Table of contents

1. Purpose 2
2. Scope 2
3. References 2
4. Definitions 2
5. Roles and responsibilities 5
6. Process flow 7
7. Procedure 11
8. Appendices 17
9. Revision history 29

1. Purpose

1.1. The purpose of this Standard Operating Procedure (SOP) is to guide software testing at Stryker, ensuring consistent quality and repeatability across all software projects. This SOP outlines the testing requirements to ensure outcomes meet or exceed business expectations.

2. Scope

2.1. This SOP applies to all software testing managed by the Global IT organization and covers common elements in all project implementations.

2.2. It applies to all Stryker IT staff and third-party employees involved in software testing.

2.3. The document outlines key testing activities that are managed and tracked during IT projects, including:

2.3.1. QA Standard Operating Procedures

2.3.2. Procedures covering:

2.3.2.1. QA processes

2.3.2.2. Roles

2.3.2.3. Responsibilities

2.3.2.4. QA Best Practices

2.4. For any inquiries related to testing or testing processes, please feel free to reach out to us via email at MAIL US .

3. References

3.1. Internal references

- ISSOP_SDM_001.T02, User Requirements
- ISSOP_SDM_001.T03, Functional Specifications
- ISSOP_SDM_001.T04, Traceability Matrix
- ISSOP_SDM_001.T05, Testing Strategy
- ISSOP_ SDM_001.T08, Deviation Record
- ISSOP_ SDM_001.T09, Testing Summary
- ISWKI_SDM_001, Software Testing

4. Definitions

4.1. Locally defined terms

4.1.1. Formal Cycle: Test cycle to test System Readiness for functional / E2E Business Process and Interfaces as defined by the testing team / business. Requires evidence of testing utilizing screenshots

4.1.2. FUT: In Functional Unit Testing phase, individualindividual units or components of a software application are tested to ensure they function correctly according to the specified requirements. A "unit" refers to the smallest testable part of the software, like a function, method, procedure, or module

4.1.3. Performance Testing - This is a is a type of software testing that evaluates how a system performs in terms of speed, responsiveness, stability, and scalability under a specific workload. The goal is to identify performance bottlenecks and ensure that the application can handle expected user loads while maintaining optimal performance. It includes various subtypes like load testing, stress testing, and endurance testing

4.1.4. QAS – Quality Assurance Services is a team that supports testing activities across IT organization as part of Service Delivery Group. You can mail the team at MAIL

4.1.4.1.

4.1.5. RCA : Root Cause Analysis

4.1.5.1. Requirement Traceability: This is process of tracking and linking requirements throughout the project lifecycle to ensure they are fulfilled. It involves mapping requirements to their corresponding design, development, test cases, and defects, allowing teams to verify that each requirement is covered and validated. This ensures that the final product meets the business needs, and any changes are accurately reflected across all

4.1.6. SISI : System Integrator

4.1.7. SIT: System Integration Testing: The primary goal of SIT (System Integration Testing) is to verify how integrated components or systems interact and to identify issues with data flow, interface handling, and overall integration. This is conducted in Type I and Type III Program

4.1.8. SIT 1 and SIT 2: There are two SIT cycles. SIT 1 is usually conducted by SI Partner testing team. SIT 2 is conducted by Stryker team with Support from Business. Purpose is to make sure that E2E business process flows are working correctly, and users will have smooth UAT

4.1.9. Sprint Testing: This is black box testing of features developed in the current sprint. This test cycle ensures integration between new and existing functionality & identify and fix bugs early

4.1.10. UAT: User Acceptance Testing is the final phase of software testing, where end users test the software in a real-world environment to ensure it meets their business requirements and is ready for production

5. Roles and responsibilities

5.1. Test Manager / Test Architect:

5.1.1. Review requirements and provide input during design workshops.

5.1.2. Resolve test team queries and prepare business test scenarios.

5.1.3. Ensure comprehensive test coverage and support test data creation.

5.1.4. Review test cases and confirm end-to-end business flow and configuration testing.

5.1.5. Assist the Automation Team with functional knowledge.

5.1.6. Analyze defects, support quick resolutions, and conduct exploratory testing.

5.1.7. Align all testing tracks and cover integration testing.

5.1.8. Perform impact analysis for changes and guide retesting efforts.

5.1.9. Identify regression candidates and analyze impacts of changes.

5.1.10. Support UAT defect analysis.

5.2. Automation Test Architect:

5.2.1. Assess existing automation framework, find gaps if any and provide recommendations

5.2.2. Implement recommended and agreed updates on existing automation framework

5.2.3. Do feasibility check, tool assessments for automation

5.2.4. Define the best practices and quality standards for automation/performance scripts

5.2.5. Build Continuous integration pipelines

5.2.6. Provide the automation feasibility analysis for the agreed applications

5.2.7. Test Advisory and consulting

5.3. Test Lead (Automation Testing, Performance Testing):

5.3.1. Implement automation framework changes based on project requirements.

5.3.2. Collaborate with the manual test lead to define automation scope and assess regression impact.

5.3.3. Manage the automation team, including estimation, work allocation, and planning.

5.3.4. Participate in sprint planning.

5.3.5. Develop reusable functions, data structures, reports, and execution schedules.

5.3.6. Review scripts to ensure they meet best practices and quality standards.

5.3.7. Share daily and weekly status reports with stakeholders.

5.3.8. Oversee the execution of the automation test suite as per the test plan.

5.3.9. Demo user stories to the business.

5.3.10. Report automation savings and assist with project closure activities.

5.3.11. Implement and maintain CI/CD pipeline in Azure DevOps (ADO).

5.4. Test Analyst (Automation Testing) :

5.4.1. Develop and maintain automation scripts

5.4.2. Execute the automation scripts as per plan

5.4.3. Failure analysis and rerun

5.4.4. Report automation results, ROI and other automation metrics to Tower leads

5.5. Test Analyst (Performance Testing):

5.5.1. Develop and execute Performance testing, load testing, stress testing on Jmeter / load Runner

5.5.2. Analyze performance test results

5.5.3. Prepare detailed Performance test summary Report

5.6. Test Analyst (Manual testing):

5.6.1. Understand user stories and requirements and report any issues.

5.6.2. Create test cases with guidance from the Test Lead.

5.6.3. Ensure test cases meet requirements, review with the BA, and log in ADO.

5.6.4. Identify test data, execute tests, and update the status in ADO.

5.6.5. Capture test results with screenshots.

5.6.6. Log defects, retest after fixes, and update status.

5.6.7. Give daily updates to the Test Lead

6. Process flow

6.1. Project Types and recommended Testing Cycles

Following tables summarizes different types of software delivery projects and their methodologies commonly adopted by Stryker IT. The tables below list down recommended testing types. These tables are a general guide. The actual testing cycle should be tailored to the specific needs and risks associated with the project and the criticality of the system being developed or maintained

6.1.1. TYPE I Projects

Project Type	Methodology	Testing Cycle	Description
TYPE I **Existing Integrations to be used**	Waterfall	Unit Testing + Single Testing Phase of SIT + Automated Regression Testing + UAT	Testing is performed after the development phase is complete, before deployment.
	Agile	Continuous Testing (per Sprint) + Automated Regression Testing + UAT	Testing is integrated into each sprint, with frequent iterations and feedback.
	Hybrid	Sprint Testing + one phase of SIT + Automated Regression Testing + UAT	A combination of single-phase testing and iterative testing during certain milestones.

6.1.2. TYPE II Projects

Project Type	Methodology	Testing Cycle	Description
TYPE II **Complex Enterprise System with modification of existing integrations and/or brand-new Integrations**	Waterfall	Multiple Testing Phases 1. Unit 2. 2 cycles of SIT with one done by SI partner and second done by Stryker testing team 3. UAT 4. Regression Testing planned with automation	Sequential testing phases, with each type of testing completed before moving to the next phase.
	Agile	Continuous Integration Testing (CI) + Regression Testing + UAT	Automated testing with each integration, along with regression testing to ensure stability.

	Hybrid	Multiple Testing Phases **1.** Unit Testing **2.** 2 cycles of SIT with one done by SI partner and second done by Stryker testing team **3.** UAT **4.** Regression Testing planned with automation	Initial comprehensive testing followed by iterative testing with continuous feedback loops.

