**Test Strategy**

Table of Contents

[Document Information 4](#_Toc63455640)

[Document Edit History/Revision History 4](#_Toc63455641)

[Document Review/Approval History 4](#_Toc63455642)

[Distribution of Final Document 4](#_Toc63455643)

[1 Introduction 5](#_Toc63455644)

[1.1 Purpose of the document 5](#_Toc63455645)

[1.2 Scope 5](#_Toc63455646)

[1.3 In Scope/Out of Scope 6](#_Toc63455647)

[1.3.1 High Level Project Scope 6](#_Toc63455648)

[1.3.2 Application/Interface level scope 7](#_Toc63455649)

[1.4 Project Objectives and Approach/ Project Overview 7](#_Toc63455650)

[1.5 Reference Documents 7](#_Toc63455651)

[2 Test Organization 8](#_Toc63455652)

[2.1 Organization Chart 8](#_Toc63455653)

[2.2 Roles and Responsibilities 8](#_Toc63455654)

[2.3 Documentation/Deliverables 9](#_Toc63455655)

[2.4 Testing Resources 9](#_Toc63455656)

[3 Test Environment/Tools/Configuration 10](#_Toc63455657)

[3.1 Test Environment 10](#_Toc63455658)

[3.2 Test Tools 11](#_Toc63455659)

[3.3 Test Data Requirement 11](#_Toc63455660)

[3.3.1 Setting up of Test data/Test Data Gathering 12](#_Toc63455661)

[4 Test Scope, Strategy and Testing Phase Specific Considerations 12](#_Toc63455662)

[4.1 Sprint Testing Cadence 12](#_Toc63455663)

[4.2 Test Approach Collaboration 13](#_Toc63455664)

[4.3 Sprint testing Approach for multiple profiles 13](#_Toc63455665)

[4.4 INT Deployment Cadence 13](#_Toc63455666)

[4.5 Journey of User Story Migration from org to PROD 13](#_Toc63455667)

[4.6 Test Strategy/Test Approach 14](#_Toc63455668)

[4.6.1 Functional Testing 14](#_Toc63455669)

[4.6.2 System Testing 14](#_Toc63455670)

[4.6.3 Regression Testing 15](#_Toc63455671)

[4.7 Sprint Testing 15](#_Toc63455672)

[4.7.1 Unit Testing 15](#_Toc63455673)

[4.7.2 Sprint Acceptance Testing 15](#_Toc63455674)

[5 Test Schedule 16](#_Toc63455675)

[6 Test Management 16](#_Toc63455676)

[6.1 Test Cycle Criteria 16](#_Toc63455677)

[6.1.1 Deloitte QA Entry Criteria 16](#_Toc63455678)

[6.1.2 Deloitte QA Exit Criteria 17](#_Toc63455679)

[6.1.3 INT Entry Criteria 17](#_Toc63455680)

[6.1.4 INT Exit Criteria 17](#_Toc63455681)

[6.1.5 UAT Entry Criteria 18](#_Toc63455682)

[6.1.6 UAT Exit Criteria 18](#_Toc63455683)

[6.2 Test Status Definitions 18](#_Toc63455684)

[6.3 Test Review Process 18](#_Toc63455685)

[6.3.1 Review Process 18](#_Toc63455686)

[6.3.2 Review Checklist 19](#_Toc63455687)

[7 Devices/OS versions 20](#_Toc63455688)

[8 Defect Management 20](#_Toc63455689)

[8.1 Defect Life Cycle 20](#_Toc63455690)

[8.2 Severity 21](#_Toc63455691)

[8.3 Defect Triage and Communication Process 22](#_Toc63455692)

[9 Dependencies, Assumptions & Constraints 23](#_Toc63455693)

[9.1 Dependencies 23](#_Toc63455694)

[9.2 Assumptions 25](#_Toc63455695)

[9.3 Constraints 26](#_Toc63455696)

[10 Risk and Contingency Planning 26](#_Toc63455697)

[11 Glossary 27](#_Toc63455698)

|  |
| --- |
| **Test Strategy** |

# Document Information

|  |  |
| --- | --- |
| **Document Name** | Testing Strategy |
| **Project Name** | <<Project Name>> |
| **Client** | <<Client>> |
| **Document Author** | <<Author Name>> |
| **Document Version** | 1.0 |
| **Document Status** | Draft |
| **Date Released** | <<Date of release of document>> |

# Document Edit History/Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION | DATE | ADDITIONS/MODIFICATIONS | PREPARED BY/REVISED BY |
| 1.0 | <<Date>> | Document Creation | <<Author Name>> |
| 1.1 | <<Date>> |  | <<Author Name>> |
| 1.2 | <<Date>> |  | <<Author Name>> |
| 1.3 | <<Date>> |  | <<Author Name>> |

# Document Review/Approval History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATE | Reviewed By | Email | Approval Status | Team |
| <<Date>> | <<Reviewer Name>> |  | Approved | <<Client>> |
| <<Date>> | <<Reviewer Name>> |  | Approved | Deloitte |
| <<Date>> | <<Reviewer Name>> |  | Approved | Deloitte |

# Distribution of Final Document

The following people are the designated recipients of the final version of the document:

|  |  |  |
| --- | --- | --- |
| NAME | Email | ORGANIZATION |
|  |  | <<Client>> |
|  |  | <<Client>> |
|  |  | <<Client>> |
|  |  | <<Client>> |
|  |  | Deloitte |
|  |  | Deloitte |
|  |  | Deloitte |
|  |  | Deloitte/<<Client>> |

# Introduction

## Purpose of the document

The purpose of the Test Strategy for <<Client>> is to document the scope and approach of testing to be performed, resources needed, and document the schedule of intended test activities executed for the capability. It identifies user stories in scope, key milestones for the testing effort, test team and cross-functional team roles, standardized methodology, processes and deliverables, and any risks requiring contingency planning.

