTONE** |  |
| **Phase 1** | | **Completion Period** |
|  |  |  |
|  | **Task to be Completed** |  |
|  | Understand the application | Completed |
|  | Walkthrough the application | Completed |
|  | Create a test environment | Completed |
|  | Experiment with the application | Completed |
|  | Request for the requirement and design materials | Completed |
|  | Request for the application to be installed on the testers machine | Completed |
|  | Capture the requirement | Completed |
|  | Break the requirement down in sections to be estimated and automated | Completed |
|  | Estimate the requirement in terms of days and number of resources required | Completed |
|  | Create a Gherkin BDD feature file which will be expanded in future phase. This document will then become the source of truth for QinetiQ. | Completed |
|  | Create a page object class for the object to automate | Completed |
|  | Create Test Method that tests a requirement as a proof of concept | Completed |
|  | Create a report of the test | Completed |
|  | Demo the Tests/Spike created | Completed |
|  | Get feedback from stakeholders | Completed |
|  |  |  |

**TEST APPROACH**

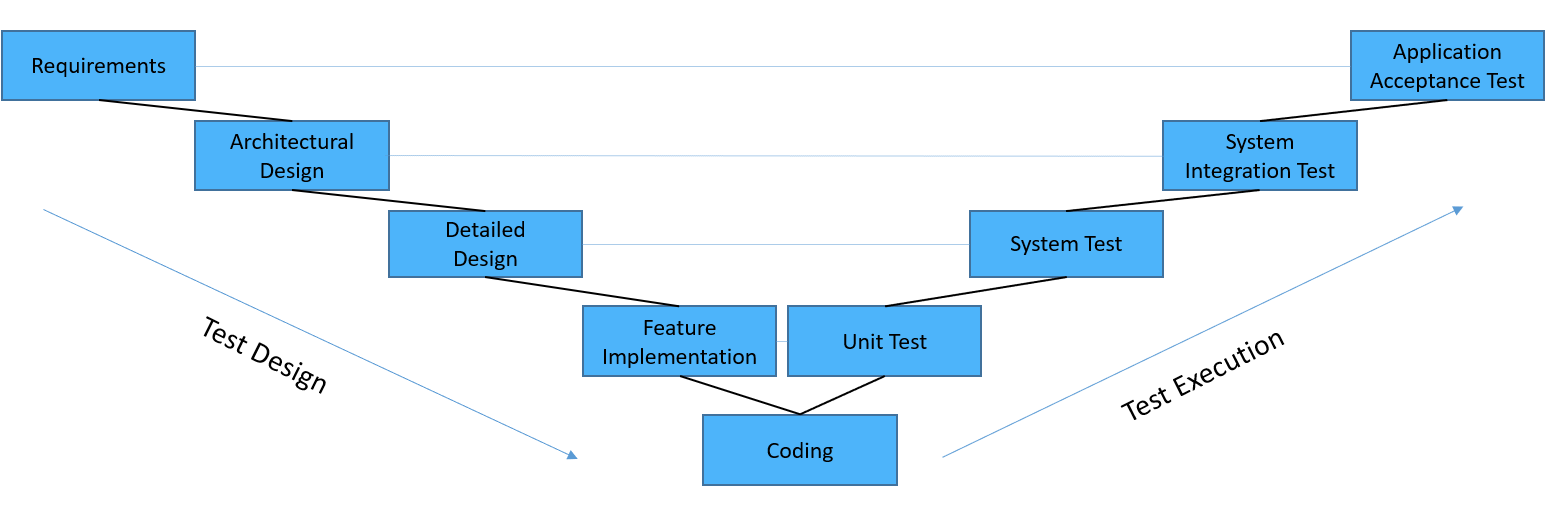
* Unit Testing (UT): It is performed in order to verify that each basic component or unit of code within SAM can function and conforms to its design specification.
* System Testing (ST): It demonstrates that SAM meets the specified requirements. Test case design should aim to specifically test each functional requirement.
* Regression Testing RT): Retesting of mandatory/critical, high priority requirements, normal flows of use cases, previous defects and functionality introduced in a recent release.

Test results using specflow, will be used as evidence that these activities have been successfully completed by SAM and will be provided for reference.

GUI Automation Test activity will be conducted with the aim to demonstrate SAM’s ability to meet the acceptance criteria for the various features agreed as in scope for delivery for Phase 1.