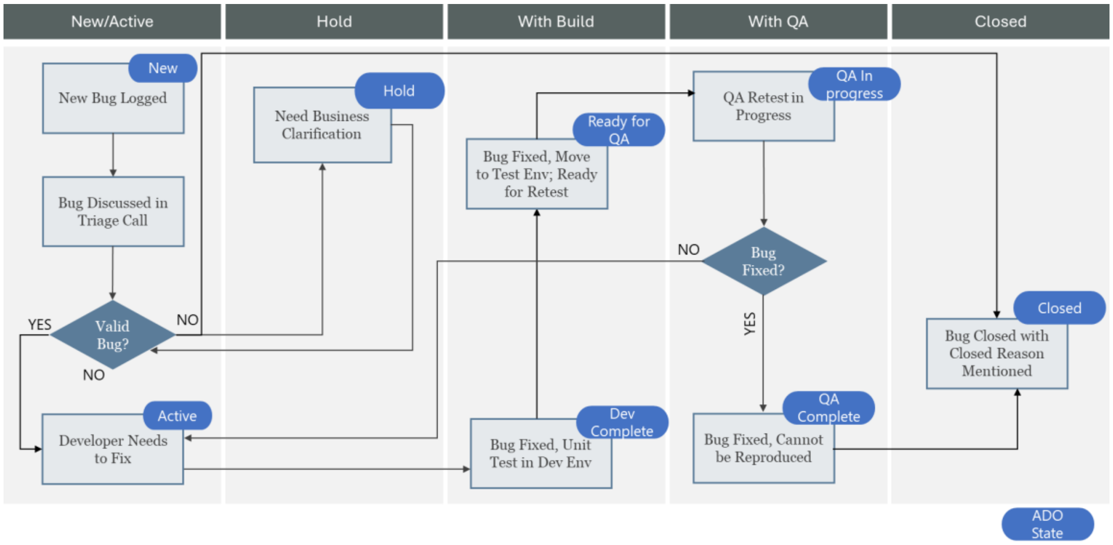
6.1.3. TYPE III Projects

Project Type	Methodology	Testing Cycle	Description
TYPE III **Legacy System Migration or Upgrades**	Waterfall	Sequential Testing (Data Migration, Integration, System) + Automated Regression Testing	Testing phases aligned with migration steps: data, integration, and system testing.
	Agile	Incremental Testing (Data Validation, Integration, Automated Regression Testing)	Continuous testing of migrated data, with ongoing integration and regression tests.
	Hybrid	Early Data Validation + Incremental Integration Testing + Automated Regression Testing	Initial data validation followed by phased integration testing with ongoing feedback.

6.1.4. .

6.2. Defect Management Process

6.2.1. Bug Lifecycle - All Bugs for all testing cycles must be recorded in ADO.



6.2.2. ADO Bug State Definition

New	The bug has been logged and is awaiting triage.
Active	The bug has been reviewed and is currently being worked on by the development team.
Dev Complete	Fix is completed but not available in test environment.
Ready for QA	Fix is completed and available in test environment.
QA in Progress	Retesting is in progress.
QA Complete	Retesting is completed and bug is correctly fixed.
In Validation Approval (Only Validated test cycle)	Bug is awaiting validation team's approval.
Closed	Validation team approved the bug.
Hold	Bug fixing is on hold

6.2.3. Bug Severity Definitions

Severity Level	Definition
1 - Critical	• An issue that affects a central requirement (key functionality/feature) for which there is no workaround. • Testing of affected functionality cannot continue without fixing. • Testing in general cannot continue without fixing.
2 - High	• An issue that affects a central requirement (key functionality/feature) and significantly impacts business ability to use application. • Testing of affected functionality can continue using available workaround.

3 - Medium	• An issue that affects a non-central requirement (key functionality/feature), but a workaround exists Testing of affected functionality can continue using available workaround
4 - Low	• An issue that affects a non-central requirement (key functionality/feature), but a workaround exists like cosmetic items, Spelling mistake etc. • If the issue is related to cosmetic changes or script changes, issue severity will get defined as “Low”

6.2.4. Root Cause Analysis

6.2.4.1. Before closing every bug, the RCA needs to be done and appropriate root cause should be selected in ADO. Below are the guidelines for RCA.

Root Cause	**Definition**
Cannot Reproduce	Defects cannot be reproduced.
Code issues	Valid Bug. Issue exists due to coding errors.
Configuration Setup Issue	Valid Bug. Config needs to change by the tester in the config module.
Data issues	Issues exist due to incorrect data or data inconsistency.
Deployment Issue	Issue related to deployment failures or errors in the deployment process.
Duplicate issue	This issue already exists. Tag existing defect number with the same functionality.
Enhancement	Not a bug. Triage done and agreed with BA as a new enhancement.
Environment Issue	Issue caused by the environment setup, including configuration, connectivity, or other environmental factors.
Existing Functionality (not a defect)	This is an existing functionality; the way the application works and is not a bug.
Incomplete Requirement	Valid Bug. Requirement needs a change.
Performance Issue	Issues related to the application's performance, such as slow response times or system crashes under load.
User Setup	Need to have a new user setup from the configuration side.
Working as Designed	Not a defect. This is as per the design and current functionality.

7. Procedure

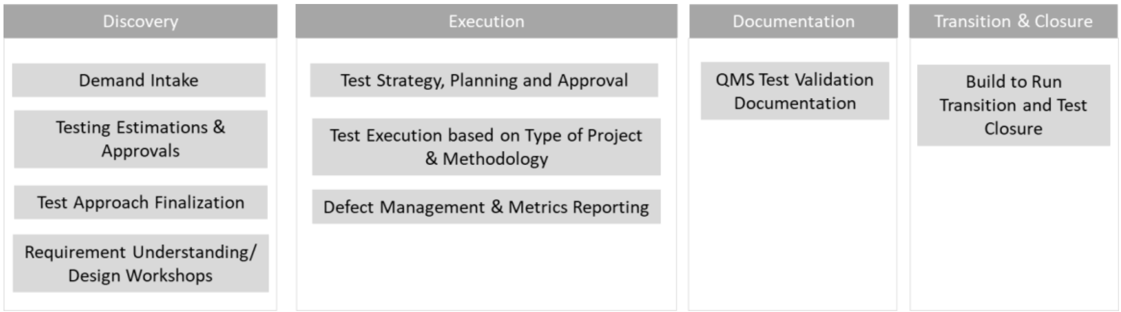
7.1. Quality Assurance Team Engagement

7.1.1. The testing team should be involved from the initiation phase, including the requirement gathering process. This ensures they fully understand the project requirements and can quickly resolve any uncertainties or ambiguities.

7.1.2. The testing team should be involved from the initiation phase, including the requirement gathering process. This ensures they fully understand the project requirements and can quickly resolve any uncertainties or ambiguities.

7.1.3. Quality Assurance Services Engagement Process is outlined as below

Demand Intake FormDemand Intake



Discovery	Execution	Documentation	Transition & Closure
Demand Intake Testing Estimations & Approvals Test Approach Finalization Requirement Understanding/ Design Workshops	Test Strategy, Planning and Approval Test Execution based on Type of Project & Methodology Defect Management & Metrics Reporting	QMS Test Validation Documentation	Build to Run Transition and Test Closure

7.2. Test Environment

7.2.1. All Testing cycles except for Unit Testing need to be performed in TEST environment. This TEST environment should be separate from Dev environment and should have test data aligned with integrating applications, if any.

7.3. Types of Testing – RACI, Scope, KPI and Metrics

Details	**RACI**	**Scope**	**KPI**	**Metrics**
Unit Testing	- **Responsible:** Developers - **Accountable:** Development Team Lead - **Consulted:** Test Engineers, Architects - **Informed:** PM, QA Lead	• Testing individual components or functions • Ensure unit functions correctly in isolation • Cover all possible paths and edge cases	• Code coverage achieved • Unit tests executed vs. planned • Defect density • Average time to fix unit-level defects	• Pass/Fail rate • Defect removal efficiency • Test execution time • Defects per KLOC
Sprint Testing	**Responsible:** QA Engineers, Developers	• Testing of features	• Number of stories	• Velocity (stories completed/sprint)

	- **Accountable:** Scrum Master, Product Owner - **Consulted:** Business Analysts, Architects - **Informed:** Project Stakeholders	developed in the current sprint • Ensure integration between new and existing functionality • Identify and fix bugs early	completed vs. planned • Test case execution rate • Defects found per sprint • Cycle time (development to testing)	• Pass/Fail rate for sprint test cases • Test automation coverage
System Integration Testing (SIT)	**Responsible:** QA Engineers, Integration Engineers - **Accountable:** QA Lead - **Consulted:** System Architects, Developers - **Informed:** Project Manager, Business Owners	• Verifying interactions between integrated systems and components • Ensuring end-to-end functionality across systems	• Requirement Coverage • Number of defects found in integrations • Defect severity level • Defect Density • Defect fix rate	• Pass/Fail rate for integration test cases • Defects per integration • Data integrity and accuracy • Test execution time per cycle • Requirement Coverage
Regression Testing	**Responsible:** QA Engineers - **Accountable:** QA Lead - **Consulted:** Developers, Business Analysts - **Informed:** Project Manager, Stakeholders	• Validating that recent code changes have not adversely affected existing functionality • Re-testing previously tested modules and features	• Percentage of regression tests executed • Defects found in regression • Time taken for regression cycle • Regression test coverage	• Regression test execution rate • Defect discovery and fix rate • Pass/Fail rate • Regression test cycle time • Requirement Coverage • Automation coverage
User Acceptance Testing (UAT)	**Responsible:** End Users, Business Analysts	• Validating that the system meets business requirements	• UAT test completion rate	• Defect discovery rate • Pass/Fail rate for business cases