This document defines the approach and structured method for preparing and executing the test cases. This is a living document that will be refined as the project progresses.

## Scope

The document articulates how the test strategy will be executed for planned testing activities. The project will follow an agile methodology and will include several releases spanned across the project timelines. Each release may include a varying number of sprints, and each sprint would last for <<number of weeks>> each. UI/Functional Testing, Sprint Acceptance Testing, will be executed each sprint after the development is complete. Integrated Testing, Regression Testing will be planned for each release.

Test Plan Deliverable is organized into the following sections:

* **Test Strategy/Test Approach**

Articulates the overall test strategy, testing objectives, principles, and defines test scope, assumptions, constraints, dependencies, and risks.

* **Testing Phases**

Articulates all the testing phases that would be applicable to CRM in context of the project

* **Entry and Exit Criteria**

Outlines the combination of certain conditions which must be met in order to commence and close a testing phase/User Story

* **Test Environmental Needs**

Provides the dependencies, boundaries, and limitations for each environment and the testing activities that will be conducted in each environment

* **Testing Tools**

Lists the names and purpose of the testing tools, supporting software, and document repository that will be used during the test phases

* **Roles and Responsibilities**

Outlines the roles and responsibilities of the participants for each of the test phase.

* **Test Schedule**

Provides the high-level test execution schedule for the planned test phases

* **Defect Management Process**

This section covers the defect management process, defect triage, defect severity levels, and defect logging guidelines

## In Scope/Out of Scope

Below table defines the testing scope for the involved stakeholders during each phase of test cycle through the different salesforce orgs. Refer <<Link of SOW document>> for additional details.

High Level Project Scope

* **IN SCOPE**

|  |  |  |
| --- | --- | --- |
| APPLICATION | IN SCOPE | Owner |
| <<Application Name>> | QA Sandbox Functional Testing | Deloitte |
| <<Application Name>> | System Integration Testing | Deloitte + <<Client>> |
| <<Application Name>> | User acceptance Testing | <<Client>> |
| <<Application Name>> | Regression Testing & validations in pre-Prod environment | <<Client>> |
| <<Application Name>> | Dress Rehearsal Testing | <<Client>> |
| <<Application Name>> | Production Validation Testing | <<Client>> |
| <<Application Name>> | QA Sandbox Functional Testing | Deloitte |
| <<Application Name>> | System Integration Testing | Deloitte + <<Client>> |
| <<Application Name>> | User acceptance Testing | <<Client>> |
| <<Application Name>> | Regression testing & validations in pre-Prod environment | <<Client>> |
| <<Application Name>> | Production Validation Testing | <<Client>> |
| <<Application Name>> | QA Sandbox Functional Testing | Deloitte |
| <<Application Name>> | System Integration Testing | Deloitte + <<Client>> |
| <<Application Name>> | User acceptance Testing | <<Client>> |
| <<Application Name>> | Regression testing & validations in pre-Prod environment | <<Client>> |
| <<Application Name>> | Production Validation Testing | <<Client>> |

* **OUT OF SCOPE**

|  |  |
| --- | --- |
| APPLICATION | OUT OF SCOPE |
| <<Application Name>> | Mobile Testing, Pre-prod Regression Testing, Prod-Verification, Performance Testing, Security Testing & Data Migration Testing |
| <<Application Name>> | Mobile Testing, Pre-prod Regression Testing, Prod-Verification, Performance Testing, Security Testing & Data Migration Testing |
| <<Application Name>> | Mobile Testing, Pre-prod Regression Testing, Prod-Verification, Performance Testing, Security Testing & Data Migration Testing |

### Application/Interface level scope

Deloitte consulting QA team would be primarily working to verify the user story functionality in INT environment for all salesforce development work accomplished, whereas <<Client>> team would be responsible and verifying all the integrations with the downstream/legacy systems in the INT/UAT environment for making sure the end to end scenarios are working as expected based on the new development.

## Project Objectives and Approach/ Project Overview

<<Client>> has undertaken a multi-year digital transformation journey, the “Project” to transform its Go-to-Market approach to improve their ability to achieve its ambition to deliver customer delight. The approach includes the broad use of digital capabilities primarily including global Salesforce Service and Sales module.

