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

Issue Date: 07/06/2019

**SAM QA Project**

Contents

[1. Test Strategy Document (TSD) Purpose and Objectives 3](#_Toc10904727)

[2. Project Scope 4](#_Toc10904728)

[3. Test Objectives & Approach 6](#_Toc10904729)

[Test Objectives 6](#_Toc10904730)

[4. Test Automation Approach 7](#_Toc10904731)

[4.1 Phase 1 7](#_Toc10904732)

[4.2 Next Phase Test Milestones 8](#_Toc10904733)

[4.2.1 Phase 2 8](#_Toc10904734)

[5. Test Activities 8](#_Toc10904735)

[5.1 Software Testing Lifecycle 8](#_Toc10904736)

[5.2 Test Automation Tools 9](#_Toc10904737)

[6. Test Design Approach 9](#_Toc10904738)

[6.1 Quality of codes and tests 10](#_Toc10904739)

[6.2 Rate of Success and Performance report 10](#_Toc10904740)

[6.3 GUI Test results should be consistent and reliable 10](#_Toc10904741)

[7. Roles & Responsibilities 10](#_Toc10904742)

[8. Test Environment 11](#_Toc10904743)

[Test Environment 11](#_Toc10904744)

[9. Entry and Exit Criteria 11](#_Toc10904745)

[Entry Criteria 11](#_Toc10904746)

[Exit Criteria 11](#_Toc10904747)

[10. Risks & Contingencies 11](#_Toc10904748)

[11. Assumptions & Constraints 12](#_Toc10904749)

[Assumptions 12](#_Toc10904750)

[12. Communication Plan 12](#_Toc10904751)

1. Test Strategy Document (TSD) 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 the automation testing of the SAM solution is to validate that the application conforms to the acceptance criteria agreed by QinetiQ as well as validating the:

* Functionality – SAM satisfies all mandatory requirements.
* Efficiency and Performance – SAM performs within acceptable response times and can support anticipated volumes in the future.
* 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.

1. Test Automation Approach

A phased approach for testing will be adopted by Test Automation 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.

* 1. Phase 1

The agreed delivery for the phase 1 (tagged discovery phase) is listed below:

* Project Initiation and Decision to automate
* Requirement gathering and scoping
* Test Tool acquisition and setup
* Environment Setup
* Automation Test Strategy creation
* Test Process Analysis and Architectural design
* Test planning and Design
* Spikes (proof of concept) to automation a section of the application
* A demo /showcase of the proof of concept
* Project roadmap and resourcing in phases

All the end of the phased, the following tasks were completed or handed over to the client:

* Walkthrough of the application
* Creation and setup of the test environment
* Experiment with the application
* Request for the requirement and design materials
* Request and installation of the application on the testers machine
* Requirement gathering and scoping (further scoping with continue through phases)
* Break the requirement down in sections to be estimated and automated
* Estimate the requirement in terms of days and number of resources required
* Create a Gherkin BDD feature file which will be expanded in future phase. This document will then become the source of truth for QinetiQ.
* Create a page object class for the object to automate
* Create Test Method that tests a requirement as a proof of concept
* Create a report of the test
* Demo the Tests/Spike created
* Get feedback from stakeholders
* Project Initiation and Decision to automate
* Test Tool acquisition and setup
* Environment Setup
* Spikes (proof of concept) to automation a section of the application
* A demo /showcase of the proof of Concept
  1. Next Phase Test Milestones

Upon completion of Phase 1 Test activity, we will start the Phase 2, activity planned to commence as planned.

* + 1. 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 document. Please refer to the excel document (Requirement Estimation.xlsx)

1. Test Activities

The Test Automation Team will use specflow to report and provide regular status updates to the SAM PM at the end of each phase. 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.

