Test Strategy

**Prepared by:**

**Company Name**

**Project Management Office (PMO)**

Contents

[1. Revision History 3](#_heading=h.gjdgxs)

[2. Approvals 3](#_heading=h.30j0zll)

[3. Definitions 4](#_heading=h.1fob9te)

[4.](#_heading=h.3znysh7) Reference Documents 4

[5. Project Overview 4](#_heading=h.2et92p0)

[6. Test Plan Purpose 4](#_heading=h.tyjcwt)

[6.1 Testing Objectives 4](#_heading=h.2dlolyb)

[7. Scope 5](#_heading=h.3dy6vkm)

[7.1 In Scope Requirements 5](#_heading=h.1t3h5sf)

[7.2 Out of Scope Requirements 5](#_heading=h.sqyw64)

[8. Integrations and Intersystem Interfaces 5](#_heading=h.4d34og8)

[9. Test estimate 6](#_heading=h.3cqmetx)

[10. Testing Schedule 6](#_heading=h.1rvwp1q)

[10.1 Unit 6](#_heading=h.4bvk7pj)

[10.2 QA & SIT 6](#_heading=h.2r0uhxc)

[10.3 UAT 6](#_heading=h.1664s55)

[11. Test Environments 6](#_heading=h.2s8eyo1)

[11.1 Hardware Configuration 7](#_heading=h.17dp8vu)

[12. Test Data 7](#_heading=h.3rdcrjn)

[13. Test Users 8](#_heading=h.26in1rg)

[14. Testing Responsibilities 9](#_heading=h.lnxbz9)

[14.1 VMODEL with Accountability 9](#_heading=h.35nkun2)

[14.2 Roles and Responsibilities for Major Test Events 9](#_heading=h.1ksv4uv)

[15. Test Types 11](#_heading=h.44sinio)

[15.1 Unit testing 11](#_heading=h.2jxsxqh)

[15.2 Functional Integration Test (QA) 11](#_heading=h.z337ya)

[15.3 System Test 13](#_heading=h.3j2qqm3)

[15.4 User Acceptance Test (UAT) 14](#_heading=h.1y810tw)

[16. Performance Testing 16](#_heading=h.4i7ojhp)

[16.1 Performance Test Approach 16](#_heading=h.2xcytpi)

[16.1.1 Test Objective 16](#_heading=h.1ci93xb)

[16.2 Scope and Assumptions 16](#_heading=h.3q5sasy)

[16.2.1 In Scope 16](#_heading=h.25b2l0r)

[16.2.2 Out of Scope 16](#_heading=h.kgcv8k)

[16.3 Business Volume Metrics 16](#_heading=h.3whwml4)

[16.3.1 Transaction SLAs 16](#_heading=h.2bn6wsx)

[16.4 Performance Testing Methodology 16](#_heading=h.qsh70q)

[16.4.1 Test Scripting Entrance Criteria 16](#_heading=h.3as4poj)

[16.4.2 Test Execution Entrance Criteria 16](#_heading=h.1pxezwc)

[16.4.3 Performance Test Runs 16](#_heading=h.49x2ik5)

[16.4.4 Test Description 16](#_heading=h.2p2csry)

[16.4.5 Workload Mix Distribution 16](#_heading=h.147n2zr)

[16.5 Test Schedule 16](#_heading=h.34g0dwd)

[16.6 Key Milestones 16](#_heading=h.1jlao46)

[16.7 Environment Setup 16](#_heading=h.43ky6rz)

[16.7.1 Application Environment Setup 16](#_heading=h.2iq8gzs)

[16.7.2 Test Tool Setup 16](#_heading=h.xvir7l)

[16.8 Tools used in the project 17](#_heading=h.3o7alnk)

[16.8.1 Silk Performer – Controller and Agent details 17](#_heading=h.23ckvvd)

[16.8.2 Tool Settings for Load Test 17](#_heading=h.ihv636)

[16.9 Performance Test Data 17](#_heading=h.32hioqz)

[16.9.1 Test Data Definitions 17](#_heading=h.1hmsyys)

[16.10 6 Performance Metrics 17](#_heading=h.41mghml)