## Reference Documents

|  |  |  |
| --- | --- | --- |
| TITLE | DESCRIPTION | LOCATION |
| Solution Architecture | Documents the overall technology architecture strategy and approach | <doc location> |
| Defect Management | All defects raised during testing | Jira |
| Testing Templates | Documents the templates testing team to refer while creating deliverables (Test cases) | <Location of testing templates from team’s location> |
| Wireframe Designs | Wireframes designed for the screens and navigation flow | Available against the Jira Story |

# Test Organization

## Organization Chart

<<Include the organization chart with details of key stakeholders from both Client and Deloitte.>>

## Roles and Responsibilities

The table below outlines the different testing phases and the responsible owner for each testing phase:

| TESTING PHASES | OWNED BY |
| --- | --- |
| Unit Testing | Deloitte |
| Sprint Testing | Deloitte |
| System Integration Testing | Deloitte + <<Client>> |
| User Acceptance Testing | <<Client>> |
| Pre- Prod Regression Testing | <<Client>> |
| Dress Rehearsal Testing | <<Client>> |
| Production Testing | <<Client>> |

This section outlines the detailed roles and responsibilities of participants for the different test phases of the release:

| **ROLE** | **DESCRIPTION OF PRIMARY RESPONSIBILITIES** |
| --- | --- |
| Product Owner  (<<Client>>) | * Represents the voice of the business * Defines the acceptance criteria to the user stories * Prioritizes features according to business need * Adjusts features and priority in every iteration, as needed * Defines Definition of Done (DOD) * Accepts or rejects user stories during sprint review |
| Business Team/UAT Users  (<<Client>>) | * Provides source information to the team * Provides expert business understanding of the organization * Represents the user’s area in identifying current or future procedures * Participates as required in User Acceptance Testing Activities |
| Test Manager  (Deloitte) | * Lead and Manage the QA team during the testing phases * Overall Defect Management (including turnaround times / escalations, Defect triage meetings) * Coordinate and manage resources, and stakeholders within the project for testing * Understand scope and identify impacted applications * Review all Test deliverables * Participate in Sprint/Release planning * Remove any impediments to the test team during execution |
| Scrum Master  (Deloitte) | * Coordinates with product owner and stakeholders * Ensure that the team is fully functional and productive * Maintains the processes and remove impediments to the ability of the team to deliver the iteration goal/deliverables * Facilitates scrum meetings and shields team from external interferences * Enable close cooperation across all roles and functions |
| Lead Developer  (Deloitte) | * Selects and commits to the stories for each Iteration * Estimates efforts to complete work * Responsible for writing code and delivering the product * Collaborates cross-functionally with Business Analysts and testers to ensure product meet specifications |
| Test Lead  (Deloitte) | * Understand requirements, design and test scope * Assist with the test planning activities * Review Test cases * Link test cases to requirements * Test Execution and record Test Results * Raise defects and capture defect evidence * Assign Defect Severity * Provide an update on the execution status and impediments if any in the daily scrum call on behalf of QA team. |
| QA Manual Tester  (Deloitte + <<Client>>) | * Understand requirements, design and test scope * Draft Manual Test Cases * Create/Update test cases * Link test cases to requirements * Execute manual test cases and log results * Log Defects and participate in ongoing status and issue resolution meetings |
| Developer  (Deloitte) | * Implement the User Stories * Address the defects logged * Draft Unit Test Classes * Execute Unit Test Classes |

## Documentation/Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **DELIVERABLE** | **PHASE** | **TOOL** | **FREQUENCY** |
| Test cases | Sprint testing | <<Client>> Testing Excel Format for each user story and uploaded against a user story in Jira | Every Sprint |

## 2.4 Testing Resources

Testing resources with their role within the project are listed in the table below:

| NAME | Workstream | TITLE | EMAIL |
| --- | --- | --- | --- |
| <<Resource>> | <<Module Name>> | Deloitte Test Manager |  |
| <<Resource>> | <<Module Name>> | Test Lead |  |
| <<Resource>> | <<Module Name>> | Tester |  |
| <<Resource>> | <<Module Name>> | Tester |  |
| <<Resource>> | <<Module Name>> | Tester |  |
| <<Resource>> | <<Module Name>> | Tester |  |

# Test Environment/Tools/Configuration

## Test Environment

The following table sets forth the Test Environments for the test effort presented in the Testing Phases.

Each environment is followed by a description that defines the activities that will be conducted in that environment

| ENVIRONMENT | DESCRIPTION | POINT OF CONTACT | OWNED BY |
| --- | --- | --- | --- |
| Development Sandboxes | This environment is managed by dev team and is utilized for unit testing. | DevOps/Dev Team | Deloitte |
| QA Sandbox | This environment is managed by Test team. | DevOps/Testing Team | Deloitte |
| INT | This environment is managed by <<Client>> team where System Unit Testing will occur. | <<Client>> IT Team | <<Client>> |
| UAT  User Acceptance Test Environment (Regression and dress-rehearsals) | This environment is managed by <<Client>> team where UAT and Sprint Acceptance Testing would occur. All the regression and dress-rehearsals are carried out in the org before migrating to pre-prod org | <<Client>> IT Team | <<Client>> |
| PRE-PROD  Pre-Prod ~~Regression~~ Testing Environment | This environment is managed by <<Client>> team where pre-prod mocks the build to be deployed to actual production org | <<Client>> IT Team | <<Client>> |
| PROD  Production Environment | This environment is managed by <<Client>> team to perform various tests in a production state or live environment. | <<Client>> IT Team | <<Client>> |

## Test Tools

A key component of the testing approach is to define how the test cases are maintained, executed, and how results are captured.

