

FUNCTIONAL EXCELLENCE TESTING

TEST STRATEGY - Orbital – SCheduling

soft launch release

25 July 2023

Table of Contents

[DOCUMENT DETAILS 4](#_Toc140573252)

[PROJECT DETAILS 4](#_Toc140573253)

[VERSION DETAILS 4](#_Toc140573254)

[REVIEWERS 4](#_Toc140573255)

[REVISION HISTORY 4](#_Toc140573256)

[MANAGEMENT SUMMARY 5](#_Toc140573257)

[INTRODUCTION 6](#_Toc140573258)

[ABOUT THE PROJECT 6](#_Toc140573259)

[PROJECT CHARTER 6](#_Toc140573260)

[ARCHITECTURE DIAGRAM / SAHLD (Solution Architecture High Level Design) 7](#_Toc140573261)

[TESTING APPROACH 8](#_Toc140573262)

[OVERALL TESTING APPROACH 8](#_Toc140573263)

[SPRINT DETAILS 8](#_Toc140573264)

[PROJECT SCHEDULE 10](#_Toc140573265)

[PROJECT (TESTING) TIMELINE 10](#_Toc140573266)

[TESTING SCOPE 11](#_Toc140573267)

[WHAT IS TO BE TESTED? (IN SCOPE) 11](#_Toc140573268)

[WHAT IS NOT TO BE TESTED? (OUT OF SCOPE) 12](#_Toc140573269)

[Risk Based Testing approach 14](#_Toc140573270)

[TEST PHASES 14](#_Toc140573271)

[PERFORMANCE TEST STRATEGY 18](#_Toc140573272)

[IT ACCESSIBILITY TESTING 20](#_Toc140573273)

[TEST DATA MANAGEMENT 21](#_Toc140573274)

[TEST ENVIRONMENT 22](#_Toc140573275)

[DEFECT MANAGEMENT PROCESS 23](#_Toc140573276)

[TEST MONITORING / REPORTING 25](#_Toc140573277)

[TEST TOOLS 28](#_Toc140573278)

[AUTOMATION SCOPE 29](#_Toc140573279)

[IMPORTANT ROLES IN THE PROJECT 30](#_Toc140573280)

[RACI MATRIX 33](#_Toc140573281)

[ASSUMPTIONS, CONSTRAINTS, AND DEPENDENCIES 35](#_Toc140573282)

[RISKS 37](#_Toc140573283)

[GLOSSARY 38](#_Toc140573284)

# DOCUMENT DETAILS

## PROJECT DETAILS

|  |  |
| --- | --- |
| **Project Name:** Orbital - Scheduling | **Project ID:** PRJ3298772 |
| **Line of Business:** SLMT DnA (Downstream) | **Investment Archetype:** High Impact |
| **Project Manager:** Hiran Ilangantileke | **Test Manager:** Girish Pai |
| **Methodology:** Agile | **CIO Top Project:** Yes |

## VERSION DETAILS

|  |  |
| --- | --- |
| **Version Number:** | *2.0* |
| **Draft/Final as of:** | *Final / 17th July 2023* |
| **Author:** | *Girish Pai* |
| **Owner:** | *Girish Pai* |

## REVIEWERS

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Role | Date |
| Reviewed By | *Sapna Ranganath* | *SLMT DnA Testing CC Lead* | *10th July 2023* |
| Reviewed By | *Joydev Kayal* | *Sr. CC Lead, SW-Eng & Test* | *14th July 2023* |

## REVISION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Reviewed By** | **Change Description** |
| 12th Jun 23 | 1.7 | Girish Pai | Sapna Ranganath | Revised based on Release 1.5 delivery plan |
| 14th Jul 23 | 2.0 | Girish Pai | Joydev Kayal / Sapna Ranganath | Incorporated review feedback |

# MANAGEMENT SUMMARY

As part of the PDF framework, Test strategy is a mandatory artefact for Stage Gate 1/2. It is required to ensure testing is done with a right approach and be able to deliver a good quality application. The test strategy is aligned to the testing requirements of the Project based on design and build phases.

The purpose of this document is to define a Testing Strategy irrespective of the methodology (Agile/Waterfall) opted in the Project. This document will define the standard deliverables and testing stages that are required for a development project and describe how these test stages will be used.

* Agile (SCRUM) is the way of working where test and development are integrated in the development lifecycle.
* One release contains n number of sprints as agreed with product owner
* Each sprint should deliver a working solution
* Continuous testing during the application development lifecycle
* Improved integration between End-Users and Development Team
* More Unit test and Back-end Test compared to GUI tests
* Iterative deployment and testing.
* More test automation and less manual testing

**Project Delivery Framework – Project activities wrt. key decision points.**

Project Delivery Framework Project Activities
Project Initiation and Planning
Final Investment Decision
Execution Phase 
Final Deploy Decision
Deploy and Operate Phase
Project Closeout Phase

WE ARE HERE

# INTRODUCTION

## ABOUT THE PROJECT

SLMT Orbital focuses on optimizing the SLMT portfolio. The Objective is to achieve a consistent view of the portfolio and facilitate an improved decision making supported by an integrated digital system. The Orbital solution will be a combination of several different, but closely coupled, modules which will be developed separately.

The following describes the key requirements of **Scheduling** module:

* Fit for purpose and dynamic scheduling tool for Short Term and Ops teams to manage portfolio physical scheduling for the next ~180 days.
* Automation and overhaul of current fleet-plan, avoiding two-step process and exchanges of emails between Optimisation Traders and Ops.
* Friendly user interface to visualise schedule, receive notifications, add events, perform simulations.
* Analytics suite including estimation of EDQ, ETA based on various voyage parameters.
* Direct access to schedule for Ops team and management.
* Data integration of commercial and operational data through a central data layer.
* Incremental changes in optimisation, built on 3D, with improvements in user interface, analytical components and more integrated data inputs and outputs.

## PROJECT CHARTER

[01\_PDA Project Charter 04\_2020.docx](https://eu001-sp.shell.com/:w:/r/sites/UGDSDSTSSLMTDataPlatform/Shared%20Documents/General/Supportability%20and%20Transition/_Archive/01_PDA%20Project%20Charter%2004_2020.docx?d=wd27bbff234624fe0a964b7f3705c835f&csf=1&web=1&e=QdblYf)

## ARCHITECTURE DIAGRAM / SAHLD (Solution Architecture High Level Design)

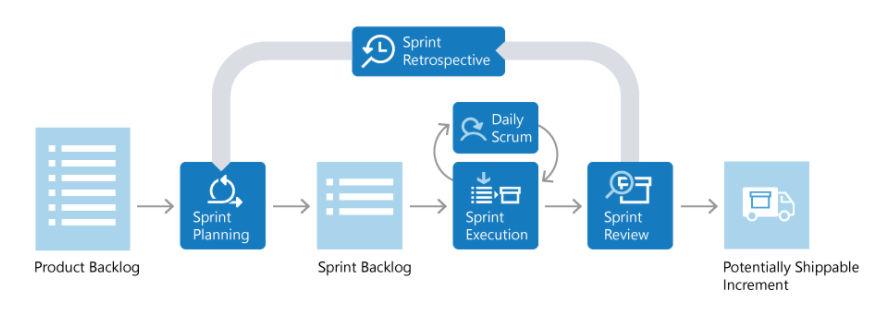
A picture containing timeline

Description automatically generated

# TESTING APPROACH

## OVERALL TESTING APPROACH

The project will follow the Agile methodology with Development and Testing run by IT Development & QA teams using the Agile values.