Software Testing Lifecycle

For the Software Testing Lifecycle (STLC), SAM follows the traditional V model methodology, as highlighted in the diagram below.



## Test Tools

Test Team will use Azure DevOps to manage all functional defects and business process issues raised during the test process.

The Test Team will use specflow to report and provide regular status updates to the SAM PM.

## Test Products

SAM is tested in steps, in line with the build and release strategies, from individual units of code through integrated subsystems, to the deployed releases and to the final system. Each phase of testing will have a product associated to it. A test product is defined in terms of scope, requirement coverage, risks mitigation, testing techniques, entry and exit criteria.

1. Roles & Responsibilities

The main test roles identified are:

|  |  |
| --- | --- |
| Role | Responsibility |
| Test Team  (TT) | Testing at every stage of the solution; this includes the creation of a Test Strategy Document.  Also is responsible for the following test components:   * Test Plan * Test Design & Script * Test Execution * Test Acceptance & Results.   Responsible for ensuring all requirements have been tested successfully, in line with the defined project test strategy and standards. |
| TestTeam  (TT) | * Will assume the responsibility for ensuring the release of the final solution, which will be done in the timescales agreed. * Ensuring that all steps in the process are documented and completed accordingly. |
|  |  |

1. Next Phase Test Milestones

Upon completion of Phase 1 Test activity, we will start the cycle, activity planned to commence as planned.

**Phase 2**

* Ability to read an XML file into the test script
* Ability to test graphs to ensure the information shown is as expected
* Ability to check a table cell (after double clicking on an object) shows the same data in the table as in the tree.
* Ability to check menu status when an object is selected – i.e. certain buttons will be greyed out, some will be available, a ribbon bar item will have focus.
* Functionalities as detailed in the Requirement Estimation.xlsx worksheet. Please refer to the excel (Requirement Estimation.xlsx)

1. Test Environment and Release Management

## 

## Test Environment

The following environments, Visual studio .NET framework, using WinAppDriver, Specflow, C#, Appium Web Driver, have been identified for SAM:

|  |  |
| --- | --- |
| **Name** | **Notes** |
| Test | Initial build & deployment.  Execution of GUI test. |
|  |  |

## Test Environment

* The Test Team will ensure that the test environment is updated after each refresh.

1. Test Deliverables

The below are the test deliverables for TSD:

|  |  |
| --- | --- |
| **Deliverable Name** | **Description** |
| Test Strategy Document (TSD) |  |
| Test Plan | GUI Automation Test Plan for SAM Phase 1 |
| Test Completion Report | Summary report that describes implementation testing, outcome of the test phase, summary of the test coverage, summary of the defects etc. |

1. Acceptance Process and Defect Management

SAM will use a standard set of definitions for classifying defects throughout the different levels of testing.

Each test case will have expected results defined. Defect management and test management processes, Azure DevOps will be used at the project level for acceptance testing.

The agreed process is that all test failures will be recorded directly by Test Team, once raised, SAM Team will be responsible for managing the fix process.

## Defect Prioritisation

Priority identifies the Defect importance by correlating impact and severity which will drive the required response and resolution time, as shown in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact** | **Severity** | | | |
| Critical | High | Medium | Low |
| Extensive | 1 | 1 | 2 | 4 |
| Significant | 1 | 2 | 3 | 4 |
| Moderate | 2 | 2 | 3 | 4 |
| Minor | 2 | 3 | 3 | 4 |

| **Priority** | **Business Definition** | **Target Fix Time** |
| --- | --- | --- |
| Critical | Fixing the defect before test phase closure is Imperative. System cannot go live with this defect. | Before end of GUI Automation Test for the Test. |
| High | Fixing the defect before test phase closure is mandatory.  System cannot go live with this defect | Before end of GUI Automation Test for the Test |
| Medium | Fixing the defect before test phase closure is desirable but not mandatory. | Before the beginning Stage testing, for the next cycle or Release. |
| Low | Fixing for the defect should be deferred to the next cycle or the next release | These defects need to have workarounds and ETA’s against the defects. Require PM approval. |