The project will use the following tools to support the test processes:

|  |  |  |  |
| --- | --- | --- | --- |
| TOOL NAME | VENDOR/IN-HOUSE | PURPOSE/TASK | POINT OF CONTACT FOR ACCESS |
| JIRA | In-House | This tool will be used for Defect Management and Sprint Management functionality | <<Client>>/Deloitte |
| Microsoft Office | Vendor | Test case documentation, Defect Screenshots, User story QA verification sign-off screenshots | Deloitte/<<Client>> |

## Test Data Requirement

Sample test data would be created by Deloitte and <<Client>> team as per the requirement.

| DATA REQUIREMENT | DATA SOURCE | METHOD | POINT OF CONTACT | OWNER |
| --- | --- | --- | --- | --- |
| **Unit Testing Data** | Developers to Create | Developers create test data based on the functionality and re-use the data, as necessary | Deloitte Tech Team | Deloitte |
| **INT Test Data** | Production | The test data would be created as per the requirement of the test case. Some of the guiding principles can be:  Equivalence Partitioning, Boundary Value analysis and  Segmentation of the data for QA team.  <<Client>> team, to create/facilitate the data setup required in the legacy systems. | Deloitte/<<Client>> | <<Client>> team / Deloitte |
| **Sprint Acceptance / UAT Testing Data** | Production | Refresh from production with scrubbing. Specifications to be laid out by the business team | <<Client>> | <<Client>> team |

### Setting up of Test data/Test Data Gathering

| APPLICATION/USE | TASK | OWNER |
| --- | --- | --- |
| Test Execution | Any data that needs to be input to the database prior to the test would be input using INSERT scripts. Alternatively, bulk data upload tools could be used.  Any existing data in the system might be reused for test execution.  When testing data migration, subsets and full data loads in certain cases will be used. | <<Client>> |

# Test Scope, Strategy and Testing Phase Specific Considerations

## Sprint Testing Cadence

<< Create a diagram or explain the activities to be performed wrt to a given sprint cycle>>

**Sprint QA Process** :

* To initiate the test-authoring (building test-cases) for the stories added in the sprint scope for salesforce functional testing on the story in org.
* Post the story is deployed in QA -org for verification, the QA team will execute the verification and log defects (if any) against the sprint user story in Jira and tag it to the appropriate developer for fixing the same.
* If all the defects have been addressed on the user story, the QA team member will sign-off the user story by undertaking the following items :

a. Add comment on the story once QA verification is successful, mark for technical review and move status to PENDING APPROVAL.

b. Mark the Assignee on the Story for technical review

c. Upload the Test cases with execution results and screenshots on the user story

* Based on the successful review and approval, the story would be set to ‘Move to INT’ and assigned to ‘<<Client>> DevOps – for INT deployment)
* <<Client>> DevOps team to set the story status as ‘SIT Ready’ post successful deployment and assign the story to <<Client>> SME
* Deloitte and <<Client>> will perform integrated testing of the user stories in the INT environment. Deloitte team will be responsible to validate the salesforce functionality as implemented by the sprint story in INT org. All the integrations testing in INT/UAT org would be owned and verified by <<Client>> team
* Based on successful salesforce functional verification of user story in INT org. Deloitte QA team to mark the comments on the user story for INT verification Pass and upload the artifacts (Test cases with test execution results and screenshots) as attachments on the Jira user story.
* <<Client>> QA SME to work with appropriate stakeholders to get the required sign-off on INT and mark the story Status to ‘Move to UAT’ for migration to UAT environment.
* <<Client>> team to work with <<Client>> DevOps team to have the User story migrated to UAT org and seeking the necessary sign-off on the stories from business stakeholders. Definition of Done is business approving user acceptance testing in the UAT environment.
* Regression testing cycles will be performed before each production deployment by <<Client>>
* Any issues reported during UAT and Regression testing will be resolved by Deloitte Team
* Dress Rehearsal will be performed by <<Client>> prior to final code launch.
* After business go-live approval is received, PROD testing will be performed by <<Client>> and will involve Deloitte as needed.

## Test Approach Collaboration

<< Create a diagram (flowchart) or explain in detail the collaboration between Deloitte and Client team for verification and sign-off of stories in INT and UAT environments.>>

Additional notes:

* Deloitte QA team would be performing the salesforce functional verification of the user story in the INT org.
* Deloitte QA team to upload the test execution results and screenshots on the user story while signing-off the user story in the INT org.
* All the integrations and downstream system impact or integration scenarios verification would be verified and owned by <<Client>> team in the INT/UAT org.
* <<Client>> QA SME would get the requisite sign-off’s from the stakeholders for the user story verified in UAT org.

## Sprint testing Approach for multiple profiles

<<Based on the scope of work and timelines for workstream, the decisions are reviewed and agreed upon by Product Owner/<<Client>> team and Deloitte team for mentioned scenario>>

## Client Org Deployment Cadence

<<Org Name>> Org is managed and owned by the <<Client>> Team.

Hence the deployment cadence has been agreed upon between <<Client>> team and Deloitte team for INT deployments during the sprint cycle.

<<Provide the details of the INT deployment process to be followed as part of the sprint cycle for user story migrations to INT org >>

## Journey of User Story Migration from org to PROD

<< Identify and highlight the steps that depict the user story migration along with the ‘Assignee’ and comments to be updated by the respective stakeholders. >>

## Test Strategy/Test Approach

The Test Strategy defines the quality objectives for the project and describes at a high-level how testing will be conducted for the project. The Test Strategy is based on the objective of overall testing effort which is to demonstrate that the <<Client>> solution meets specifications, system and technical requirements, correctly performs functional and technical processes based on the requirements.