* Agile (SCRUM) is the way of working where Test and Development are integrated in the Development lifecycle
* One Release contains several Sprints as agreed with Product Owner
* Each Sprint should deliver a working solution
* Continuous testing during the application development lifecycle
* Improved integration between End-Users and Development Team
* Iterative deployment and testing.
* More test automation and less Manual testing.

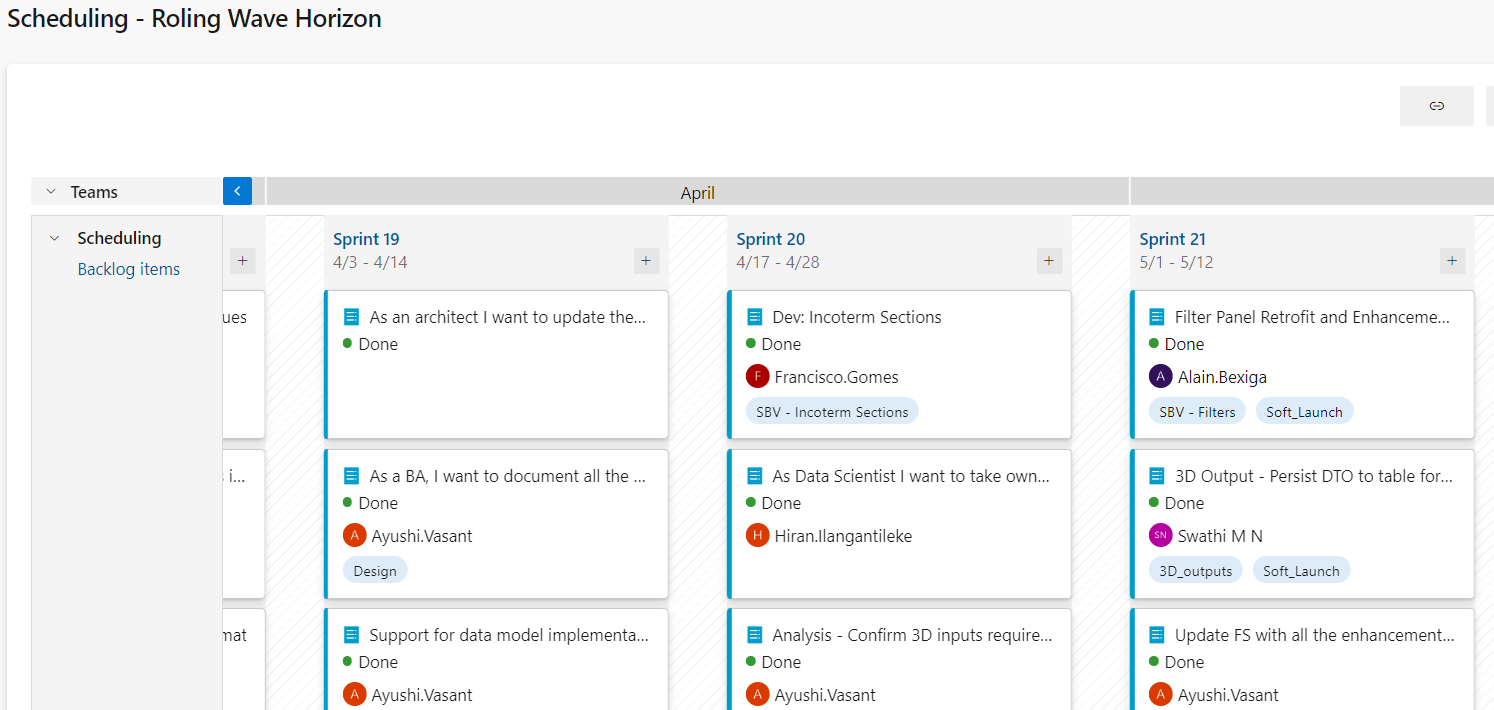
## SPRINT DETAILS

The Soft Launch Release will be delivered with Sprints typically lasting for 2 weeks. The business go-live date is targeted for Q3’2023.

**Scheduling Soft Launch roadmap:**

[Scheduling Product Roadmap of all planned Epics](https://sede-ds-adp.visualstudio.com/SLMT%20Orbital%20Project/_deliveryplans/plan/d1431882-69d9-4a79-9a50-b9cc8cfee079)

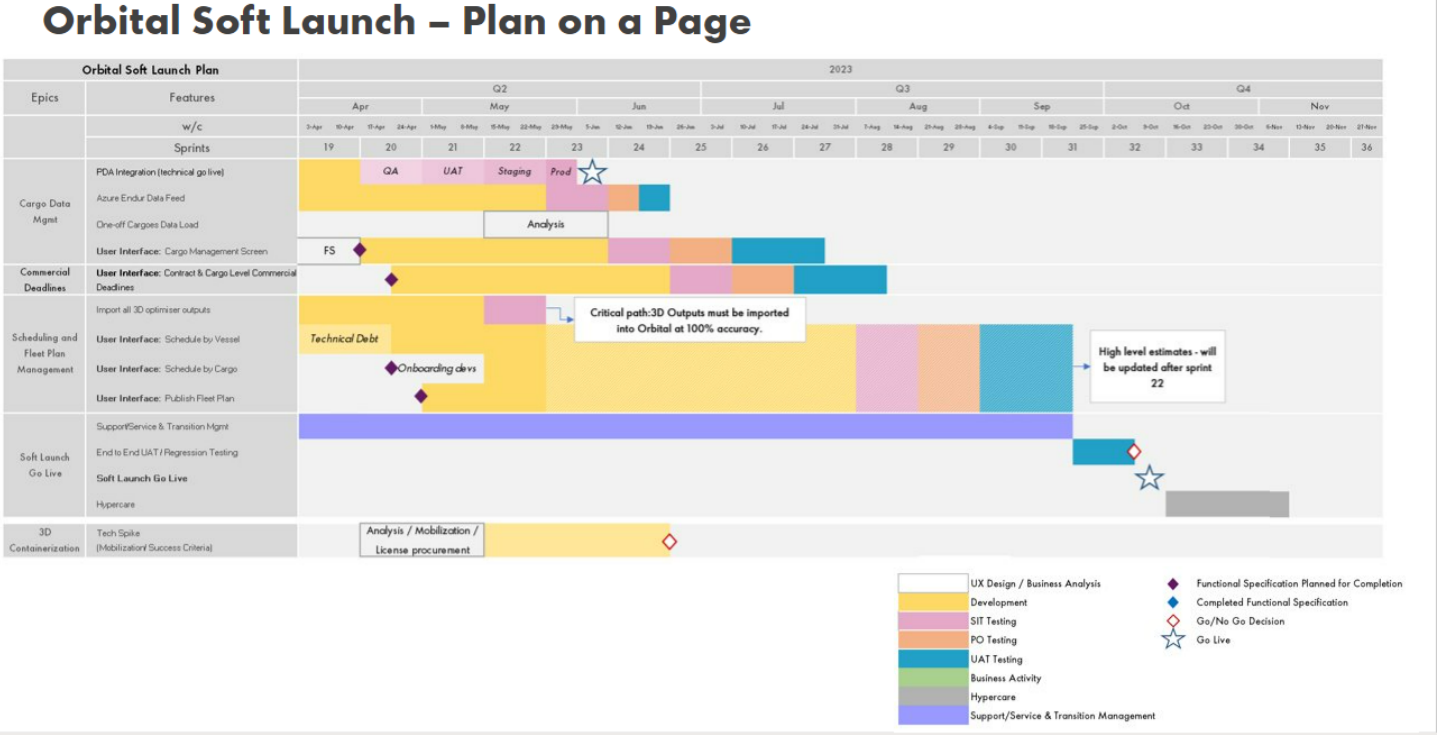
Snapshot of Scheduling Delivery Plan in Azure DevOps