	- **Accountable:** Product Owner, Project Manager - **Consulted:** QA Lead, Development Team - **Informed:** Stakeholders	• Ensuring that the system is ready for production use by end users	• Number of defects found in UAT • Time to resolve UAT issues • User satisfaction with testing process	• Defect severity and impact • UAT test execution time • User satisfaction rate

7.4. Test Approach and Entry / Exit Criteria

Type of Testing	Test Approach	Entry Criteria	Exit Criteria
Unit Testing	• White-box testing - Automated tests (JUnit, NUnit, etc.) • Test with various input scenarios • CI tools to automate unit tests	• Code completed for the unit • Development environment set up • Unit test cases prepared and reviewed • Code passes static analysis	• All unit tests passed • Code coverage meets defined threshold • No critical or high-severity bugs remain • Test results documented
Sprint Testing	• Agile testing approach • Test within each sprint • Focus on functionality required by user story, needing feature acceptance • Automate where possible	• User stories or sprint backlog items are fully developed • Development environment is stable • Test cases for sprint features are prepared • Acceptance criteria is defined	• All sprint test cases passed • No critical or high-severity bugs remain • Test results reviewed in sprint retrospective
System Integration Testing (SIT)	• Black-box testing approach • Validate data flow, interfaces, and communications between systems • Test error handling and recovery • End-to-end test coverage	• All integrated systems are available for testing • Data flow between systems is established • Integration test cases are prepared • Development environment is stable	• All integration test cases passed • No critical or high-severity bugs remain • Successful data exchange between all systems • Test results documented
Regression Testing	• Automated regression testing where possible • Focus on previously tested areas • Identify test cases most affected by recent changes	• Recent code changes are complete • No open critical defects from prior testing • Regression test suite is prepared	• All regression tests have passed • No critical or high-severity defects • Regression test report shared

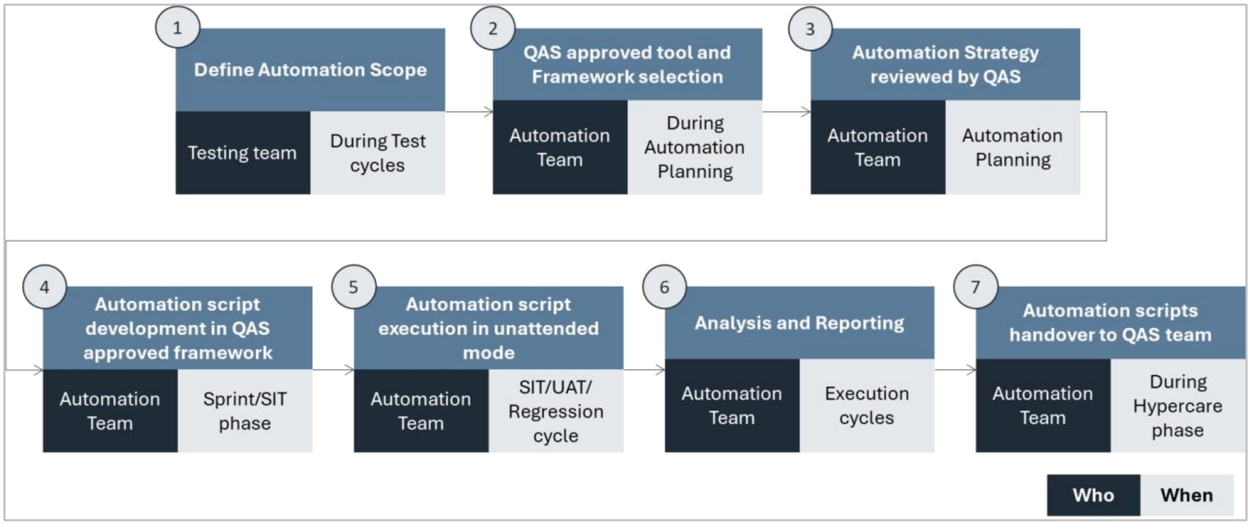
	• Continuous execution of regression suite	• Stable build is available for testing	• No impact on existing functionality identified
User Acceptance Testing (UAT)	• Business-driven testing approach • Focus on validating real-world business scenarios • Involve actual end users • Manual testing of key business processes	• Functional, system, and integration testing completed • No critical defects outstanding • UAT environment ready • Test scenarios and business cases are prepared and approved	• All critical business processes are tested and passed • UAT sign-off by users/stakeholders • No critical defects in the UAT phase • UAT results documented

7.5. Transition to Run/ Service Delivery Process

KT47	Test to Test Transition Checklist is completed and shared
KT48	Handover of Manual and Automation Test Scripts completed
KT49	Handover of Performance Test Scripts completed
KT50	Key Performance Indicators are within threshold limits

7.6. Test Automation

7.6.1. Test Automation Process - The process for test automation involves a series of systematic steps to ensure effective and efficient testing.


1
Define Automation Scope
Testing team
During Test cycles
2
QAS approved tool and Framework selection
Automation Team
During Automation Planning
3
Automation Strategy reviewed by QAS
Automation Team
Automation Planning
4
Automation script development in QAS approved framework
Automation Team
Sprint/SIT phase
5
Automation script execution in unattended mode
Automation Team
SIT/UAT/ Regression cycle
6
Analysis and Reporting
Automation Team
Execution cycles
7
Automation scripts handover to QAS team
Automation Team
During Hypercare phase
Who
When

7.6.2. Test Automation Frameworks

7.6.2.1. Tricentis Tosca Automation Framework: This framework is based on the licensed Tricentis Tosca tool and is utilized for automating SAP (both GUI and Fiori) and desktop-based applications. The Tosca framework leverages Reusable Test Blocks (RTBs), enabling the creation of reusable and generic test cases.

7.6.2.2. SAGE (Selenium Automation for Generalized Execution): Built using Selenium, this framework is designed for automating web-based applications. It offers a versatile testing solution that caters to both end-to-end and standalone test cases.

7.6.2.3. API Automation Framework: This framework, developed with Java and Rest Client, is used for automating REST and SOAP APIs.

7.6.2.4. Mobile Automation Framework: Built with Appium and Java, this framework automates iOS and Android mobile applications on simulators.

7.6.2.5. File Validation Framework: Developed using Java, this framework is designed for automating file and data validation. It is primarily used for migration and bulk load file validations.

8. Appendices

8.1. Appendix A, Testing Tools

8.2. Appendix B, Testing Best Practices

8.3. Appendix C, Test Documentation for QMS Processes

8.4. Appendix D - Test Metrics

8.5. Appendix E, Testing Workflows

8.6. Appendix F, Demand Intake Form Manual

Appendix A

Testing Tools

Tool	Tool Type	Uses (with respect to Testing)
Microsoft ADO	Test Management	• Azure DevOps (ADO) will be used for test management, including planning, execution, results, requirements traceability, bug management, and status reporting. • It will serve as the repository for creating and maintaining test cases, managing test execution, logging bugs, and reporting test status. Both technical and functional teams will use Microsoft ADO. • Real-time test metrics will be generated, with status reports and bug management metrics extracted from ADO. Status reports will be shared daily with the project team during test cycles. • Ensure traceability of test scripts to user requirements. • Manage bugs effectively. • Execute test plans as per the cycle. **Note:** New users must complete Microsoft ADO training before gaining access.
Tricentis TOSCA	Automated Software Testing	Test Automation software will be used to Automate Business Processes that can be used for regression testing after the project goes live
OpenText Load Runner	Software Performance Testing	Used for Performance, Load, Stress, and Scalability testing.
Selenium	Automated Software Testing	Test Automation software will be used to Automate web-based applications
Sauce Labs	Cloud Platform	Sauce Labs is a cloud-based platform for automated testing of web and mobile applications. It provides a comprehensive environment for running tests across various browsers, operating systems, and devices

Appendix B

Testing Best Practices

Best Practices

1 Early Involvement of Testing Team:
  a. Involve the testing team early in the project to review requirements and scope.
  b. The Test Lead should create test scenarios and the test strategy during requirement gathering workshops.

2 Test Case Writing Guidelines:
  a. Test cases should be detailed enough for any tester to follow.
  b. Test scenarios and SIT (System Integration Testing) test cases should be reviewed by the Business Analyst (BA) or business stakeholders.
  c. Create test case scripts as use cases.

3 Test Case Execution Guidelines:
  a. Every test script requires actual results or comments, regardless of whether it passes or fails.
  b. Failed test scripts must be linked to a defect (bug).
  c. Take screenshots for steps that indicate <<TAKE A SCREENSHOT>>.
  d. If you need to pause testing, press "pause execution," then "Save and Close." Resume by clicking "resume test."

4 Defect Management:
  a. Defects should be closed with a root cause analysis.
  b. Report defects promptly and ensure they are properly documented.

5 Documentation:
  a. Add a Project Architecture Diagram to the Test Strategy document.

6 Automation Testing:
  a. For automated tests, record scripts as early as possible; don't wait until coding is completed.