### Functional Testing

Functional Testing will focus on any requirements for test that can be traced directly to User Stories or Business Functions and Business Rules. The goal of these tests is to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results. Identified below is an outline of the testing recommended for each iteration:

|  |  |
| --- | --- |
| **Objective** | Functional testing is required to validate how well the system executes the functions that it is supposed to execute for positive, negative and boundary conditions |
| **Technique** | The testing team will list out all the possible Test Scenarios and therein all the relevant Test Cases. It will then run through this list of tests manually to verify that all functions on the Application result in the expected behavior. Any deviations from expected behavior would be listed as defects in the defect tracking system which will then be passed to the development team for rectification. |
| **Completion Criteria** | Functional testing would be marked as completed if and when all planned tests have been executed, and all identified defects have either been rectified or accepted by the client as ‘known issues’. |
| **Special Considerations** | Non-availability of the build or proper acceptance criteria of the User stories will affect the timelines of the testing cycle. |

### System Testing

|  |  |
| --- | --- |
| **Objective** | System testing helps validate that the application performs all functions as outlined in the high-level functional requirements document. |
| **Technique** | Manual Testing will be performed |
| **Completion Criteria** | System testing would be marked completed when all test cases have been successfully executed and all open defects are either closed or have been accepted by the client as “open issues” that do not need to be addressed in a particular release cycle. |
| **Special Considerations** | All components of the application need to have been successfully implemented and integrated for system testing to begin.  Also, the testing environment would be ready with all the necessary data input. |
| **Responsibilities** | Responsible to ensure that the testing environment is available on time. Provide High Level Product requirements and workflows.  Deloitte QA team to verify the acceptance criteria for the user story are verified and working as expected in INT org.  <<Client>> team to verify all the integrations from salesforce to legacy systems and vice-versa for these transactions in INT org. |

### Regression Testing

|  |  |
| --- | --- |
| **Objective** | Regression testing is carried out to ensure that any functionality which was found to be working fine prior to a bug/ prior sprints should not break post the fix deployment/current sprint implementations |
| **Technique** | This includes identifying a set of test cases that need to be performed as the first test suite on any deployed build for Testing. This is known as the regression test suite.  As a fix is received by the testing team, all the relevant test in the regression test suite would be identified and executed first to ensure that the basic functionality that was working in the previous release is not broken in the new release.  This will be applicable to builds deployed within a Sprint for Testing |
| **Completion Criteria** | When all the tests in the regression test suite are executed then the regression testing is said to be completed. |
| **Special Considerations** | A regression test suite covering the basic functionality needs to be ready prior to this testing. |
| **Responsibilities** | Deloitte QA team to perform the sprint regression for incremental build from sprint N to sprint N + 1.  <<Client>> QA/SME Team will be owning the regression for Integrations, INT/UAT/Pre-Prod/Prod dress-rehearsals before migrating the changes to PROD orgs. |

## Sprint Testing

### Unit Testing

|  |  |
| --- | --- |
| **Objective** | Unit testing is the process of testing individual units of functionality. A unit can be defined as a task or the smallest testable part of an application. Validates that the developed product/functionality meets user story acceptance criteria |
| **Technique** | Developers will perform Unit Testing of each user story as a part of every sprint when a new functionality is introduced. This will be done through manual testing and Salesforce test classes. Note: Salesforce.com recommends a minimum of 75% code coverage for a production environment. |
| **Environment** | Unit Testing will take place in the Development Sandbox Environment |
| **Responsibilities** | Developers will be responsible for Unit Testing and defect resolution |

### Sprint Acceptance Testing

|  |  |
| --- | --- |
| **Objective** | As a part of Sprint Acceptance Testing/UAT, QA Members from the scrum Team will be performing functional testing and Key Processes Testing. |
| **Technique** | User Story exit criteria/acceptance criteria will be determined in the sprint planning process by the Product Owner, Scrum Master and Functional Team. The test scenario will be created by the QA Members from the Scrum Team based on the acceptance criteria of the user story and following the test strategy. |
| **Environment** | Sprint Acceptance Testing will take place in the UAT Environment |
| **Responsibilities** | The Product Owner and Functional Team will be responsible for User Story exit criteria, which will allow the QA Members from the Scrum Team to perform test case identification.  Details of Deloitte QA and <<Client>> QA team during the sprint cycles is documented as part of section 4, please refer the same for details.  The QA Members from the Scrum Team will be responsible for writing Test cases, execution, and reporting bugs (if any) identified during verification phase. |

# Test Schedule

<<Create a diagram that represents the schedule for the implementation for all the workstreams>>

This project is using an agile approach. Each sprint will be of 3 weeks duration.