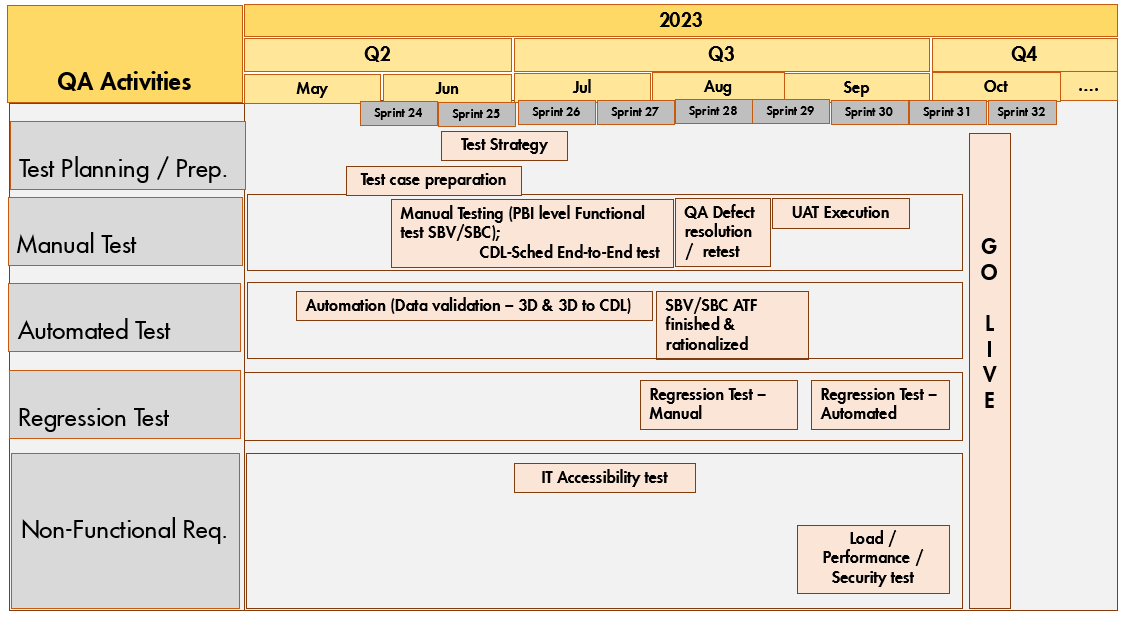
# PROJECT SCHEDULE

## PROJECT TIMELINE

[Orbital](https://www.figma.com/file/Tv3c8TKVrUBtaCAwboK3Ol/Soft-Launch_Refinement-Board?type=whiteboard&node-id=0-1&t=2IdaIVpy9IVcDMXR-0) Soft Launch – [Plan on a Page](https://www.figma.com/file/Tv3c8TKVrUBtaCAwboK3Ol/Soft-Launch_Refinement-Board?type=whiteboard&node-id=0-1&t=2IdaIVpy9IVcDMXR-0)



**Test Schedule**



# TESTING SCOPE

## WHAT IS TO BE TESTED? (IN SCOPE)

The scope of Scheduling module is to design the new Scheduling visualization screen for the Soft Launch Release. It incorporates all the changes and additions that have been made since the Product Owner’s approval on 7th November 2022.

The Orbital Application screen’s visualizations of 3D Optimizer in which the user can visualize the output in 3 separate ways. Each of the views will serve specific use case and user type:

* **Schedule by vessel (incl. Shorts, DES-DES & FOB-FOB as separate sections)**
* **Cargo overview**
* **Map overview**

**Document Reference:**

[Scheduling - Visualisation Schedule By Vessel and Cargo\_1.5.docx](https://eu001-sp.shell.com/:w:/r/sites/UGDSSLMTBCGOptimisationShellandGuestsTeam/Shared%20Documents/Scheduling/05%20Design/Functional%20Specifications/Scheduling%20Visualisation%20-%20Schedule%20by%20Vessel%20and%20Cargo/Scheduling%20-%20Visualisation%20Schedule%20By%20Vessel%20and%20Cargo_1.5.docx?d=wfbc93fd31d6e42efb6552de85a2c792b&csf=1&web=1&e=Z2rBgF)

(Functional Specification)

[Scheduling Epics Backlog - Boards (visualstudio.com)](https://sede-ds-adp.visualstudio.com/SLMT%20Orbital%20Project/_backlogs/backlog/Scheduling/Epics)

(Azure DevOps - Product Backlog Items)

Related documents

|  |  |
| --- | --- |
| **Document type** | **Filename & Link to location** |
| Data Mapping for Vessel Screen | [Detailed data mapping link](https://www.figma.com/file/pqfA3uW6Muz9XFbzwL6IDs/Shell-Orbital-Design?node-id=3835%3A208455&t=khGdQ3SUllYbBz6k-0) |
| Figma Designs | [Shell Orbital Design – Figma](https://www.figma.com/file/pqfA3uW6Muz9XFbzwL6IDs/Shell-Orbital-Design?node-id=1199-62107&t=Qt6na2MwofhRcl3b-0) |
| Change Log | [ChangeLog.xlsx](https://eu001-sp.shell.com/:x:/r/sites/UGDSSLMTBCGOptimisationShellandGuestsTeam/Shared%20Documents/Scheduling/05%20Design/Functional%20Specifications/ChangeLog.xlsx?d=wded1fd76becf474fa9514fda1d0cb6f1&csf=1&web=1&e=ZRVUPy) |
| 3D Tables and details | [3D to Orbital InterfaceV1.2 (002).docx](https://eu001-sp.shell.com/:w:/r/sites/UGDSSLMTBCGOptimisationShellandGuestsTeam/Shared%20Documents/General/Testing%20and%20QA/All%20CDM%20%26%20Scheduling%20Requirement%20Documents/3D%20to%20Orbital%20InterfaceV1.2%20(002).docx?d=w9ec2bed1aba844738c3972d14a439e2b&csf=1&web=1&e=0lPz1c) |

**Functional Requirement:**



**Non-Functional Requirement:**

|  |  |
| --- | --- |
| **NFR** | **Scope** |
| Load / Performance | A Load test to simulate higher volume of data or number of Users (20, 50, 100 & 200) simultaneously accessing the application and validate application behaviour and response time of UI visualizations.  (A Load test must be executed every time the Module undergoes a major Design change or significant increase in data volume or OT User base). |
| IT Accessibility | Application can be utilized by everyone irrespective of hearing, movement, sight or Cognitive disability. |
| Penetration / Security | Application must be secure from hacking or penetration by unauthorized/external entities. |

## WHAT IS NOT TO BE TESTED? (OUT OF SCOPE)

* Any Product Features defined in FS NOT having a corresponding PBI in ADO but delivered in a Sprint.
* Testing of 3D Application functionality.
* Testing of any 3rd Party Interfaces that do not impact Scheduling functionality.
* Explicit testing of Microservices API implementation. This will be covered with a test combination between GUI and corresponding data check in the database.
* Open Defects from Release 1.0 not included in Soft Launch scope.
* Any manual business processes.

# Risk Based Testing approach

Risk based testing uses risk to prioritize and emphasize the appropriate tests based on risk of failure, the function of their importance and likelihood or impact of failure during test execution.

The risk-based testing methodology includes planning, design, and execution of testing. Everything is carried out according to the priority. But the main focus areas within the application are:

* Functionality that is critical for the business.
* Features that are used often and prone to failures.

Analysis will be done to identify specific areas of the Application that qualify for a risk-based testing approach.