[16.10.1 Client-Side Metrics 17](#_heading=h.2grqrue)

[16.10.2 Server-Side Metrics 17](#_heading=h.vx1227)

[16.10.3 Server-Side Applications 17](#_heading=h.3fwokq0)

[16.10.4 Server Side Monitoring Counters 17](#_heading=h.1v1yuxt)

[16.11 Test Deliverables 17](#_heading=h.4f1mdlm)

[16.12 Status and Issue Reporting 17](#_heading=h.2u6wntf)

[17. Project Testing Related Tools 17](#_heading=h.19c6y18)

[18. Defect Management 18](#_heading=h.3tbugp1)

[19. Objectives of the defect review meetings 18](#_heading=h.28h4qwu)

[19.1 Purpose of the defect review meeting 18](#_heading=h.nmf14n)

[19.2 Defect reporting and resolution process 18](#_heading=h.37m2jsg)

[19.3 Defect escalation procedure 19](#_heading=h.1mrcu09)

[19.4 Defect severity definitions 20](#_heading=h.46r0co2)

[19.5 Defect life cycle stage 21](#_heading=h.2lwamvv)

[20. Results and metrics reporting 23](#_heading=h.111kx3o)

[21. Communication and Escalation 24](#_heading=h.3l18frh)

[22. ASSUMPTIONS/CONSTRAINTS/RISKS/ISSUES 25](#_heading=h.206ipza)

[22.1 Assumptions 25](#_heading=h.4k668n3)

[22.2 Constraints 25](#_heading=h.2zbgiuw)

[22.3 Issues 25](#_heading=h.1egqt2p)

[22.4 Risks 25](#_heading=h.3ygebqi)

# Revision History

| **Version No.** | **Date** | **Revised By** | **Description of Change** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Approvals

The undersigned acknowledge that they have reviewed the Master Test Plan and agree with the information presented within this document. Changes to this plan will be coordinated with, and approved by, the undersigned, or their designated representatives. The Project Sponsor will be notified when approvals occur.

| Signature: |  | Date: |  |
| --- | --- | --- | --- |
| Print Name: | Janina Johnson |  |  |
| Title: | Deputy PMO Director |  |  |
| Role: | Program PMO Director |  |  |

| Signature: |  | Date: |  |
| --- | --- | --- | --- |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: | Program Director |  |  |

| Signature: |  | Date: |  |
| --- | --- | --- | --- |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: | Test Manager |  |  |

| Signature: |  | Date: |  |
| --- | --- | --- | --- |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: | PMO TCOE Auditor |  |  |

# Definitions

| **Term** | **Meaning** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Reference Documents

| **Documents** | **Repository Path** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Project Overview

# Test Strategy Purpose

# Application Scope

## In Scope Applications

*<List the applications or technologies that are covered by this Test Strategy >*

| Application Name | Application Owner | Technology |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Integrations and Intersystem Interfaces

The following tabular contents will list down the various Interfaces/Applications involved in the Integration Testing of Superdome Project and also contains the individual point of contact that will be used for coordinating any Integration Testing.

A diagram might work better or nice to have in addition

| **System ID** | **Application/Functional Area** | **Testing Responsibility/SME** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Test Environments

The following diagram identifies the environments used for testing.

Identify environment to be used for production fixes v. development

## Hardware Configuration

Assigned “Infrastructure” contact name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  | **Test** | **Stress** | **QA** | **Production** |
| --- | --- | --- | --- | --- |
| Application Server  Box #, Memory, CPU |  |  |  |  |
| Weblogic Configuration |  |  |  |  |
| Database Server |  |  |  |  |

The following diagrams identify the environments used for testing compared to production. Identify environment to be used for production fixes, development

* Productive Environment

<Insert Diagram Here>

* QA Environment

<Insert Diagram Here>

* Test Environment

<Insert Diagram Here>

* Stress Environment

<Insert Diagram Here>

# Test Data

Test Lead(s) will define high level data requirements as needed for testing for the key areas. Identify unique data required by the application such as User Ids and Passwords. Attention to detail such as drop down boxes and user entered information will need to be passed along to the performance tester. Also determine if DB needs to be seeded with additional data in order to more closely replicate production. For example if the production DB has 13 million rows of data in production the same amount should be used in test.