# Test Management

## Test Cycle Criteria

### Deloitte QA Entry Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Test Management tool | All User Stories have acceptance criteria defined and signed off and meets the Definition of Ready (DOR) criteria |
| Test Cases have been created/reviewed |
| Test Process | Smoke Test has passed |
| Unit Testing | Development Team Unit Testing is complete |
| Test Data | Required Test Data is created or available and back up created if required |
| Environment | Test Environment/Salesforce is up and Running with the correct Build installed |
| Testing team is provided with necessary access to Test Environment/Salesforce orgs |
| Appropriate users are made available in the Test Environment for QA team to accomplish verification of the user story |

### Deloitte QA Exit Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Test Metrics | All the testcases have been executed for the user story |
| Test Metrics | All the acceptance criteria listed on the user story is verified and passed in the QA environment. All the test execution results and screenshots are attached on the user story. |
| Defects | There are no open Critical, High and Medium severity defects for the user story |
| Discrepancies | Any agreed discrepancies have been recorded and an action plan to resolve should be in place and needs to be approved by the Product owner |
| Design Approval | All design reviews have been completed and approved by <<Client>> Design Board. Technical review completed during story demos to ensure that what has been configured/coded matches approved design. |

### <<Client>> INT Entry Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Test Process | QA exit criteria has been met. Deloitte QA team to verify the salesforce functionality implemented and <<Client>> team is responsible for verifying the integrations and functionalities working in the downstream systems. |
| Test Data | Required Test Data is created or available |
| Environment | INT environment is up and Running with the correct Build installed and all the integrations are active in this environment. |
| Deloitte team is provided with necessary access to INT environment to verify the salesforce application functionality as part of the sprint user story scope. <<Client>> team to own and verify the integrations getting impacted as part of this functionality in the downstream systems. |

### <<Client>> INT Exit Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Metrics | Deloitte QA team to verify all the salesforce functionalities are verified and passed in INT environment.  <<Client>> team to verify all the integrations are working as expected in the INT environment with this functionality deployed.  <<Client>> QA SME to mark the story to ‘Move to UAT’ post successful verification and sign-off in INT org. |
| Metrics | All the testcases for the user story to be verified are executed in the INT org. |
| Defects | All Critical Defects have been addressed, re-tested and verified by the Deloitte/<<Client>> Team as raised |
| Considerations | Deferred defects are documented, and plan of action is approved by PO or client stakeholders before providing a go ahead for UAT/Production move |

### 6.1.5 <<Client>> UAT Entry Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Test Process | INT exit criteria has been met |
| Test Data | Required Test Data is created or available |
| Defects | All Critical Defects have been addressed, re-tested and verified by the Deloitte/<<Client>> Team as raised in INT org |
| Environment | UAT Environment/Salesforce is up and Running with the correct Build installed |
| Business team is provided with necessary access to UAT Environment |

### 6.1.6 <<Client>> UAT Exit Criteria

| APPLICATION/AREA | TASK |
| --- | --- |
| Test Process | All the acceptance criteria listed on the user story have been successfully tested and signed-off by Business/<<Client>> SME/PO |
| Defects | All critical Defects have been addressed, re-tested and verified by the UAT Team |
| Considerations | Deferred defects (if any) are documented before providing a go ahead for Production move |

## Test Status Definitions

The definition of Test Statuses is as follows:

| **TEST OUTCOME STATUS** | **DEFINITION** |
| --- | --- |
| NO RUN | Test Case has not been executed for the Sprint |
| PASS | All the Test Steps in the Test Case have been executed and Passed. The application performs as mentioned in the Expected Results |
| FAIL | Any of the Test Step in the Test Case fails. The application does not perform as presented in the Expected Result |
| BLOCKED | The Test Case or Step is blocked from testing by an already raised/known defect |

## Test Review Process

### Review Process

Following review process will be applied to all the Testing Deliverables. Below figure shows the implementation of the Process to a Sprint Deliverable: Test Scenarios and Test Cases

### Review Checklist

Following parameters will be utilized to create the Checklist for Self-Review/ Peer-Review process:

* Template Adherence

Columns in the Test Case spreadsheet in accordance with the Pre-defined Templates

* Test Case steps and Expected Result are synchronized

Expected Result should be carrying the exact outcome of the Test Step

* Validation Results

Test Result column should be carrying the Test Case Result

* Precise Test Description

Acceptance Criteria should not be copied for the Test Case description. It should be crisp and should highlight the key intent of Validation

* Preconditions

Preconditions to the test case execution if required should be available in the Test Case

* Test coverage

Appropriate level of positive and negative test scenarios is identified for the test coverage of the user story.

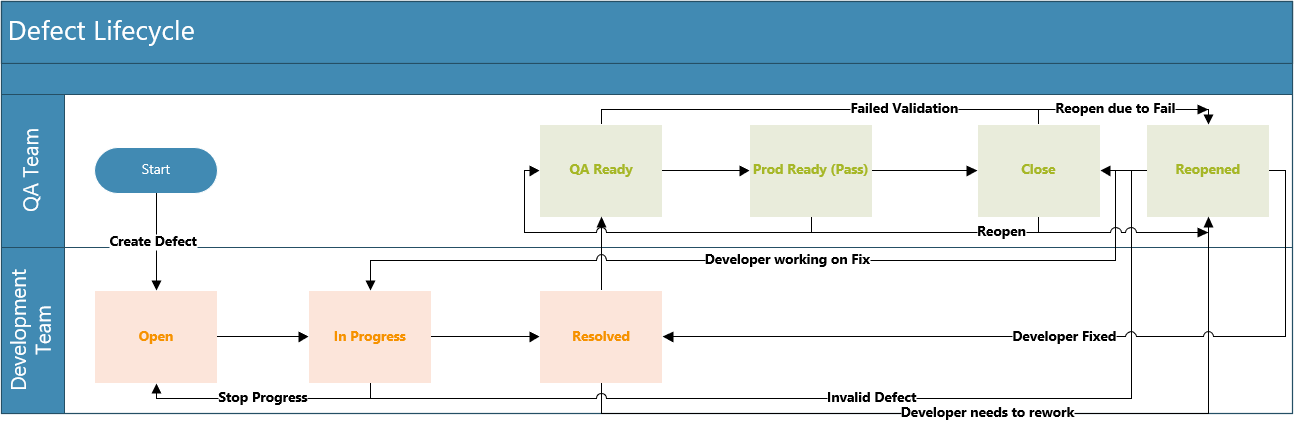
# Devices/OS versions

* <<Mention the list of in scope browser versions and breakpoints>>

# Defect Management

## Defect Life Cycle

Following Jira defect life cycle will be followed as part of the project:



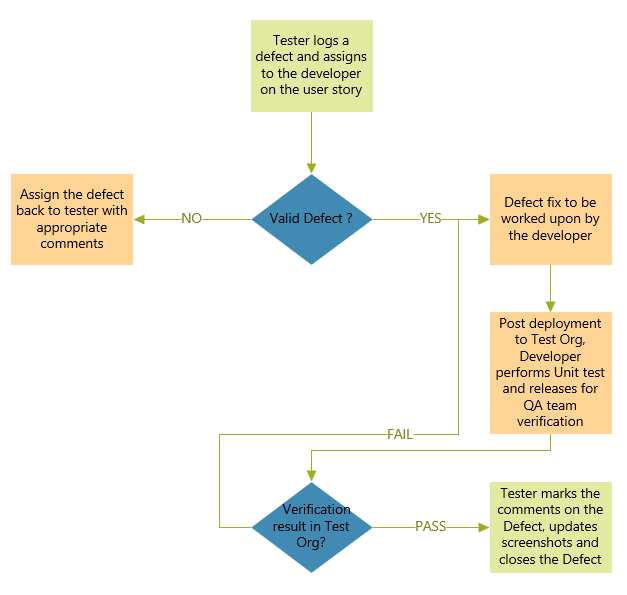
|  |  |
| --- | --- |
| **STATUS** | **DESCRIPTION** |
| Open | Defect logged by the tester, this is the default Status of the observation logged and assigned to the Developer |
| In progress | Defect has been assigned to a Developer for analysis. Once the problem is determined, the fix will be made. |
| Resolved | The observation is fixed in dev environment and awaiting deployment in Test environment |
| QA Ready | The observation has been unit tested by developer in Test environment and released for verification for the QA team. The QA tester is revalidating the observation in the Test environment. |
| Prod Ready (Pass) and Close | The defect is validated in the test environment and the observation has been addressed successfully |
| Re-opened | The observation is not addressed completely, hence assigned back to the developer for fix |
| Won’t Fix (Deferred) | Defect has been deferred to future, based on the discussion with all the stakeholders. Mark appropriate comments and update the status. Note this status is available in the ‘Resolution Dropdown’ available once user selects the ‘Resolve Issue’ option. |
| Duplicate | A similar observation is logged already, mark appropriate comments and the Defect ID of the similar observation logged. Note this status is available in the ‘Resolution Dropdown’ available once user selects the ‘Resolve Issue’ option. |
| Cannot Reproduce | Not a valid observation, the application is behaving as expected. Note this status is available in the ‘Resolution Dropdown’ available once user selects the ‘Resolve Issue’ option. |

## Severity

|  |  |
| --- | --- |
| **Severity LEVEL** | **DESCRIPTION** |
| Severity 1:  P1- Blocker | Details: Application is unavailable or critical processing cannot continue and there is no work-around. The defect causes severe usability problems that prevent the use of the application, or the loss or corruption of data. Examples of severity 1 defects may include: entire system is down, user is unable to open cases, site performance, data security or privacy issues, excessive page load time All Severity 1 defects must be fixed prior to go live. |
| Severity 2:  P2- Critical | Details: The defect prevents the user from using the application in the manner intended due to a major function being disabled or incorrect. However, this defect will not stop the users from utilizing the application. Examples of severity 2 defects may include: Fields missing in customer account migration, chat failing to find next agent available, unable to edit account records, phone numbers do not initiate a phone call when tapped, excessive page load time, etc. All Severity 2 defects must be fixed prior to go live. |
| Severity 3:  P3- Major | Details: The application does not function as expected, however, the deviation is not serious or does not impact an entire functional area, and an acceptable workaround exists.  Examples of severity 3 defects may include: errors when saving records, limiting profile access, person accounts not linking, excessive page load time Severity 3 defects must be reviewed and a specific plan to fix or defer fix to the Hypercare period immediately following go live must be developed. |
| Severity 4:  P4- Minor | Details: Problem relates to minor cosmetic or documentation content. Software/data acceptance is not dependent on problem resolution. This level indicates superficial problems, which have no immediate impact on application function or availability.  Examples may include: minor alignment shifts of images/content, spelling errors in content, field validation rules are incorrect, link is underlined when it should not be, etc. Severity 4 defects must be reviewed and agreed to defer resolution post go live. Resolution will not be required in scope for this statement of work. |

## Defect Triage and Communication Process

The following diagram represents the high-level defect triage process.