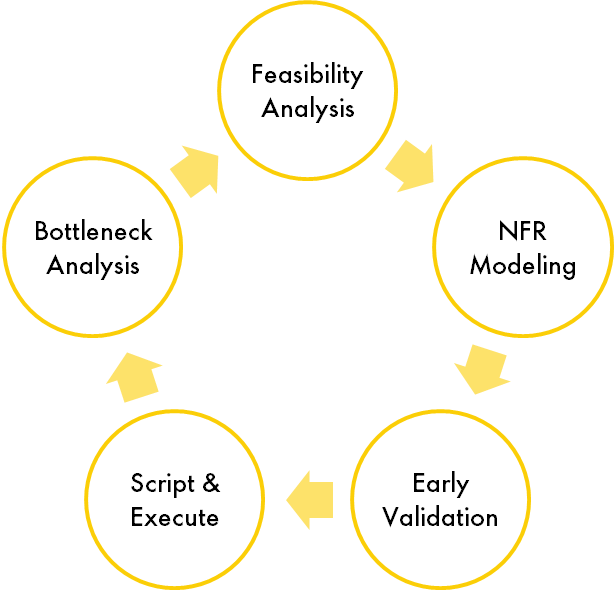
# TEST PHASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Phase** | **Objective** | **Entry Criteria**  (*Gives the prerequisite items that must be completed before testing can begin*) | **Exit Criteria**  (*Defines the items that must be completed before testing can be concluded*) | **Ownership**  (*Which team is responsible for this phase*) |
| Unit Testing | To ensure code is developed correctly | * Dev environment available. * Completed code as per the acceptance criteria * Completed test scripts | * Unit test scripts have been successfully executed and actual results documented in Azure DevOps (ADO). * Identified defects have been fixed and retested * Should not have any Critical/High bug open | Dev Team |
| Smoke Testing | To ensure the main functionality/ UI visualizations of the module work properly before planned testing begins | * Environment availability * Completed test scripts covering the main functionality/ UI visualizations | * Smoke Test scripts should be all passed and no blocking issues are open and actual results documented in ADO * Identified defects have been fixed and retested * Ideal situation, Smoke Test should not fail. | Testing Team |
| System Testing | To ensure all functionalities of SBV & SBC screens are working | * Completed Unit Test & Smoke Test * No Critical/ High bugs are open * Orbital & 3D Environments are available * Test Data (in a production like state) is consumed from 3D database * Completed test scripts reviewed with BA | * 100% of planned test scripts have been executed and 90% test scripts are Passed * Actual test results are documented in ADO * All Critical/High bugs are fixed, retested & closed * Open bugs have been Triaged for a resolution plan and aligned/signed off by Product Owner/ SME * Test Report is published | Testing Team |
| Integration (End-to-End) Testing | To ensure data flow between CDM and Scheduling modules are working | * Completed Unit & System Testing * Code changes migrated to Integration test environment * Test environment is configured, and Testers have necessary access * Completed test scripts reviewed by BA/DA   and Test Data available | * 100% of planned test scripts have been executed and 90% test scripts are Passed * All Critical/High bugs are fixed, retested & closed * Open bugs have been Triaged for a resolution plan and aligned/signed off by Product Owner/ SME * Test Report is published | Testing Team |
| Regression Testing | To ensure that existing functionality is working fine and new changes have not introduced any new bugs | * Completed test phases planned prior to regression testing * No Critical/High bugs are open from prior test phases * Manual Regression test pack is finalized / Automated Test script is ready | * All test scripts have been executed and actual results documented in ADO. * No Critical/High bugs are open * Open bugs have a proper workaround; or Triaged for a resolution plan and signed off by Product Owner/SME | Testing Team |
| User Acceptance Testing | To ensure Business User’s expectations are met | * Completed all test phases planned prior to UAT * No Critical/High bugs open and open bugs have a resolution plan published * Code migrated to the UAT environment * Test report from previous test phases agreed with Product Owner/SME * Required data available in test environment * UAT Test Plan aligned with Product Owner/SME | * 100% of planned test scripts have been executed and 90% test scripts are Passed * All Critical/High bugs are fixed, retested & closed * Open bugs have a proper workaround; or Triaged with a resolution plan and signed off by Product Owner/SME * Test Summary Report * UAT Signoff by Product Owner/SME | Business Users |
| Load / Performance & Penetration Test | To ensure that system performance satisfies the operational requirements of the business and secure from hacking by unauthorized/ external entities | * Completed system test and stable application * FE focal engaged to evaluate & finalize the scope agreed with project stakeholders * Test environment available * Test tools/scripts are ready | * Test scripts have been successfully executed and actual results documented * Identified defects have been either fixed and retested or registered in ADO and signed off by Product Owner/SME * Test Report * Defined benchmarking criteria should be achieved. | FE team |
| IT Accessibility | To ensure application can be utilized by everyone irrespective of hearing, movement, sight or Cognitive disability | * Code is migrated to test environment * Availability of tool details –Dyna Trace /Jaws etc. on Tester’s system | * Verify the UI implementation conforms to the Shell mandated WCAG guidelines * Identified deficiencies are fixed or aligned with Testing CC team for ITAS Support | Testing team / Testing CC team Support |

# PERFORMANCE TEST STRATEGY

A major component of the Technical Infrastructure test is the performance capability. Performance Testing ensures that system performance satisfies the operational (non-functional) requirements of the business.

**PT**



Load Testing Tools

APM Tools

The following factors were considered to arrive at the decision for Performance test,

|  |  |  |  |
| --- | --- | --- | --- |
| **SL #** | **Criteria** | **Evaluation** | **Remarks** |
| 1 | SaaS solution/implementation? | No | Application is designed & built Shell in-house |
| 2 | Any New Technology/ Migration/ Upgrades involved? | Yes | Introduced Cypress to implement some of the UI elements |
| 3 | Is Slow Performance in application observed? | No | Test results from previous Performance test (Feb’23) indicate response times to be stable and in general low, however, application has since undergone major UI enhancements. |
| 4 | Is number of concurrent users expected to be high in number? | Yes | Expected current User base is upto 20 Users, however, a User base in multiples of 50, 100 & 200 will be simulated to address any future contingency. |
| 5 | Any new interface is introduced? | No |  |
| 6 | Any new Infrastructure/ Datacentres included? | Yes | A new table in 3D Database is created to source the Scenario data to Orbital at regular frequency (~05 mins) |
| 7 | Any issues in Response time observed? | No | QA test phase in progress |
| 8 | Any load is expected than usual? (Ex: Peak periods) | No | This will be evaluated further with each Release |
| 9 | Is geographical location different than roll out and app hosting server? | Yes | OTs are based out of 3 geographical locations (UK, UAE & Singapore) |

**Performance Scope**

Based on the above factors and Design changes implemented for Soft Launch Release, the following areas will be targeted to test **load performance** of the Orbital Scheduling module to get an early indication of performance impact, if any.

|  |  |
| --- | --- |
| **Target Area** | **Validation** |
| SBV & SBC screen visualization rendering with data from Orbital database | * The test would be to run ‘n’ instances of scenarios simultaneously using a tool such as Load Runner. * Having a number of Users accessing the system simultaneously to prove multiple Users can access the system without slowness in the screen response time. |
| Data import from 3D into Orbital database | * To run this test on the selected 3D system, the status of ‘n’ scenarios in this table could be set to be ready to copy and the time to pull the data recorded. |

**Tools**

**Load Runner Enterprise** is the approved tool in Shell that will be used to execute the Load / Performance tests.

**Timeline**

Performance test is ideally run in a Production-like environment to derive the best measurement. For Soft Launch Release, it will be performed in UAT environment in alignment between FE focal, Project Manager and Dev Technical Lead.