* + Describe data needed by environment
  + Test data refresh requirements
  + Data seeding requirements
  + Data requirements for end-to-end business processes
  + Means by which Certify Data can be utilized for test data creation during development execution

| **Application/Service** | **Assigned Resource** | **Data Requirements** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Test Users

Each test case will require one or more test users. Test users must be created to replicate real business users allowing defects related to authorisation profiles and delegation of duties to be identified. Using unrealistic role assignment for test users will invalidate all functional tests.

A catalog of test users should be maintained. Automatic provisioning of test users needs to be established as part of the setup of the test environment.

# Test Types

## Unit testing

| Purpose | This preliminary test is performed by the development team for testing of individual configuration, custom programs and/or technical services (e.g. Fax, EDI etc.) to ensure that they function according to the detailed technical specification.  Unit test is a white box test and should test all possible flows. Both positive and negative conditions should be tested. |
| --- | --- |
| Development Phase | Development and Testing |
| Test Scope | All configurations, code validation, memory testing, integration, code complexity, etc. |
| Test Environment | Development Environment |
| Test Data | Manual data created by developers |
| Interface Requirements | NA |
| Role | Developer |
| Entry Criteria | * Formal reviews for process models, functional spes and technical specifications have been completed * All Inspection related defects have been corrected * All documentation and design of the architecture must be made available * Development of the component is complete and compiles without error * All Unit test cases are documented |
| Exit Criteria | * All Unit test cases completed successfully * All source code is unit tested * No outstanding critical defects * All outstanding defects are entered into the defect tracker * All test results have been documented |

## Functional Integration Test (QA)

| Purpose | Functional test validates that full operability of interconnected functions, methods or objects within a functional area. This includes a set of logically related activities or business processes to achieve a defined business process.  Functional test cases will typically consist of a series of business processes or stories joined together to achieve a business process. The smaller size of test cases will enable testing multiple data sets and permutations.  It happens after or in parallel with the development phase as and when all components for a specific flow are complete. Functional tests will be done by an independent testing team in a QA environment.  During subsequent integration testing activities these business process (functional) tests are combined to build end-to-end integration test scenarios. |
| --- | --- |
| Development Phase | Development and Testing |
| Test Scope | All functional tests, requirement/story coverage using test design techniques like Orthogonal Analysis, Decision Tables, Equivalence Partitioning, etc. |
| Test Environment | Test Environment |
| Test Data | Manual data created by Test team |
| Interface Requirements | Interface connectivity required for impacted systems |
| Role | QA Team |
| Entry Criteria | * All specs are frozen and the requirements change control process has begun * Proper test data is available * Test plans and test cases are reviewed and signed off * Unit Testing has been completed * Specifications for the product have been completed and approved * All test hardware platforms must have been successfully installed, configured and functioning properly. * All standard software tools including testing tools must have been successfully installed and functioning properly. * All personnel involved in the system test effort must be trained in tools to be used during the testing process. * All personnel involved in the system test effort must be trained the usage of the application and new features. * All functional test cases are documented |
| Exit Criteria | * Test case execution completed with 90% passed * All defects are recorded in Quality Center or Solution Manager * No outstanding “showstopper or severe” defects * All test results have been documented * All code has been migrated into the QA environment * Coverage of code/functionality/**requirements** is 100% of functional requirements. |