7 Additional Guidelines:
  a. Work with the manual test lead to identify and define the scope of automation and assess regression impact.
  b. Implement and maintain CI/CD pipelines in Azure DevOps (ADO).
  c. Publish daily and weekly status reports to stakeholders.
  d. Participate in sprint planning and manage the automation team effectively

Good Candidates for Test Automation

1. Good candidates for test automation include:
  a. Regression test -Ensuring that new code changes do not negatively impact existing functionality.

   b. Smoke Tests: Quick checks to ensure basic functionalities work after a build.
   c. Integration Tests: Verifying that different modules or services work together as expected.
   d. Data-Driven Tests: Running the same set of tests with different data inputs to validate various scenarios.
   e. End-to-End Tests: Validating entire workflows and user journeys from start to finish.
   f. Configuration Tests: Verifying different configurations of hardware, software, and network setups.
   g. Cross-Browser and Cross-Platform Tests: Ensuring the application works consistently across different web browsers and operating systems.
   h. Repetitive Tests: Any test that needs to be run frequently during development cycles.
   i. Pre-requisite & clean up steps needed for Regression tests

Bad Candidates for Test Automation

2. Bad Candidates for Test Automation include:
   a. Complex pre-requisites or cleanups required
   b. Low-value tests would be seldom or rarely used
   c. Test for high volatile code (frequent changes)
   d. Test requiring human validations
   e. Tests highly dependent on external system
   f. Non-Deterministic tests

Appendix C

Test Documentation for QMS Processes

Each project is assigned a Quality Reviewer from Stryker QMS. Following test documentations are required as part of QMS processes.

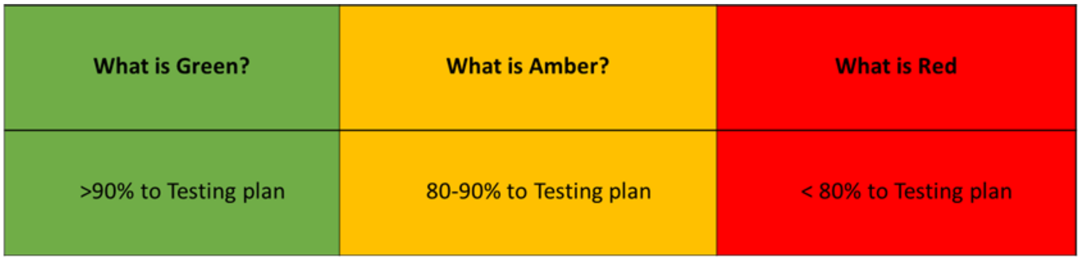
Document name	Description	Template Link
Test Strategy	Should be sent to QR for approval at the start of the project Should be signed by QAS team (Reach out to tcoe@stryker.com)	T05 Testing Strategy.docx
SIT Test suite (only if it is a validated cycle)	Pat of test closure document: Export the SIT test suite from ADO, include all test steps, execution logs and attachments QR should provide approval / eSignature on test suite	T07 Test Script
UAT test suite	Part of test closure document: Export the UAT test suite from ADO, include all test steps, execution logs and attachments QR should provide approval / eSignature on test suite Include all Executed test cases in UAT. Add following details 1. Test suite details 2. Script Id 3. Title 4. Status 5. Executed By 6. Independent Reviewer	T07 Test Script
Traceability Matrix	Prepare a traceability matrix for all requirements to test cases There could be multiple test cases linked to single requirements Include following in a tabular format 1. User story/ URS id 2. UAT Test script Id	T04 Traceability Matrix
Deviation Log	Include all the defects and its details Add resolution for all open defects – when it will be fixed, are there any work arounds, etc.	Deviation log
Test Summary report	Summarize Testing across all phases – sprints, SIT, UAT Include coverage, automation etc. as per the template	T09 Testing Summary.docx

Appendix D

Test Metrics

1. Quality of Project

a. Quality of a Program/project can be defined as Red, Amber or Green based on below metrics.



What is Green?	What is Amber?	What is Red
>90% to Testing plan	80-90% to Testing plan	< 80% to Testing plan

b. Testing Plan can deviate from its plan because of 2 things

i. Non availability of code/environment/data/resources to test

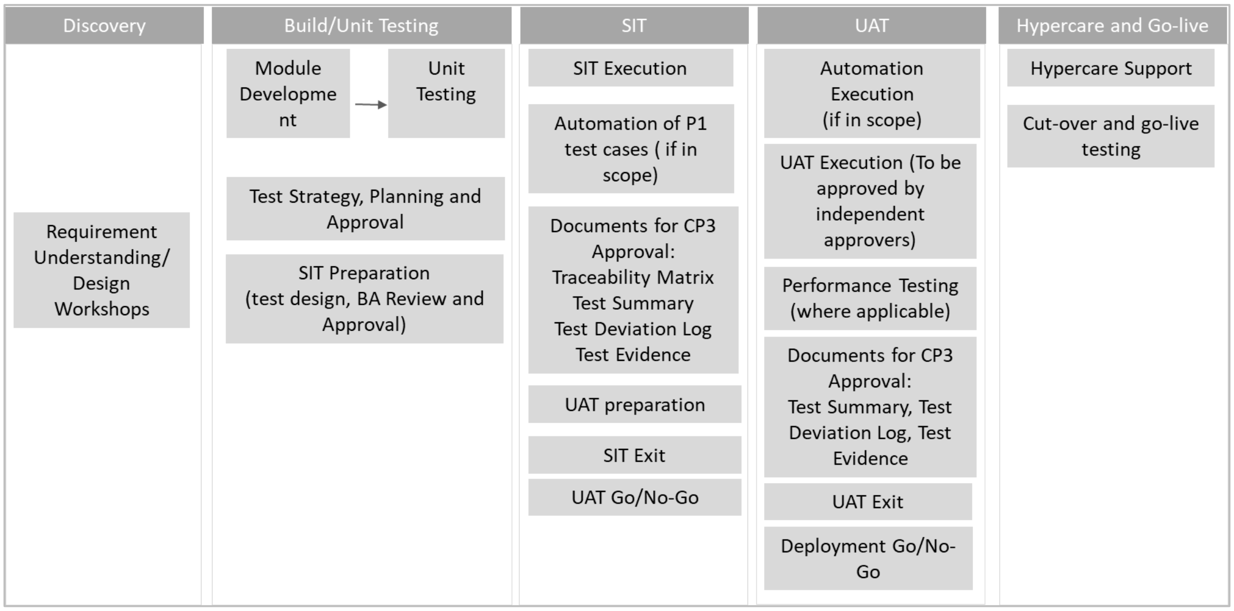
ii. Testing was done but failed

2. Testing Metrics Definition

Metric Description	Data Source	Metric	Measurement Formula	Acceptable Score
Defect Leakage SIT1 to SIT2 / SIT to UAT	ADO	Defect Leakage SIT1 to SIT2 / SIT to UAT	(No. of defects in SIT2/No. of Defects in SIT1)*100 Or (No. of defects in SIT/No. of Defects in UAT)*100	<10%
Defect Leakage SIT2 to UAT	ADO	Defect Leakage SIT2 to UAT	(No. of defects in UAT/No. of Defects in SIT2)*100	<5%
Requirement Coverage	ADO	Requirement Coverage	% Requirements associated with test cases	100%
Test Case Effectiveness	ADO	Test Case Effectiveness	(No. of defects linked to test cases/Total no. of defects) *100	>95%
Defect Leakage Post Go Live	Service Now	Defect Leakage Post Go Live	(No. of production defects/Total no. of defects identified in testing)*100	<5% P1 or P2 <10% P3 or P4 or P5
Regression Automation Penetration	ADO	Regression Automation Penetration	No. of automated regression test cases/No. of total regression test cases	>70%

Appendix E
Testing Workflows

1. Type I Projects - Waterfall



Discovery

Requirement Understanding/ Design Workshops

Build/Unit Testing

Module Developme nt → Unit Testing

Test Strategy, Planning and Approval

SIT Preparation (test design, BA Review and Approval)

SIT

SIT Execution

Automation of P1 test cases (if in scope)

Documents for CP3 Approval:
Traceability Matrix
Test Summary
Test Deviation Log
Test Evidence

UAT preparation

SIT Exit

UAT Go/No-Go

UAT

Automation Execution (if in scope)

UAT Execution (To be approved by independent approvers)

Performance Testing (where applicable)