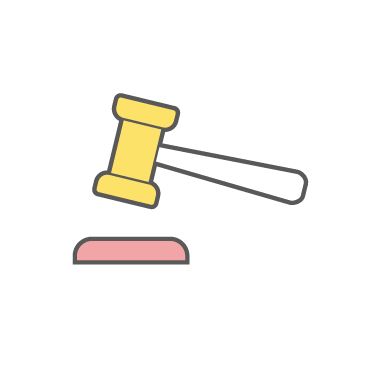
# IT ACCESSIBILITY TESTING

Accessibility means the ability of using all the websites, tools, techniques, hardware, and software products, by the people irrespective of their disabilities such as hearing, movement, sight, and cognitive ability.

Diversity & Inclusiveness

Growth in Business

Avoid Legal compliance, fines



ITAS

Benefits

As per PDF guidelines, IT Accessibility testing is mandatory for all new IT Projects delivered within Shell. If it is an Agile Project, then embed ITAS testing within the Sprint and test before releasing new feature.

|  |  |  |  |
| --- | --- | --- | --- |
| **SL #** | **Criteria** | **Evaluation** | **Remarks** |
| 1 | Business requirement for the application to be ITAS compliant? | NA | Since it’s now mandatory for all IT Projects in Shell to perform ITAS testing. |
| 2 | Does application have a UI/Front end? | Yes |  |
| 3 | At any given point of time, is this application to be used by people with special needs? | <*To be determined*> | Objective is to ensure the system is compliant for present & future. |
| 4 | Is SaaS solution/implementation? | No |  |
| 5 | Any increase in User base expected in future? | Yes |  |
| 6 | Is the application going to be external market facing? | No |  |

Tests will be performed on the application to check compliance against the following behaviour:

* Do all images have meaningful alternative text?
* Is every focusable element operable using the keyboard alone?
* Is there a consistent visible focus indicator when navigating using the keyboard?
* Have all controls, frames, and page titles been labelled meaningfully and uniquely?
* If custom controls/components have been implemented, have these been tested to assure that assistive technology recognizes name, role, state, and value (where applicable)?
* Have custom controls/components been verified to work as expected using assistive technology, such as a screen reader? (JAWS and NVDA)
* Are error messages interpretable by assistive technology?
* Do you have sufficient foreground and background colour contrast? (WCAG Colour contrast Checker)
* Have captions been included with any audio or audio/visual media?

Tools**:** AXE, Dragon / JAWS

Timeline: ITAS tests will be covered as part of QA testing phase.

# TEST DATA MANAGEMENT

This section describes approach to ensure the non-production environment for testing complies with IRM, Data compliance & GDPR policies.

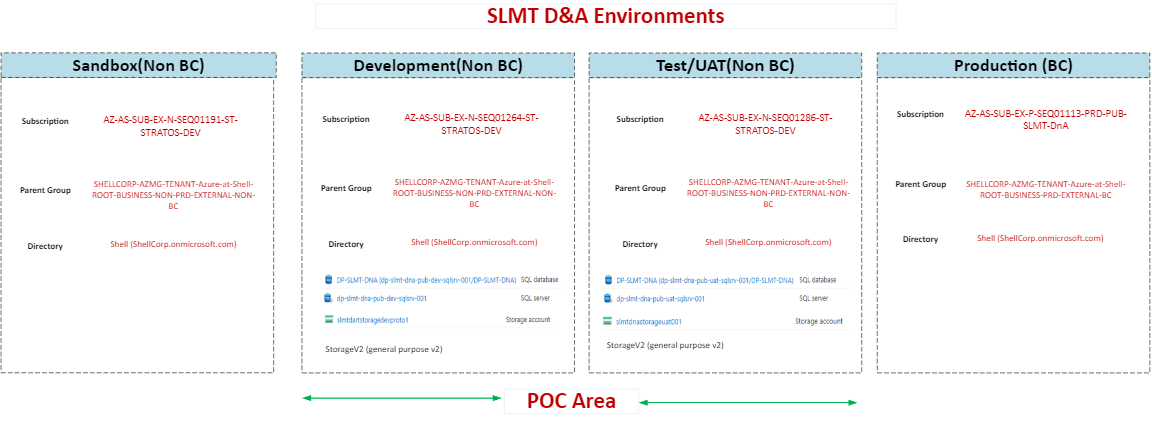
|  |  |  |  |
| --- | --- | --- | --- |
| **SL #** | **Criteria** | **Evaluation** | **Remarks** |
| 1 | Is the project using production data into lower landscape for testing? | Yes | Scheduling SBV/SBC visualizations are reliant on 3D Production data |
| 2 | Any personally identifiable sensitive information is being used? | Yes | Cargo Operator’s email & personal Ids are stored |

* Scheduling module is reliant on 3D Production data to produce the SBV and SBC visualizations. Hence, Production data is loaded across QA and UAT test environments, consumed from 3D Application database.
* This helps to compare the visualizations with the 3D Grid and 3D Vertical Schedule report to ensure the data is in sync. The required 3D Production data is exposed for consumption by Orbital CDL database and is refreshed at short intervals (~5 mins).

Necessary NPE (Non-Production Environment) controls related to **Audit Logging** and **User Access Management** (**UAM)** have been implemented in lower environments (QA, UAT) to ensure compliance and protect Production data.

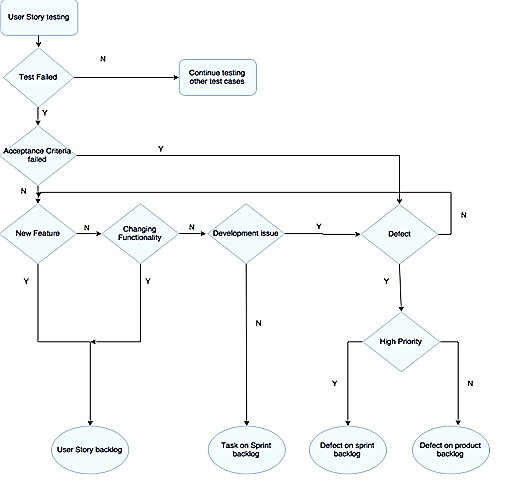
# TEST ENVIRONMENT

The following test environments will be used for development and testing of the application through the Sprints & Releases.



# DEFECT MANAGEMENT PROCESS

**AGILE**



In Agile whenever the defects are raised:

* They are added to the Product backlog for further discussion with the development team and PO
* Development Team and the Product Owner should determine together the priority of the Defect:
  + “Low Priority Defects” are discussed during the Backlog Refinement sessions and can be handled like any other User Story on the Backlog
  + “High Priority Defects” are the ones that need to be resolved as soon as possible. So, as soon as it becomes clear that a Defect has an effect on the Sprint content, the Development Team should re-discuss the Sprint content with the Product Owner

**Defect Severity Matrix & Resolution SLA**

|  |  |  |
| --- | --- | --- |
| **Severity** | **Application/Business Impact** | **Resolution SLA** |
| **Critical** | * Is considered a **Showstopper** that stops the User from moving forward * Blocking issue that will **fail to meet the** **Acceptance Criteria** if NOT fixed. * User cannot continue to test the functionality as related visualizations are impacted; no workaround is available. * The Defect blocks the Scenario from being tested.   Example: Functionality not delivered/ causing major system errors/ instability | With highest priority / ASAP |
| **High** | * Is considered a major issue but not a Showstopper * Does NOT impact the Acceptance Criteria * the Defect only impacts the current functionality being tested. * User cannot continue to execute the current test script; workaround is available to continue to test other test scripts / rest of the visualization * Performance related issue slowing down the progress of testing * Very important issue that MUST be fixed with priority   Example: Certain aspects of the functionality are not working | Must be fixed within the current Sprint timeline |
| **Medium** | * The defect has a minor impact on functionality. * User can continue to execute the current test script. * Issue will be fixed only when all Critical and High Defects are fixed.   Example: Cosmetic issues (Alignment of field names/Vessel Head, Mis-spelt label names, Incorrect fonts, etc) | Will be added to Product Backlog for discussion during Backlog Refinement session |
| **Low** | * The Defect does NOT impact functionality. Minimal / no impact to Acceptance Criteria or outside of original requirement scope   Example: Items not part of the approved Requirement/Design or a Change Request | Will be added to Product Backlog for discussion during Backlog Refinement session |

# TEST MONITORING / REPORTING

Test Monitoring is an important activity to ensure the progress of testing is being tracked and reported to the stakeholders. In this section we will describe the test metrics and how, and to whom, the report status and other periodic test reports are published for ongoing progress tracking.