## System Test

| Purpose | SIT validates a set of business processes that define a business scenario in a comprehensive and self-contained manner on a macro level.  This is an end-to-end test of the business process. Typically a business scenario will involve testing multiple SAP modules test cases together. The primary objective of this testing, is to discover errors in the integration between different modules and to verify that the modules work together correctly as one function. E2E test validates the integration within SAP and between SAP and Legacy (all Non-SAP) applications. All testing related to validation of the data interchange between SAP and Legacy applications are categorized as Interface testing.  Security role based authorization test is performed to ensure that all the security profiles and roles are being implemented as designed. Security profile is designed and built based on the job role (i.e., positions) of the end users. Security roles are assigned at the business transaction level.  The objectives of security testing are   * Ensure that user has access to the required transactions to perform their job * Ensure that the user does not have access to transactions other than what is required for the role * Ensure that accesses to critical system administration transactions are controlled. * Ensure that only authorized person has the right to view the information on Screens and Reports. * Ensure that Delegation being done in SAP (where ever system allows user to delegate his authority to other user/s) are tested from the viewpoint of the Delegator and to whom it is being delegated. |
| --- | --- |
| Test Scope | * Full End to end business process * Performance Testing * Regression * Interface testing with interfacing systems * Security role based authorization testing * End to End scenarios executed with user id mapped to actual security roles * batch jobs execution using scheduled runs * Printers and other devices |
| Development Phase | Development and Testing |
| Test Environment | QA Environment or Pre-Prod |
| Test Data | Data from Mock cutover or Test Data Management tool |
| Interface Requirements | Interface connectivity required for all interfacing systems |
| Role | QA Team |
| Entry Criteria | * All specs are frozen and the requirements change control process has begun * Proper test data is available * Test plans and test cases are reviewed and signed off * SIT 0 has been completed * All functional test cases are documented |
| Exit Criteria | * Test case execution completed with 100% passed * All defects are recorded in Quality Center or Jira * No outstanding “showstopper or severe” defects * All test results have been documented * All code has been migrated into the Pre-Prod environment * No new defects have been discovered for a week prior to System Testing. * Coverage of code/functionality/**requirements** is 100% of functional requirements. |

## User Acceptance Test (UAT)

| Purpose | User acceptance test is performed by business users. The users test the complete, end-to-end business processes to verify that the implemented solution performs the intended functions and satisfies the business requirements. |
| --- | --- |
| Development Phase | Final Prep or Implementation |
| Test Scope | * UAT * Full Regression |
| Test Environment | Pre-Prod or Implementation |
| Test Data | Mock cutover or Test Data Management tool |
| Interface Requirements | Interface connectivity required for all interfacing systems |
| Role | Process Team & Business Users |
| Entry Criteria | * The application works functionally as defined in the specifications * No outstanding “showstopper or severe” defects * All areas have had testing started on them unless pre agreed by UAT stakeholder/Test and Project managers * Entire system functioning and all new components available unless previously agreed between UAT stakeholder/Test manager and project managers * All test cases are documented and reviewed prior to the commencement of UAT |
| Exit Criteria | * The Acceptance Tests must be completed, with a pass rate of not less than 98%. * No outstanding “showstopper or severe” defects * Less than 5 significant defects outstanding * All Test cases have been complete * No new defects have been discovered for a week prior to Production Implementation. * All test results recorded and approved * UAT test summary report documented and approved * UAT close off meeting held. |

# Performance Testing

## Performance Test Approach

### Test Objective

## Business Volume Metrics

### Transaction SLAs

## Performance Testing Methodology

### Test Scripting Entrance Criteria

### Test Execution Entrance Criteria

### Performance Test Runs

### Test Description

### Workload Mix Distribution

## Tools used in the project

### Controller and Agent details

### Tool Settings for Load Test

## Performance Test Data

### Test Data Definitions

## 6 Performance Metrics

### Client-Side Metrics

### Server-Side Metrics

### Server-Side Applications

### Server Side Monitoring Counters

## Test Deliverables

## Status and Issue Reporting

# Project Testing Related Tools

| Phase/activity | Test tool requirement |
| --- | --- |
| Test case documentation (Manual & Automation) | Konoah or Zephyr |
| Requirement Management | JIRA |
| Test cases automation development and execution | Sahi or Worksoft Certify |
| Test execution and results reporting | Konoah or Zephyr |
| Defect reporting and tracking | Jira |
| Document storage | Confluence |
| Business Process Flow | BizAgi |
| Test Data Management | IBM Optim |
| Service Virtualization | NA |
| Code Analysis |  |
| Code profiler |  |
| Code Coverage |  |
| System Monitoring (during performance testing) |  |
|  |  |
|  |  |

# Defect Management

Defect Review Meetings will be held on a daily basis with SME leads, test leads from all location, test managers, business leads and integration manager. The goal of this meeting is to ensure that defects are being resolved in a timely fashion and that any issues or questions are resolved. It is at these meetings that progress tracking of defect resolution and closure is communicated.