Documents for CP3 Approval:
Test Summary, Test Deviation Log, Test Evidence

UAT Exit

Deployment Go/No-Go

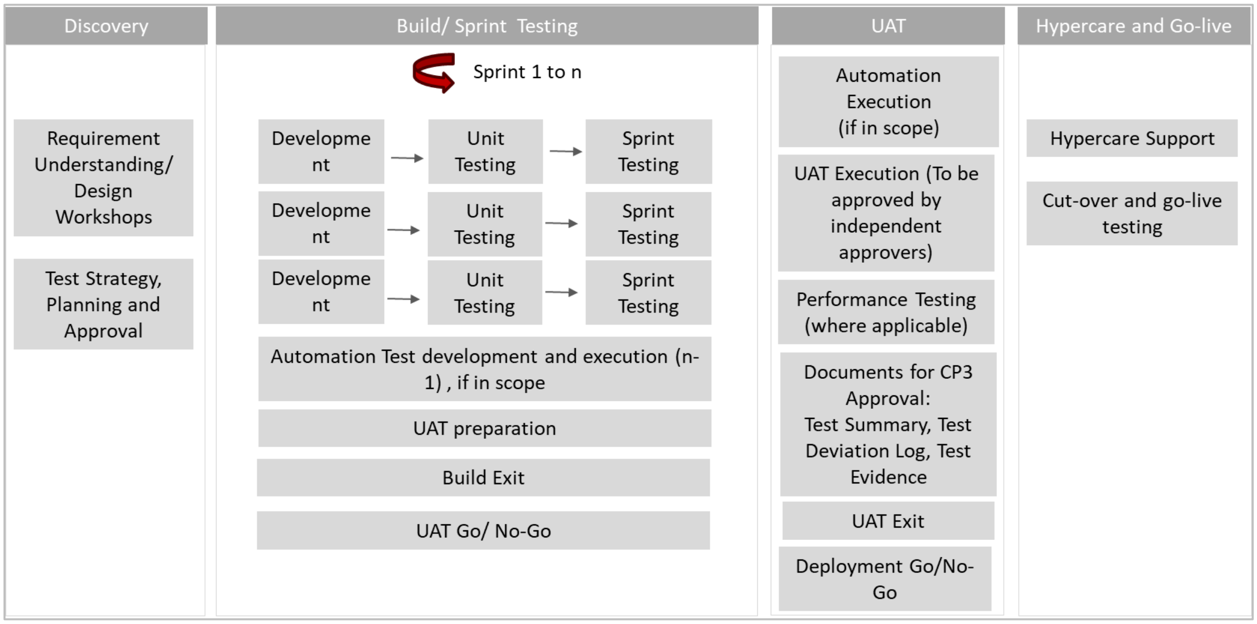
Hypercare and Go-live

Hypercare Support

Cut-over and go-live testing

2. Type I Projects – Agile

Testing is integrated into each sprint, with frequent iterations and feedback



Discovery

Requirement Understanding/ Design Workshops

Test Strategy, Planning and Approval

Build/ Sprint Testing

Sprint 1 to n

Developme nt → Unit Testing → Sprint Testing

Developme nt → Unit Testing → Sprint Testing

Developme nt → Unit Testing → Sprint Testing

Automation Test development and execution (n-1) , if in scope

UAT preparation

Build Exit

UAT Go/ No-Go

UAT

Automation Execution (if in scope)

UAT Execution (To be approved by independent approvers)

Performance Testing (where applicable)

Documents for CP3 Approval:
Test Summary, Test Deviation Log, Test Evidence

UAT Exit

Deployment Go/No-Go

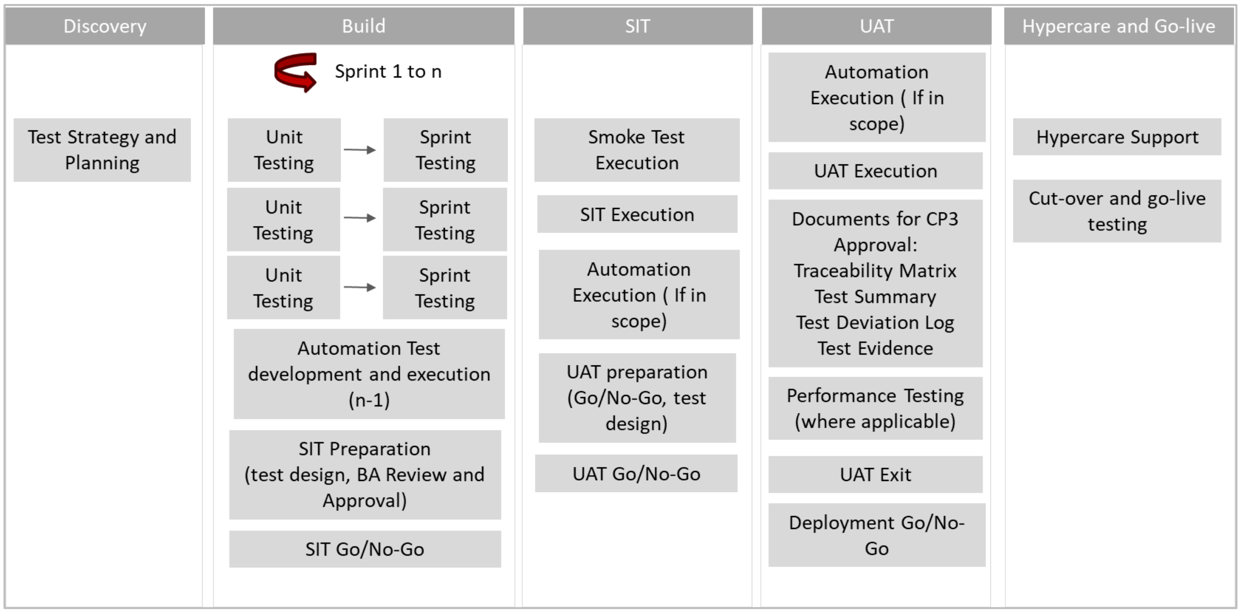
Hypercare and Go-live

Hypercare Support

Cut-over and go-live testing

3. Type I Projects – Hybrid

A combination of single-phase testing and iterative testing during certain milestones



Discovery
- Test Strategy and Planning

Build
- Sprint 1 to n
- Unit Testing → Sprint Testing
- Unit Testing → Sprint Testing
- Unit Testing → Sprint Testing
- Automation Test development and execution (n-1)
- SIT Preparation (test design, BA Review and Approval)
- SIT Go/No-Go

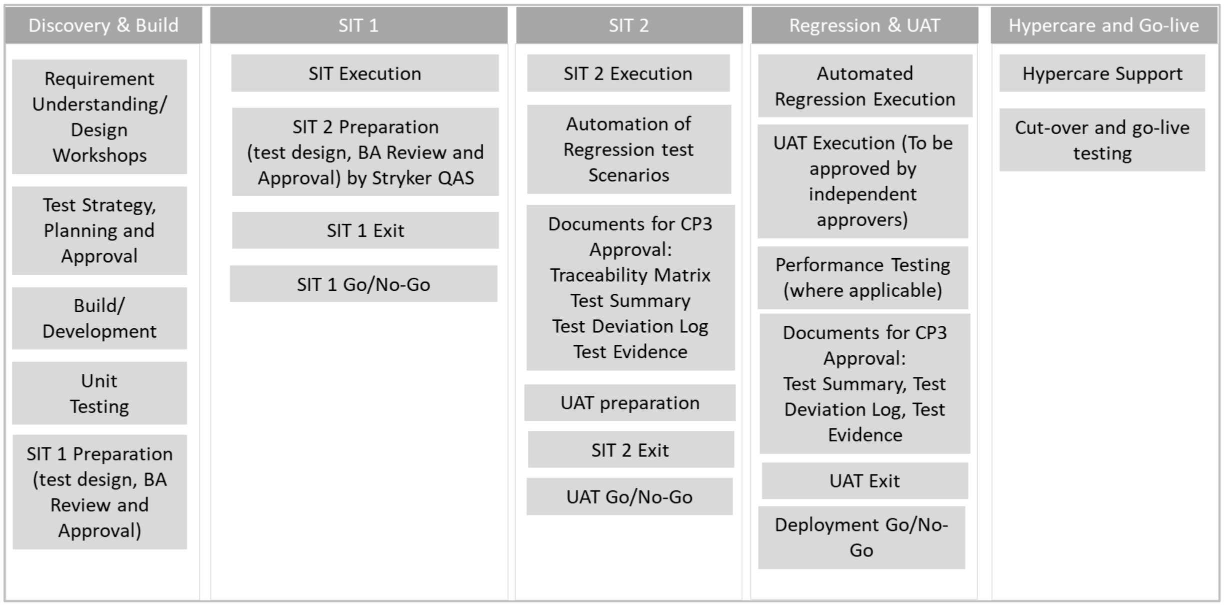
SIT
- Smoke Test Execution
- SIT Execution
- Automation Execution (If in scope)
- UAT preparation (Go/No-Go, test design)
- UAT Go/No-Go

UAT
- Automation Execution (If in scope)
- UAT Execution
- Documents for CP3 Approval: Traceability Matrix Test Summary Test Deviation Log Test Evidence
- Performance Testing (where applicable)
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

4. Type II Projects – Waterfall

A Sequential testing phases, with each type of testing completed before moving to the next phase



Discovery & Build
- Requirement Understanding/ Design Workshops
- Test Strategy, Planning and Approval
- Build/ Development
- Unit Testing
- SIT 1 Preparation (test design, BA Review and Approval)

SIT 1
- SIT Execution
- SIT 2 Preparation (test design, BA Review and Approval) by Stryker QAS
- SIT 1 Exit
- SIT 1 Go/No-Go