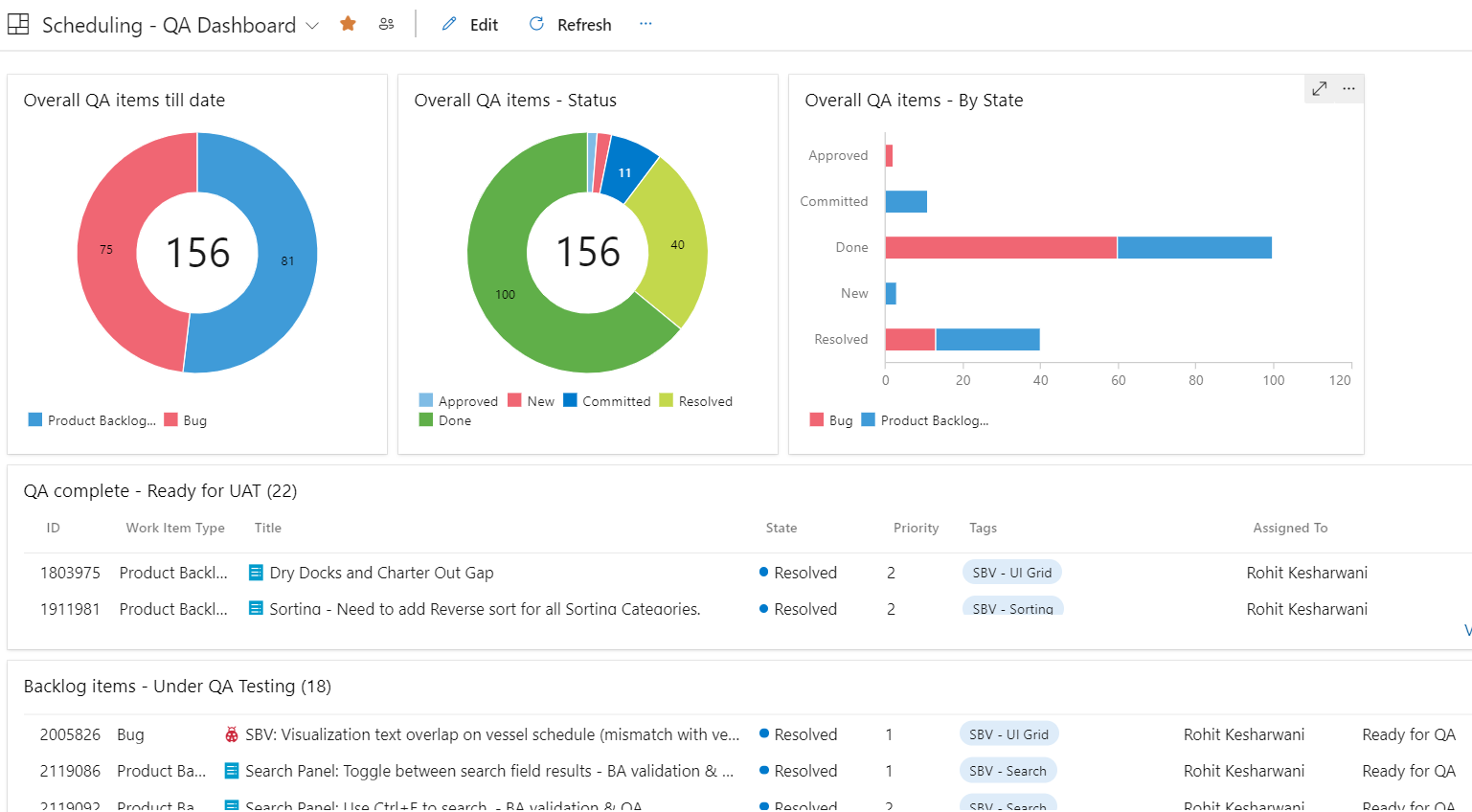
Below are the test metrics that will be reported:

|  |  |
| --- | --- |
| **Test Metrics** | **Objective** |
| Defect detected by module | To determine health of individual modules |
| Quality of User Stories | All user stories must have acceptance criteria, priority, and business value |
| Defect impact | Defect detected based on priority/severity |
| Defect root cause | **Defect type** and **Found During** values will be captured in ADO to analyse the root cause - viz. Design issues, Architecture issues, Coding issues |
| Rework %age | (Effort spent in bug fixing and re-testing)/Total effort spent |
| Defect Removal Efficiency | DRE = (Total number of Pre-UAT Defects) / (Total number of Pre-UAT Defects + Total number of UAT Defects) \* 100 |
| Performance metrics | Page load times, Response times, UI Latency, Throughput, Server Utilization, Availability% |

The below Agile ceremonies and reporting will be followed in the project,

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting / Report** | **Frequency** | **Stakeholder / Responsible** | **Agenda / Content** |
| **Daily Standup** | Daily | PM, Scrum Master, Business Analyst, Product Owner (as needed), Development team, QA team | Review tasks status of the team in ADO Backlog board for the day / Identify potential impediments affecting progress and actions to resolve offline. |
| **Sprint Planning / Refinement** | Monday before the start of the Sprint | PM, Scrum Master, Product Owner, BA, QA team, Development team | Review prioritized Backlog and finalize the scope for the next Sprint. Discuss team capacity, Clarify requirements with Product Owner. |
| **Sprint Review / Demo** | End of every Sprint | PM, Scrum Master, Product Owner, BA, QA team, Development team | Showcase the functionalities developed in the Sprint for assessment before migration to test environments. |
| **Defect Triage meeting** | Mon & Thu | PM, BA, Testing team, Development team  UAT team and Product Owner (UAT phase only) | Review Open defects identified in the Sprint and agree on resolution timeline as per SLA defined in Defect Severity matrix |
| **Test Execution Status report** | 3x weekly | IT Test Lead | [ADO dashboard](https://sede-ds-adp.visualstudio.com/SLMT%20Orbital%20Project/_dashboards/dashboard/0ffd4182-5533-464d-97d4-55d47907dfe1) is created and published to project stakeholders |
| **Test Summary Report** | End of QA phase | Project team and Product Owners | Published at the end of QA test phase with a summary of Test Execution, Defect metrics highlighting risks & issues to enable a Go/NoGo decision. |
| **UAT Kick-off meeting** | Before start of UAT phase in every Release | PM, Project BA, Change Manager, QA team, Product Owner, Business Tester | To walkthrough Business Tester on UAT Scope, Test execution and defect management process. |
| **UAT Progress review** meeting | Daily Standup | Business Tester, Product Owner, PM, Project BA, Dev Technical Lead, QA team | Review test progress, identify issues/blockers impacting the tests. |
| **UAT Signoff** | End of UAT phase | Business Tester, Product Owner/SME | After successful completion of the UAT, Business testers will Sign off the acceptance of the results. |