* 1. Software Testing Lifecycle

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

* Unit Testing: 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. This will be performed by the developer on the project.
* System Testing: It demonstrates that SAM meets the specified requirements. Test case design should aim to specifically test each functional requirement. This will be performed by the system tester assigned to the project.
* Regression Testing: Retesting of mandatory/critical, high priority requirements, normal flows of use cases, previous defects and functionality introduced in a recent release. This will be performed by the system tester assigned to the project with the support of the automation tester. The code for the automation will be used for the regression testing.
* Automation Testing: This confirms that the SAM application meets the requirement and will be used for the regression by the system tester and smoke testing to be used by the developers.
  1. Test Automation Tools

The following tools will be used during the testing activities:

**Programming Language**

* + C#

**IDE**

* + Visual Studio

**UI Automation**

* + WinApp / Appium driver
  + Inspect.exe from Windows Kit
  + Page Object Model
  + Specflow
  + Nunit

**UI Automation Report**

* + Specflow Report
  + TFS report
  + Screenshots
  + Fail logs

**Continuous Development**

* + TFS / Azure DevOps

1. Test Design Approach

The proposed design approach is that we use a direct navigation approach to the test screen. This means that:

* Some tests do not necessarily need to start at the welcome screen or their parent screen
* Less clicks and navigations to the form for specific test
* Use navigations more for end to end testing
* Use KCL Parser where appropriate for a preconditions
* Use Page Object Model for reusability of the form objects
* Always tag tests to build cycle e.g. Smoke Testing, Integration, Regression, e.t.c
  1. Quality of codes and tests
* Create future proof test framework
* There is no point in creating non-maintainable codes
* Same language as the application under test
* For this reason, the responsibility of writing, maintaining and executing the tests becomes a shared responsibility with developers
* Easy to understand and manage structure
* Easily portable
* Handwritten codes that are easily to maintain
  1. Rate of Success and Performance report
* Create report that can be used to measure the rate of success and speed
* Visibility of Reports: The report should be visible to stakeholders and readily available on demand
* The metrics should be visible to all stakeholders
* Give a clear visibility to the stakeholders on what to expect in automating
* TOOLS: TFS (Azure DevOPs)
  1. GUI Test results should be consistent and reliable
* Test failure log should be relevant and useful for diagnosing the issue
* Use appropriate logging libraries
* Good and Reliable GUI tests
* The GUI test should be reliable and meaningful
* Simulating a user’s interaction with the application
* Fail and Pass reliably
* Fail Early
* Learn from failure as soon as it occurs
* Deliver to the consumer as soon as possible
* Get real and fast feedback
* Fail Better
* With early and frequent failures all that is needed is to maximise the learning opportunities
* Fix failure as soon as possible
* Review consistent failing tests

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 Automation Plan * Test Automation Design & Script * Test Automation Execution * Continuous Integration and Deployment * Test Automation Acceptance & Results.   Responsible for ensuring all agreed requirements have test framework created for the as defined within the project test strategy and quality standards. |
| TestTeam  (TT) | * Will assume the responsibility for ensuring the quality of the final test framework and handing over it to the client. * Ensuring that all steps in the test automation process are documented and completed accordingly. |
|  |  |

1. Test Environment

## Test Environment

The following environments: Visual studio .NET framework, using WinAppDriver, Specflow, C#, Appium Web Driver and TFS will be setup for the purpose of creating and running the test framework.

1. Entry and Exit Criteria

## Entry Criteria

* Functional Specification Document (FSD)
* Test Strategy Document (TSD)
* Test Environment

## Exit Criteria

* All agreed Test case have been executed and documented.
* Successful GUI Automation Test (at least 80% success rate)
* Demo has been done for the agreed tasks

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. |
| Application misconfiguration and installation | Establish a communication plan for issue resolution and escalations points. |
| Application performance issues | Establish a communication plan for issue resolution and escalations points. |
| Application objects restrictions or non-reliability of controls | Establish a communication plan for resolution and escalations. |

1. Assumptions & Constraints

## Assumptions

The following assumptions and constraints have been identified:

* Test Automation 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 | Frequency | Target Audience |
| --- | --- | --- |
| Discovery Phase (Phase 1) | Test Automation team work closely with the development team and visible on-site | Test Automation Team |
| Phase 2 -N | Communication as at when needed and a monthly demo  Test Automation shall work remotely | UI Team, Test Automation Team |
| Progress Tracking Test Execution | Monthly | SAM PM / Test Automation Team |
| Defects Triage | Monthly | With SAM PM, |
| Test Phase Closure | At the end of every phase | With SAM PM |