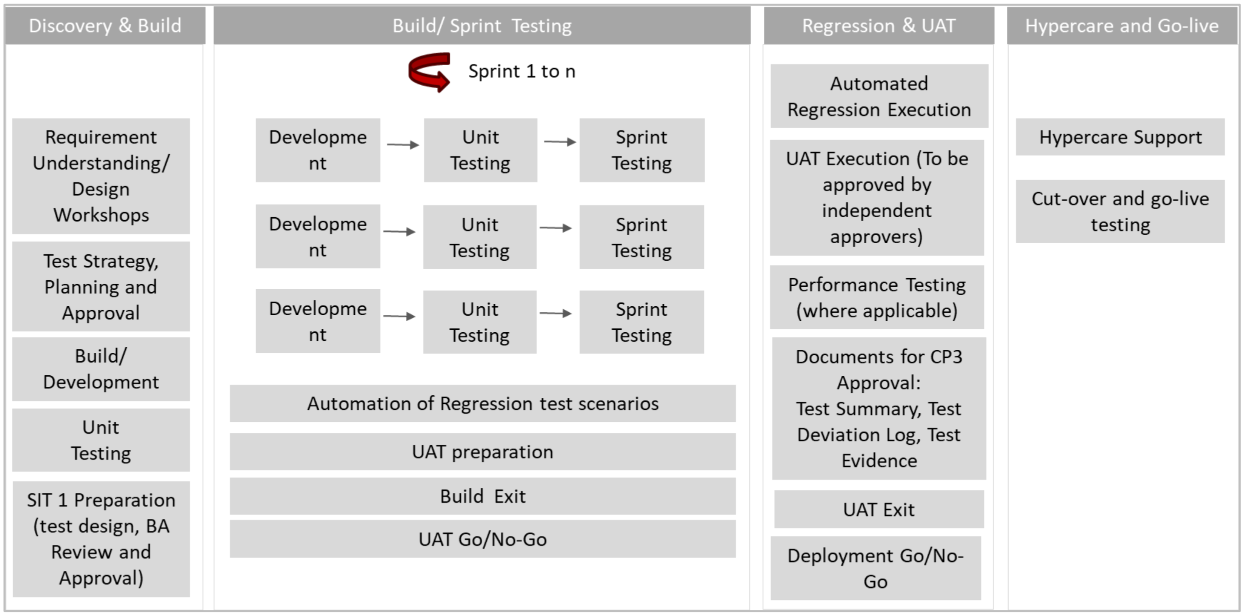
SIT 2
- SIT 2 Execution
- Automation of Regression test Scenarios
- Documents for CP3 Approval: Traceability Matrix Test Summary Test Deviation Log Test Evidence
- UAT preparation
- SIT 2 Exit
- UAT Go/No-Go

Regression & UAT
- Automated Regression Execution
- UAT Execution (To be approved by independent approvers)
- Performance Testing (where applicable)
- Documents for CP3 Approval: Test Summary, Test Deviation Log, Test Evidence
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

5. Type II Projects – Agile

Automated testing with each integration, along with regression testing to ensure stability



Discovery & Build
- Requirement Understanding/ Design Workshops
- Test Strategy, Planning and Approval
- Build/ Development
- Unit Testing
- SIT 1 Preparation (test design, BA Review and Approval)

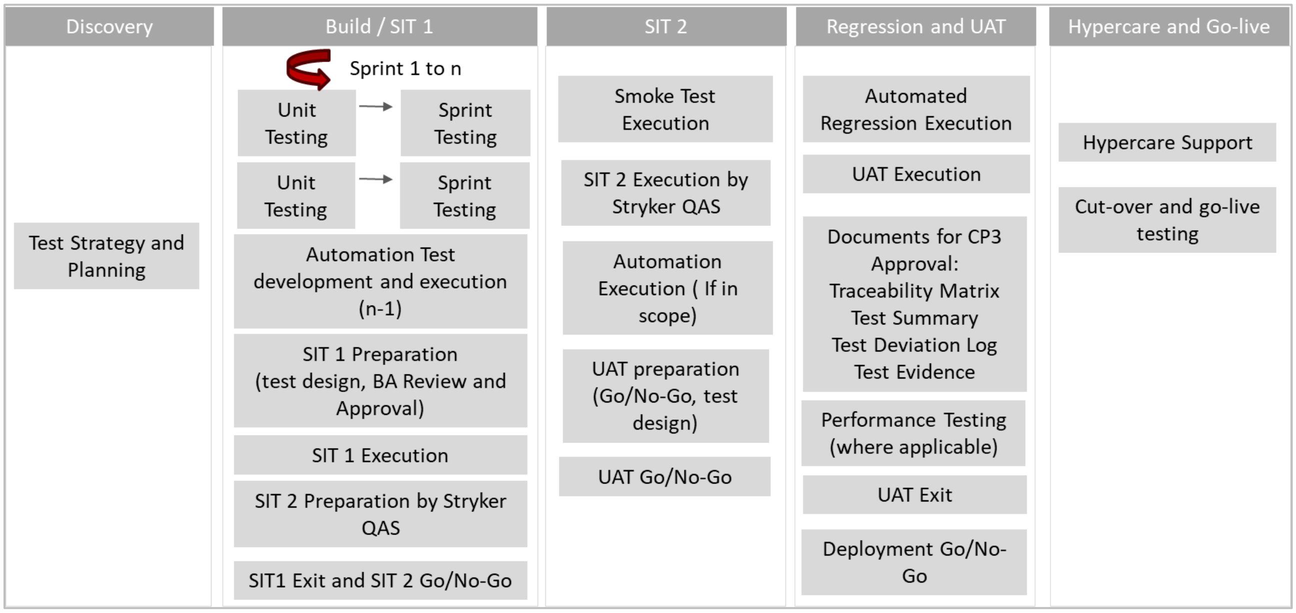
Build/ Sprint Testing
- Sprint 1 to n
- Development → Unit Testing → Sprint Testing
- Development → Unit Testing → Sprint Testing
- Development → Unit Testing → Sprint Testing
- Automation of Regression test scenarios
- UAT preparation
- Build Exit
- UAT Go/No-Go

Regression & UAT
- Automated Regression Execution
- UAT Execution (To be approved by independent approvers)
- Performance Testing (where applicable)
- Documents for CP3 Approval: Test Summary, Test Deviation Log, Test Evidence
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

6. Type II Projects – Hybrid

Initial comprehensive testing followed by iterative testing with continuous feedback loops



Discovery
- Test Strategy and Planning

Build / SIT 1
- Sprint 1 to n
- Unit Testing → Sprint Testing
- Unit Testing → Sprint Testing
- Automation Test development and execution (n-1)
- SIT 1 Preparation (test design, BA Review and Approval)
- SIT 1 Execution
- SIT 2 Preparation by Stryker QAS
- SIT1 Exit and SIT 2 Go/No-Go

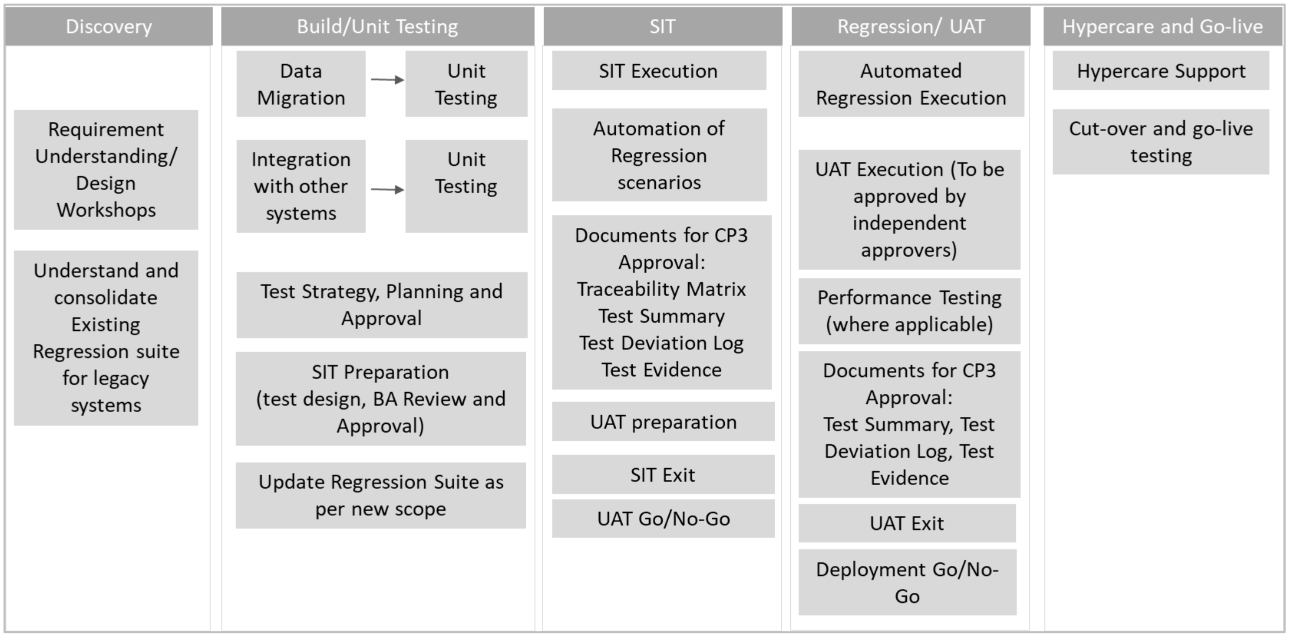
SIT 2
- Smoke Test Execution
- SIT 2 Execution by Stryker QAS
- Automation Execution (If in scope)
- UAT preparation (Go/No-Go, test design)
- UAT Go/No-Go