# Objectives of the defect review meetings

## Purpose of the defect review meeting

* To help prioritize defect fixes for Implementation, Legacy support, and Conversion teams.
* To discuss and assign priority and severity to defects, discuss the expected turnaround time and the planned turnaround time.
* To monitor and review the progress of defect fixes that is due or overdue as of current date.
* To determine the extent of retesting required due to a fix/ enhancement.
* To escalate defects/ issues to PMO when a quick resolution is required, or in case of a deadlock on ownership of defects/ issues.
* To identify whether defect is assigned to right team
* Identify defects that need to be deferred to subsequent releases

## Defect reporting and resolution process

**Prerequisite:** Development team & Business Team should have access to defects section of Jira and are able to update the defect details. Quality center should be configured to send auto emails when a new defect is logged, assignee is changed and the status is moved to re-test

Below diagram help in understand the defect life cycle process quickly and easily. Defect severity definition and recommended SLAs are available in appendix. Various stages of defects and the subsequent stages are also listed in appendix in detail.



Defect life cycle

## Defect escalation procedure

Below table provides information on when to escalate a defect

| Defect Severity | # Blocking test cases | Slipped SLA | Candidate for escalation |
| --- | --- | --- | --- |
| Any Level | >10% of total test cases | Yes | C:\Users\kiran_duvvuri\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0ZZO05LW\MC900433800[1].png |
| Critical | >5% of total test cases | Yes | C:\Users\kiran_duvvuri\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0ZZO05LW\MC900433800[1].png |
| Any Level | Any number | Yes / Go–No Go meeting is scheduled within 5 days from current day | C:\Users\kiran_duvvuri\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\0ZZO05LW\MC900433800[1].png |

#### Defect communication and escalation procedure

**First level of notification**: As soon as the defect is logged in to quality center, auto generated email would be sent to the assigned person. Since the defect will be assigned to development team alias, all the team who are subscribed to the alias would get the email.

**Daily status review meeting**: Along with the test execution status discussions, all the outstanding defects would be discussed in the meeting. Development team, business team, basis team, QA management and other stakeholders as appropriate would join the meeting. Defect details and estimated time of fix would be documented in the quality center accordingly.

**Defect disposition meeting**: this is a twice a week meeting where in only high impact defects as identified are the candidates for escalation would be discussed in detail. Development team management, QA team management along with respective leads would discuss the finer details and put an action plan to resolve them.

**Escalation email to development team/SME team manager**: QA Manager from would send an email with details of defects which need immediate attention to development team/SME team manager and on need bases a triage call involving senior management would be organized to discuss associated risks, have a resolution plan, and to review the status.

*Note:*

1. *Above mentioned escalation criteria can be adjusted during execution based on number of days left for the release go-no go decision.*

## Defect severity definitions

| Severity | Definition | Expected time for Closure |
| --- | --- | --- |
| Critical | A complete software system, or a subsystem, or software unit (program or Module) within the system lost its ability to perform its required function (=Failure) and no workaround available  OR  Testing of a significant number of tests cannot continue without closure  OR  Potential show stopper for Go/ No-Go decision to enter next stage or Cutover without closure | 1 Business Day |
| Major | The software system, or subsystem, or software unit (program or module) within the system produces Incorrect, Incomplete, or Inconsistent results  OR  Defect impairs the usability (capability of software to be understood, learned, used and attractive to the user when used under specified conditions [ISO 9126] | 2 Business days |
| Minor | Everything that not Major or Critical | 3 Business days |