**Sample QA Test Execution Status Dashboard in ADO**



**Ways of Working (QA Proposal)**

* QA team should be involved in all project related discussions, workshops at any stage of the project to stay updated on project developments that would help better test planning.
* BA to ensure every functionality captured in the FS is converted into a Backlog item in ADO to avoid gaps in test coverage.
* Ensure a clear Acceptance Criteria is defined for each Backlog item which aligns to the expectations of Product Owner and can be compared with test results at the end of every test phase.
* BA to take ownership of managing the Backlog items in ADO. QA can support as needed.
* QA must be involved in any discussions related to functional requirement changes to help realign the test scope/plan and assess the impact, if any. Delayed communication to QA can have a direct impact on test coverage, effort and test schedule against overall Delivery plan.
* Proper Unit Testing is performed by Dev and test results uploaded to ADO before start of QA phase. This will help reduce the number of QA defects as well as enable QA to focus more on functional testing.
* Clear communication by Dev to QA with a list of PBIs that are being delivered to QA environment as opposed to deploying an uncertain count of PBIs which causes confusion. A Release note will help track against the planned backlog items and schedule.
* Dev publishes a list of open issues resulting out of coding challenges or Unit Test and a resolution plan to help QA plan their tests around them.
* Team updates the Discussion Board in ADO Bug form to always reflect the latest status of the bug. This will help reduce discussions, follow-ups and provide a status even during absence of the Bug owner.

# TEST TOOLS

The following test tools are used in the Project across the different test levels and for test lifecycle management,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Lifecycle Management** | **Functional Test Automation** | **API Test Automation** | **Performance Testing** | **IT Accessibility** |
| Azure DevOps  To manage Backlog items, Test Cases & Defects & Reporting | Selenium (Python)   * To automate Data validation from 3D to Orbital Database and Scheduling Regression Test scripts | * Postman * RESTAPI * To automate data flow validation between Orbital and Interfacing systems, where feasible (Endur, etc) | Load Runner Enterprise   * To simulate and execute Load/Performance tests | * AXE * Dragon / JAWS * To validate system’s compliance to support IT accessibility requirements |

# AUTOMATION SCOPE

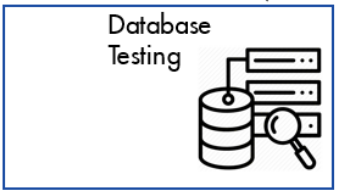
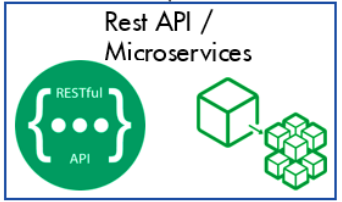
Testing will adopt a pragmatic approach for test automation and will be done manually as well as in an automated way with an objective to maximise automation and reduce manual testing with each Release.

The current focus areas will be where manual tests must be repeated each time the source code or data changes and is prone to errors such as,

* Data synchronization between 3D and Orbital databases with every refresh
* Overall stability of SBV/SBC screens every time new features are delivered

Strategy – We will leverage the standard CoE framework with Selenium/Python for UI validations and REST-API for Interface related tests in the Project. The test automation framework being built for Scheduling will be extended to CDM, Portfolio Optimization and remaining modules.

Automation Area

|  |  |  |
| --- | --- | --- |
| **Area** | **Test Scope** | **Automation Framework/Tool** |
| **Data Validation** | Data comparison between,   * 3D Shipping path Grid and 3D Database table (Orbital source) to ensure data quality * 3D and Orbital CDL databases to ensure data is in sync * 3D Vertical Schedule with SBV/SBC screens to ensure data quality | Testing CoE Framework with Selenium/ Python |
| **Web UI** | * Regression test suite covering critical Web UI test scripts for SBV/SBC screens. * The Automated Regression test script will be run with every major deployment to test environments (QA, UAT) | Testing CoE Framework with Selenium/ Python |
| **Interfaces (API)** | * Automate API testing between Orbital and Interfacing systems (Endur, Symphony, etc.), where feasible | Postman, REST API |

**Advantages**

* No limit on execution iterations.
* Regression pack will be updated at the end of every Release to include every major functionality that is delivered to Production.
* Will help to catch any issue that can break Production, in early stages.
* Can be a decision factor for a Go/No-Go to avoid impact to Production.
* Testing resources can always focus more on the new functionalities without having to spend effort on Regression testing for every defect fix or stability of existing functionalities.
* Can potentially reduce manual Regression testing in the long run for every Release.

# IMPORTANT ROLES IN THE PROJECT

Below are key roles and responsibilities of individual team member in the project delivery team.

|  |  |
| --- | --- |
| Roles | Responsibilities |
| Project Manager | * Create Project charter * Work closely with the business for the signoff * Liaise with Business Analyst Test Manager and Lead developer on progress and issues |
| Product Owner | * Accountable to the project sponsor for scope, priorities and maintaining product backlog |
| Business Analyst | * Act as Focal with the Product Owner * Participate in Discovery and Requirement gathering phases ​ * Manage requirements translated into PBIs and maintain in ADO​ * Participate in Defect Triage meetings to review defects * Support End Users during UAT phase |
| Scrum Master | * Ensuring smooth running of the sprints and removing impediments |
| IT Test Lead | * Prepare Test Strategy * Strategize and plan QA activities * Manage overall Defects closure​ * Provide overall test status through ADO Dashboard * Provide Go/No-Go decision for QA Test phase before handover to UAT * Prepare Test Closure Report |
| IT Tester | * Participate in requirement walkthrough with BA / Dev team * Prepare / Execute Test Cases and maintain test results in ADO * Provide status update of the progress of test execution * Report technical defects in ADO and track them to closure * Highlight risks with deliverables against Sprint timelines |
| IT Automation Tester | * Prepare Automation test scope * Prepare & execute automated test scripts * Report defects and track to closure |
| FE Testing team | * Evaluate Project need for Performance/Penetration tests and provide guidance * Prepare Performance Test Scope, estimation and test plan * Execute Performance/Penetration tests and publish test results, deficiency (if any) * Manage defects related to performance issues * Work closely with the Project manager for the performance test result signoff |
| Dev. Technical Lead | * Plan and execute development activities * Monitoring UT and providing confirmation to IT Test Lead once UT is completed​ * Ensuring testing completed as per timeline and status of testing correctly reflected in agreed tool​ * Ensuring team availability for testing and defect resolution​ * Ensure-unit test results are captured in the tool |

# RACI MATRIX

RACI matrix or linear responsibility chart, describing the participation by various roles in completing tasks or deliverables for the project.