Regression and UAT
- Automated Regression Execution
- UAT Execution
- Documents for CP3 Approval: Traceability Matrix Test Summary Test Deviation Log Test Evidence
- Performance Testing (where applicable)
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

7. Type III Projects – Waterfall

Initial Testing phases aligned with migration steps: data, integration, and system testing



Discovery
- Requirement Understanding/ Design Workshops
- Understand and consolidate Existing Regression suite for legacy systems

Build/Unit Testing
- Data Migration → Unit Testing
- Integration with other systems → Unit Testing
- Test Strategy, Planning and Approval
- SIT Preparation (test design, BA Review and Approval)
- Update Regression Suite as per new scope

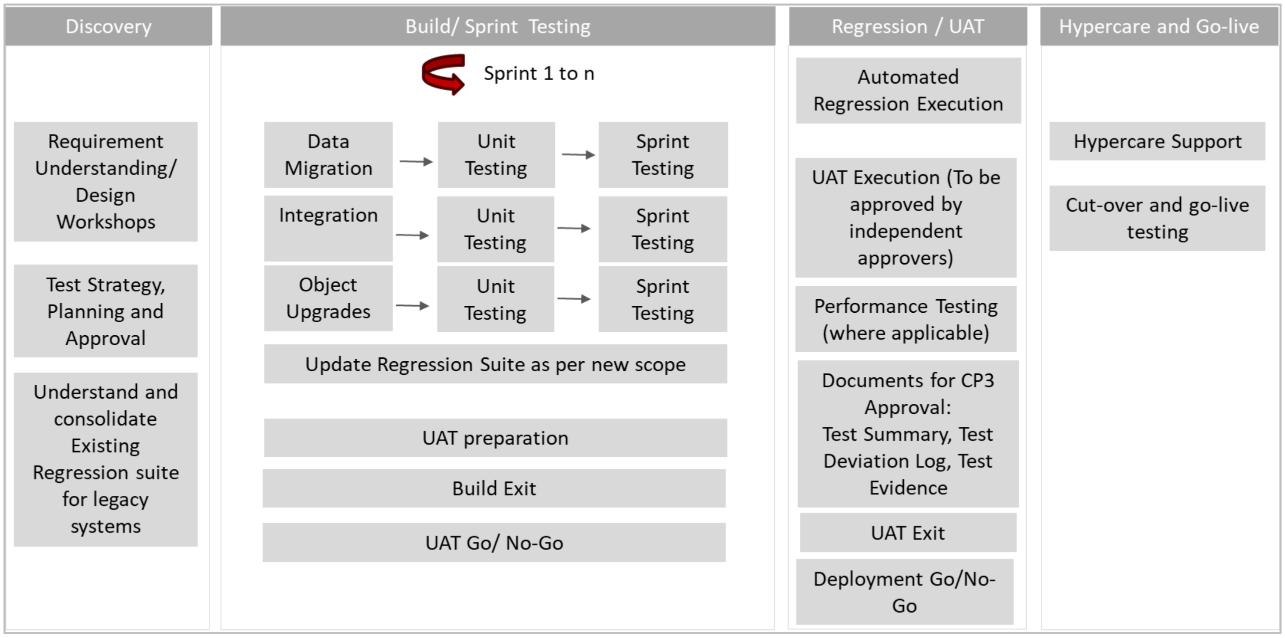
SIT
- SIT Execution
- Automation of Regression scenarios
- Documents for CP3 Approval: Traceability Matrix Test Summary Test Deviation Log Test Evidence
- UAT preparation
- SIT Exit
- UAT Go/No-Go

Regression/ UAT
- Automated Regression Execution
- UAT Execution (To be approved by independent approvers)
- Performance Testing (where applicable)
- Documents for CP3 Approval: Test Summary, Test Deviation Log, Test Evidence
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

8. Type III Projects – Agile

Continuous testing of migrated data, with ongoing integration and regression tests



Discovery
- Requirement Understanding/ Design Workshops
- Test Strategy, Planning and Approval
- Understand and consolidate Existing Regression suite for legacy systems

Build/ Sprint Testing

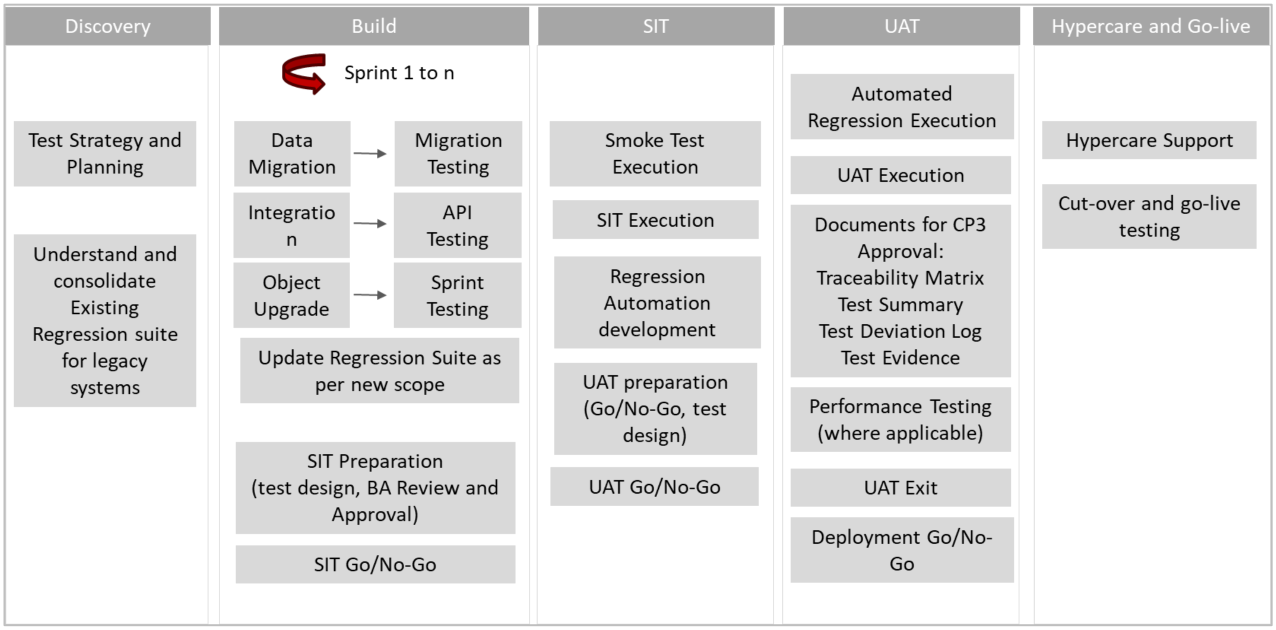
Sprint 1 to n
- Data Migration → Unit Testing → Sprint Testing
- Integration → Unit Testing → Sprint Testing
- Object Upgrades → Unit Testing → Sprint Testing
- Update Regression Suite as per new scope
- UAT preparation
- Build Exit
- UAT Go/ No-Go

Regression / UAT
- Automated Regression Execution
- UAT Execution (To be approved by independent approvers)
- Performance Testing (where applicable)
- Documents for CP3 Approval: Test Summary, Test Deviation Log, Test Evidence
- UAT Exit
- Deployment Go/No-Go

Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

9. Type III Projects – Hybrid

Initial data validation followed by phased integration testing with ongoing feedback



Discovery
- Test Strategy and Planning
- Understand and consolidate Existing Regression suite for legacy systems

Build
- Sprint 1 to n
- Data Migration → Migration Testing
- Integratio n → API Testing
- Object Upgrade → Sprint Testing
- Update Regression Suite as per new scope
- SIT Preparation (test design, BA Review and Approval)
- SIT Go/No-Go

SIT
- Smoke Test Execution
- SIT Execution
- Regression Automation development
- UAT preparation (Go/No-Go, test design)
- UAT Go/No-Go

UAT
- Automated Regression Execution
- UAT Execution
- Documents for CP3 Approval: Traceability Matrix Test Summary Test Deviation Log Test Evidence
- Performance Testing (where applicable)
- UAT Exit
- Deployment Go/No-Go