## Defect life cycle stage

As part of the Defect Life Cycle definition and Defect Management process, various Defect stages will be identified as mentioned below

| Defect Status | Description | Required Previous Status | Next Possible Status |
| --- | --- | --- | --- |
| New | - Defect identified and raised by a Team - Defect is not reviewed by the Assigned Team | NA | Open Assigned |
| Open | - Assigned Team acknowledges the defect by moving the defect to open status - No one has been assigned to analyze the defect | New | Assigned Rejected Deferred Duplicate |
| Assigned | -Defect is assigned to a user (developer) for analyses. | New  Open | Need more info Rejected Deferred Duplicate Fixed Retest |
| Need More Info | - Defect is assigned to the tester for getting additional information about the problem for more analysis. | New  Open | Assigned Rejected Deferred Duplicate Fixed Retest |
| Rejected | - An invalid defect has been logged. The defect can be rejected by the Assigned Team for various reasons  - Invalid data used by tester  - Invalid test case executed by tester  - Test steP&A followed by the tester were incorrect  Note: If the defect is rejected because requirements were changed and the Testing team was not notified for the update requirement, then the defect shouldn't be rejected. It should be Closed | Open  Assigned | Assigned |
| Fixed | - Assigned Team moves the defect to fixed when the defect is fixed and is ready to be deployed | Assigned | Retest |
| Retest | - Assigned team moves the defect to Retest when the defect has been deployed for testing on the required environment | Fixed | Closed Re-Open |
| Re-Open | - If a defect in Retest Fails, then the defect is Reopened and assigned back to the previous team which fixed the defect Note: If the retest of the defect fails because of a reason different than what the defect was logged for then a new defect should be open for the new issue. The current defect shouldn't be reopened in such cases | Retest | Assigned Fixed Retest |
| Closed | - Defect passes the retest and can be closed | Fixed Retest | <NOCHANGE> |
| Deferred | - Defect is acknowledged by the Assigned Team and cannot be fixed with the Release timeline because of any constraints. The defect then will be deployed to production with known risk - To be able to move a defect to Deferred status an approval from all the key stakeholders is required - A CR needs to be initiated for doing the change in future - The approval email for deferring the defect needs to be attached as a part of the defect | Assigned Re-Open | <NOCHANGE> |
| Duplicate | - The defect is a duplicate of and existing open defect and is same as the previous one.  - Previous defect Id needs to be updated in this case | Open Re-Open | Assigned |

# Results and metrics reporting

Below listed metrics would be published to provide Testing CoE stakeholders with an update and status of the release.

| Report Name | Details | Frequency |
| --- | --- | --- |
| Weekly status report | * PMO status report to be sent every Tuesday by 3.00 * Participate in Thread leads status meeting every Wednesday | Weekly |
| Daily Status Report during test execution phase | Test Execution status of all the tracks. This report contains   * Test execution planned Vs completed * Test Case pass/fail numbers, * Defects open/close , age and severity * Risks and Issues * Milestone achievements | Every working day |
| Defect Churn Rate | * Need definition from Richa |  |
| QA/UAT Velocity Rate | * Need definition from Richa |  |
| Closure Report | * Summary of test execution phase just concluded to all stakeholders for their sign off | End of each test phase |

# Communication and Escalation

Below details will provide a view of how communication and escalation can be done against IBM QA team

| Category | Type | Participants | Mode | Type of reporting |
| --- | --- | --- | --- | --- |
| Bi-Weekly project meeting | Project | * test lead * IBM test manager * IBM test lead | Telephonic conference | * High level project status, * Key issues and risks, * Action tracker |
| Weekly status meeting | PMO | * test lead * IBM test lead | Telephonic conference | * Progress as against plan * Key issues and risks * Action tracker |
| Daily status reporting | Project | QA stakeholders  IBM QA team | Email | Daily reporting of tasks and progress of the same against plan |

Escalation hierarchy –

| Name | Role | Issue Age | Email address |
| --- | --- | --- | --- |
| Track Leads | Thread leads | >3 days |  |
|  | Test Manager | >5 days |  |
| PMO |  | > 7days |  |

# ASSUMPTIONS/CONSTRAINTS/RISKS/ISSUES

Assumptions, constraints, risks and issues are external factors that the decision maker has little or no control over, thus they inhibit the decision making process.

## Assumptions

Assumptions are entered using Microsoft® SharePoint Decision Data Entry Form that populates the Decision Log.

## Constraints

Constraints are entered using Microsoft® SharePoint Decision Data Entry Form that populates the Decision Log.

## Issues

Issues are entered using Microsoft® SharePoint Issue Data Entry Form that populates the Issue Register/Log.

## Risks

Risks are entered using Microsoft® SharePoint Risk Data Entry Form that populates the Risk Register/Log.