A RACI Matrix is defined for each of the critical test phases and activities: The matrix will clarify roles and responsibilities for each stakeholder involved in that specific test phase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Process / Activity** | **Responsible\*** | **Accountable^** | **Consulted”** | **Informed~** |
| Requirement sync-up with 3 Amigos (BA, Test and Dev) and walkthrough | BA, PO | BA, PO | Dev, QA | Dev, QA |
| FS mapping to PBIs | BA | Scrum Master, Dev, QA | PO | BA, Dev, QA |
| Azure DevOps setup and maintenance | Scrum Master | BA, Dev, QA | BA, Dev, QA | BA, Dev, QA |
| PBI handover from Dev to QA | Dev | PM, Tech. Lead, Scrum Master | BA, QA | PO |
| Test Strategy design and walkthrough | Test Lead | Test Lead, PM | BA, Dev, CC Lead | BA, Dev, CC Lead |
| Setup & manage ADO folder structure | Test Lead | QA, Scrum Master, PM | PM, Dev, Scrum Master, QA, BA | Dev, PO, BA |
| Test Case review - Peer/Lead and with BAs | QA, BA | Test Lead, BA | BA, PM | Dev |
| Test Data Management | QA, Tech. Lead | Data Architect, PM | BA, PO | PO |
| Upload Unit Test results in ADO before handover to QA | Dev | Tech. Lead, Scrum Master | BA, PM | QA |
| Smoke Test | QA | Test Lead | Tech. Lead, Dev, PM | PM, Dev |
| System Integration Test (SIT) | QA | Test Lead | Tech. Lead, Dev, PM | PM, Dev, PO |
| Regression Test | QA | Test Lead | Tech. Lead, Dev, PM | PM, Dev, PO |
| Automated Test | QA | Test Lead | Dev, PM | Dev, PM, Testing CC Lead, PO |
| IT Accessibility (ITAS) | QA | Test Lead | Testing CC, BA, PO, PM | Dev, PM, Testing CC/ Lead, PO |
| Performance/Load Test | FE | PM, Test Lead, PM | Tech. Lead, BA, PO, PM | Dev, PM, PO, Testing CC Lead |
| Penetration Test | FE (IRM) | PM, Test Lead, S&C Analyst, IRM | BA, PM, IRM | Dev, BA, PO, Testing CC Lead |
| UAT Support wrt following,   * PBI demo & handover / highlight known issues * Define UAT Scope * Support UAT Testers with System usage * ADO / Defect Management * Obtain Signoff | BA, PO | PM | QA, UAT Tester, Tech. Lead | Scrum Master, Dev |
| UAT | UAT Tester, PO | PO, BA, PM | QA, Dev | Scrum Master |
| Defect Management | Test Lead, QA, Scrum Master, Dev | BA, PM, Tech. Lead | Dev, PO | Dev, QA, Testing CC (DRE) |
| Test Reporting in ADO | Test Lead | QA, PM, Scrum Master | BA, Dev | PO |
| SIT Signoff | Test Lead | QA, Dev, PM | BA, PO | Dev, QA |
| Sprint/Release Demo | Dev, Tech Lead | PM, Scrum Master, BA | PO | Dev, QA |
| Production Hypercare | SOM team | SOM Lead, PM, Tech. Lead | PO, Scrum Master | Dev, QA |

**\***R = RESPONSIBLE - The “doer” is the individual(s) who completes the task. The “doer” Is responsible for action/implementation. Responsibility can be shared. The degree of responsibility is determined by the individual with the “A”.

**^**A = ACCOUNTABLE - The accountable person is the individual who is ultimately answerable for the activity or decision. This includes “yes” or “no” authority and veto Power

**“**C = CONSULT - The consult role is individual(s) (typically subject matter experts) to be consulted prior to a final decision or action. This is a predetermined need for two-way communication. Input from the designated position is required

**~**I = INFORM - This is individual (s) who needs to be informed after a decision or action is taken. They may be required to take action as a result of the outcome. It is a one-way communication.

# ASSUMPTIONS, CONSTRAINTS, AND DEPENDENCIES

Assumptions can be of any aspect of the project like resourcing, test environment, requirements, etc.

Constraints are factors that inhibits the software testing process from achieving the requirements.

Dependencies represent a list of human and physical resources required, and predecessor activities, that must complete for your planned activities to succeed and deliver.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ASSUMPTIONS, CONSTRAINTS, DEPENDENCIES SUMMARY** | | | | | | | |
|  |  |  |  |  |  |  |  |
| **Assumptions** | |  | **Constraints** | |  | **Dependencies** | |
|  |  |  |  |  |  |  |  |
| **Total Assumptions** | |  | **Total Constraints** | |  | **Total Dependencies** | |
| 07 | |  | 02 | |  | 04 | |
|  |  |  |  |  |  |  |  |
| **Impact** | |  | **Impact** | |  | **Impact** | |
| 0 | Low |  | 0 | Low |  | 0 | Low |
| 03 | Medium |  | 0 | Medium |  | 02 | Medium |
| 04 | High |  | 02 | High |  | 02 | High |
| 0 | Critical |  | 0 | Critical |  | 0 | Critical |

|  |  |
| --- | --- |
| Assumption | Impact |
| Test Strategy is defined based on the project documentation in SharePoint, ADO and inputs obtained from relevant Project Stakeholders | Any project related discussions, changes not documented or available to QA team would lead to gaps in defining the Test Strategy |
| QA Test Coverage will be based on the Backlog items created in ADO and delivered with Sprints. Any features defined in FS and not captured as a PBI for delivery will be considered not in QA test scope. | Product features delivered without a Backlog item will be missed in QA test coverage and passed on to UAT and Production, untested. |
| All relevant documents (Architecture documents, Functional requirements, technical requirements, Data flow documents) are signed off by relevant stakeholders before the Development phase begins. Any additional change will go through project change request process. | Rework on test cases and additional testing effort would be required. |
| All signed off documents are available in Project SharePoint folders. Any review comments or changes in document will be informed/reviewed with QA team. | Gap in test scope to the extent of the changes leading to incorrect test case design and validation of the functionality. |
| Technical Design is complete and available to the testing team. | Design changes will impact the test design causing rework and additional effort. |
| Necessary access is given to Application and Test environments and full access to required Database for test user ids. All user ids and passwords related to roles, privileges and authentication are created and available to Testers. | Delay in start of Test execution and test completion against the Sprint timelines. |
| Code is installed in the test environment in a timely manner and the environment setup is as expected without issues | Application may not work as expected resulting in an unstable test environment to begin QA testing. |

|  |  |
| --- | --- |
| Constraint | Impact |
| Absence of Unit testing in Dev, Test results and summary of open issues | QA testing bandwidth consumed in finding Unit test bugs thereby reducing time & focus on functional testing |
| QA exclusion from project workshops | No access to important project related developments that can impact test planning |

|  |  |
| --- | --- |
| Dependency | Impact |
| On time delivery of unit tested and stable code in test environment. | Delay in test execution start having a knock-on effect on test closure against planned Test Schedule |
| Defect resolution as per Resolution SLA defined in [Defect Matrix](#_Defect_Severity_Matrix) | Delay in retest and closure of defects having a knock-on effect on test closure against planned Test Schedule |
| Data quality in test environment | Bad quality of data can impact the visualizations which can lead to incorrect testing or a greater number of defects |
| Having testing resources onboard in time to support test execution | Delays in onboarding resources impacts test completion timelines and additional load on existing team members |

# RISKS

| Risk Description | Risk Identified Date | Risk Age (Identified date – Current date) | Risk Impact (H/M/L) | Risk Probability (H/M/L) | Risk Rating(H/M/L) | Mitigation Strategy | Contingency Plan | Risk Owner (Role in the Project) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Changing Requirements | 17th Jul’23 | 0 | H | M | H | Regular connect with BAs to understand requirement scope/changes | Walkthrough test coverage with BAs to identify impact area and rework | BA |
| Lack of communication | 17th Jul’23 | 0 | M | M | M | Regular connects with Dev team to review Sprint deliverables | Effective use of Daily Stand-up, Planning and Retrospective meetings | PM |

# GLOSSARY

A Glossary of short terms used in this document with their full forms.

|  |  |
| --- | --- |
| **Short Form** | **Full Name** |
| UT | Unit Test |
| ST | System Test |
| E2E | End to End test |
| UAT | User Acceptance Test |
| PT | Penetration Test |
| ITAS | IT Accessibility Test |
| WCAG | Web Content Accessibility Guidelines |
| Dev. | Development team |
| QA | Quality Assurance (Testing team) |
| PO | Product Owner |
| BA | Business Analyst |
| PM | Project Manager |
| S&C | Security & Control |
| FE | Functional Excellence team (Testing Centre of Excellence) |
| Testing CC | Testing Capability Centre |
| IRM | Information Risk Management team |
| SBV | Schedule by Vessel screen in Scheduling module |
| SBC | Schedule by Cargo screen in Scheduling module |