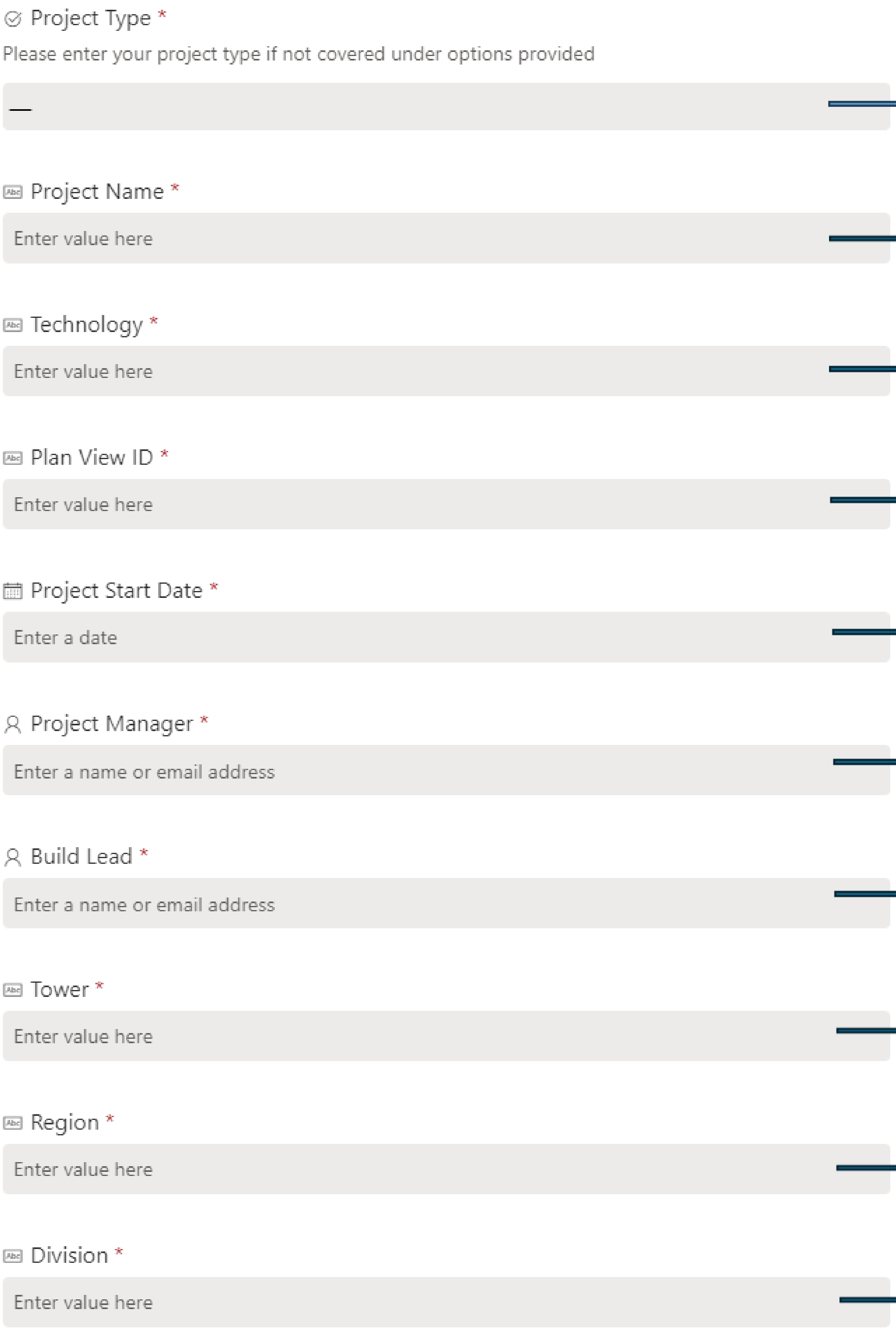
Hypercare and Go-live
- Hypercare Support
- Cut-over and go-live testing

Appendix F

Demand Intake Form Manual

Demand Intake Form

This document provides a detailed overview of the fields included in the Demand Intake Form, which is designed to streamline the process of capturing, assessing, and prioritizing project requests. The purpose of this guide is to clarify the significance and usage of each field, ensuring consistent and accurate data entry.



Project Type *

Please enter your project type if not covered under options provided

—

This field contains 3 options (Please refer to section 6 for options). If your type is not covered under the provided options, please enter the type and select it from dropdown

Project Name *

Enter value here

This is a text field. Please enter the name of your project in this field.

Technology *

Enter value here

This is a text field. Please enter the technologies that are being used to develop the project in this field.

Plan View ID *

Enter value here

This is a text field. Please enter the Plan View ID of your project in this field.

Project Start Date *

Enter a date

This is a Date field. Please select the Start date of your project here.

Project Manager *

Enter a name or email address

This is a name field. Please enter the name/email of your project manager in this field.

Build Lead *

Enter a name or email address

This is a name field. Please enter the name/email of Build Lead of your project in this field.

Tower *

Enter value here

This is a text field. Please enter the Tower name under which your project comes under.

Region *

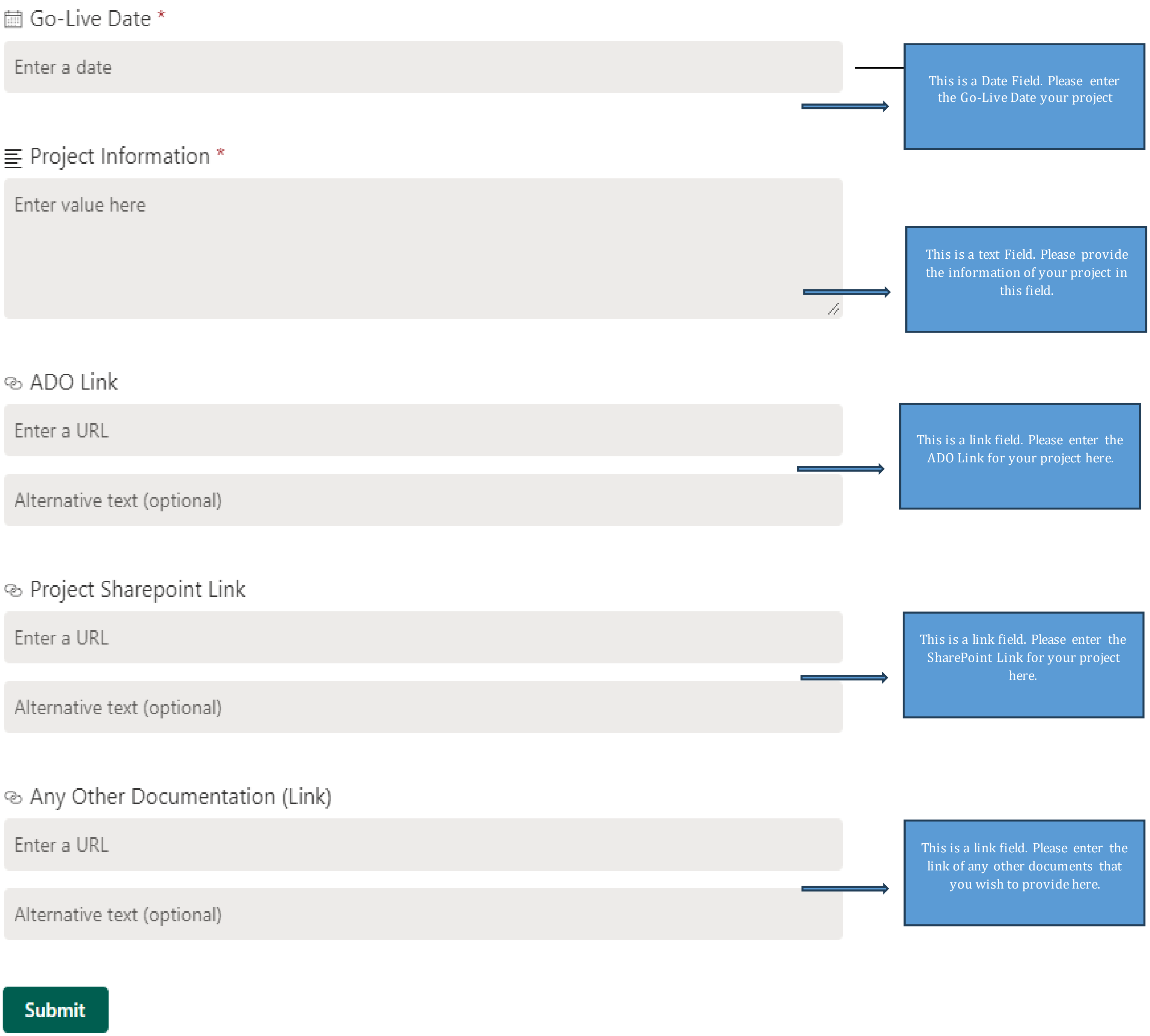
Enter value here

This is a text field. Please enter the Region for which your project will be deployed in.

Division *

Enter value here

This is a text field. Please enter the Division of your project.



Go-Live Date *

Enter a date

This is a Date Field. Please enter the Go-Live Date your project

Project Information *

Enter value here

This is a text Field. Please provide the information of your project in this field.

ADO Link

Enter a URL

Alternative text (optional)

This is a link field. Please enter the ADO Link for your project here.

Project Sharepoint Link

Enter a URL

Alternative text (optional)

This is a link field. Please enter the SharePoint Link for your project here.

Any Other Documentation (Link)

Enter a URL

Alternative text (optional)

This is a link field. Please enter the link of any other documents that you wish to provide here.

9. Revision history

Revision number	Effective Date	Description	Reason (CR/CN)

Note: If this section is used, list no more than 3 previous revisions.