## Defect Severity

The approach for dealing with any defects and associated severity is below:

| **Severity** | **Definition** | **Response Time** |
| --- | --- | --- |
| 1-CRITICAL | **Critical** impact on the ability to use application and its components.  **Critical** impact on system users.   * System Crashes | 4 hours |
| 2-HIGH | **High** impact on the ability to use application and its components.  **High** impact on system users.   * Content on critical application is incomplete | 1 business day |
| 3-MEDIUM | **Medium** impact on the ability to use application and its components.  **Medium** impact on system users   * Desirable functionality does not work as specified, a work around exists, but requires additional administrative support. | 2 business days - or as agreed depending on volume of Severity 1 and 2 Defects |
| 4-LOW | **Low impact** on the ability to use application and its components.  **Low** impact on system users   * Spelling or format error on internal report - no material impact on the end users * Anything agreed as nonessential or redundant for successful system function and operation | 3 business days or as agreed depending on volume of Severity 1 and 2 Defects |

## Defect Impact

The below is the criteria for determining the impact:

|  |  |
| --- | --- |
| **Impact** | **Definition** |
| Critical | The change is immediately necessary to prevent severe business/ service/ user impact. |
| High | The change is needed as soon as possible because of potentially damaging business/ service/ user impact. |
| Medium | The change will solve re-occurring problems or repair missing functionality.  This change can be scheduled. |
| Low | The change will lead to improvements, changes in workflow, or configuration. This change can be scheduled. |

1. Entry and Exit Criteria

## Entry Criteria

* Functional Specification Document (FSD)
* Test Strategy Document (TSD)
* Test Environment + Test Data Readiness
* Test summary and Completion reports
* Mitigation actions identified for any outstanding severity 3 and 4 defects
* Test Plan is completed, reviewed and signed-off by the Project Manager and key stakeholders
* Test Environment shaken out, shakeout testing is passed and available for testing
* Test Management tool /Defect Management Tool Readiness

## Exit Criteria

* All agreed Test case have been executed and documented.
* Successful GUI Automation Test (at least 80% success rate)
* No critical, major and show stopper defects
* All open severity 3 & 4 defects have been reviewed and assigned a resolution timeframe
* All UAT defects have been documented in the defect management tool
* GUI Automation Test Summary Report has been formally signed off

1. Dependencies

For successful delivery of the test effort outlined in this test strategy/plan the following dependencies/pre-requisites must be delivered as scheduled to ensure that test activities can commence and/or complete as planned.

* Provision Test Team with access to test environment
* Provision Test Team with access to defect management system

1. Risks & Contingencies

The following risks and contingencies ascertain to the testing strategy:

|  |  |
| --- | --- |
| **Risks** | **Contingencies** |
| Testing environment not available | Establish a communication plan for environment issue resolution and escalations points. |
| Testing environment misconfigured | Establish a communication plan for environment issue resolution and escalations points. |
| Testing environment performance issues | Establish a communication plan for environment issue resolution and escalations points. |
| Test environment restrictions or differences relative to production | Establish a communication plan for resolution and escalations. |
| Connectivity issues with all the interfaces. | Where the availability of an environment instance is limited, test cases related to the particular system will be scheduled to be executed at the particular window where the instance is available. (i.e. interline partner bookings) |

1. Assumptions & Constraints

## Assumptions

The following assumptions and constraints have been identified:

* Test Team will provide defect resolution during testing activity and comply with the priority rating and expected turnaround times above described.

1. Communication Plan

This section describes the mechanisms and forums to be used in communicating progress and resolving issues across both parties throughout the testing lifecycle.

| Event | Agenda | Frequency | Target Audience |
| --- | --- | --- | --- |
| Test Phase Kick-Off | * Communicate test plans and schedules to test team * Provide Environment details, etc. | Once only at the beginning of a Test Phase | Test Team |
| Test Phase Kick Off | * Review and confirmation of test phase entry criteria | Once only at the beginning of a Test Phase | Project Manager, Test Team |
| Progress Tracking – Test Execution | * Planned vs Actual * Defects (Open / Closed / New etc.) | Monthly | to SAM PM |
| Defects Triage | * Outstanding defects * Prioritising defects * Defect Resolution | Monthly | With SAM PM, |
| Test Phase Closure | * Test Phase Exit criteria verification * Outstanding defects summary and resolution plan * Test Execution Summary and Coverage | Test Completion Report to be provided at the end of every test phase | to SAM PM |
| Testing Issues and Risks | * Discuss issues & risks. Mitigation strategies | Monthly | to SAM PM |