Apart from adhering to the established defect triage process, additional communication protocol is to be followed depending on the severity of the defect:

|  |  |
| --- | --- |
| PRIORITY LEVEL | COMMUNICATION PROCESS |
| P1 | Communicate the Issue to the Test Manager/ Dev Lead. Log the Defect in the defect management tool and highlight in the Scrum Stand-up |
| P2 | Communicate the Issue to the Test Manager/ Dev Lead. Log the Defect in the defect management tool and highlight in the Scrum Stand-up |
| P3 | Log the Defect in the defect management tool |
| P4 | Log the Defect in the defect management tool |

# Dependencies, Assumptions & Constraints

## Dependencies

This section will list the dependencies between the different testing phases identified during the development of this document. Each dependency will have an associated impact and the owners of these dependencies. The table below identifies the dependencies identified for the different testing phases:

|  |  |  |  |
| --- | --- | --- | --- |
| TESTING PHASE | DEPENDENT UPON | POTENTIAL IMPACT OF DEPENDENCY | OWNER |
| Unit Testing | Development of software components needs to be completed before unit testing commences. | Unit testing may get delayed and overall schedule may be impacted. | Deloitte |
| Unit testing checklist is reviewed and ready for validation. | Thorough unit testing may not be performed, and defects may get pushed to QA/System Integration Testing. |
| Test data should be defined for the unit testing checklist prior to unit testing execution commencing. | Lack of data may delay certain functionality to be validated during unit testing. |
| Sprint Acceptance Testing / UAT | Development of Software Components and Unit testing should be completed before the Sprint Acceptance testing commences | Testing may get delayed or defect count may increase due to the lack of unit testing | Deloitte/<<Client>> team |
| Pre-Prod environment | Smoke and End to end testing may be delayed |
| Defect Fixes | All the critical/high defects will be addressed based on the agreed SLA |
| Boundary System Changes | Testing maybe delayed whenever there is a change in the boundary system with which salesforce interacts. |
| Test Data Setup | Deloitte team would be creating the transactional data required for verification of the story in QA org. However data dependency from other legacy systems would be owned by <<Client>> team to be made available for verification of salesforce implemented functionality. |
| System Integration Testing/ UAT | Development of software components / modules needs to be completed before system integration testing commences. | System integration testing schedule may get delayed which will impact the overall schedule.  Lack of integration of components may also result in defect leakage. | <<Client>> team/Deloitte |
| Appropriate Test users should be made available in the Org for QA verification. | Insufficient privileges or test users in the INT org, may impact the Deloitte QA teams user story verification for salesforce functionality |
| Delay in deploying the stories to INT org by <<Client>> DevOps for verification | Impacts the overall sprint schedule and increases the risk of completing the QA execution for all the stories marked for INT verification during the sprint cycle | <<Client>> team |
|  | All the integrations with the downstream systems (if any) are verified by the <<Client>> team and user story is signed-off for same | Delay in signing-off the user story through INT org during sprint cycle might impact the overall sprint scope to be delivered | <<Client>> team |
| User Acceptance Testing / UAT | UAT Testing environments need to be available prior to UAT test execution commencing. | User Acceptance Testing may get delayed due to lack of environment infrastructure which may result in schedule slippage. | <<Client>> team |
| UAT test cases written and traceability matrix completed by UAT team | UAT may get delayed due to lack of test cases |
| UAT verification from the <<Client>> business stakeholders to be completed during the sprint cycle | UAT sign-off delay from <<Client>> business will result in delay in sprint closure |

## Assumptions

This section lists the assumptions made during the development of this strategy, including any assumptions that could introduce risk into the testing process. The table below highlights assumptions identified for different testing phases:

| ASSUMPTION | IMPACT OF ASSUMPTION BEING INCORRECT | IMPACT LEVEL | OWNER |
| --- | --- | --- | --- |
| Acceptance Criteria for each iteration is reviewed and approved by <<Client>> team. Test Case creation will commence once the approved acceptance criteria is available for each User Story. | Any future changes to the Acceptance Criteria of a User Story may require changes / updates to the respective test cases. If the impacts are not significant, a Change Request will be documented; otherwise changes that affect the scope, budget, schedule, and/or effort or requested changes to signed-off deliverables of the project are formally documented through the Change Control process, prioritized, analyzed, reviewed, and approved before implementation. | High | <<Client>> team/Deloitte |
| Unit Testing – Deloitte  Sprint Acceptance Testing – Deloitte  INT – Deloitte (verifies salesforce functionality for the sprint user story) / <<Client>> team (verifies the integrations and functionality of the sprint user story)  UAT Testing – <<Client>> team  Pre-prod Regression Testing - <<Client>> team  Production Testing – <<Client>> team | Lack of clarity on who’s testing within each testing phase and what is expected from a roles and responsibility standpoint. | Medium | Deloitte/<<Client>> team |
| Transactional Test Data is created by the testing team for the user story verification purpose during the sprint testing. Any data dependency outside the salesforce system is owned by <<Client>> team. | Lack of data may delay certain functionality to be validated during System Integration Testing. | High | <<Client>> team/ Deloitte |
| Stable test environment is available before the testing commences | Overall schedule and quality may get impacted | High | Deloitte/<<Client>> team |
| All high Priority defects have been addressed by the dev team and verified by the test team | Exit criteria won’t be met in case high priority defects remain open hence would impact the overall schedule | High | <<Client>> team/Deloitte |
| Deployments to the org is streamlined to enable the Deloitte QA / <<Client>> team to verify the user stories | Lack of delay in deploying the user stories to QA/ INT and UAT environment will impact the schedule and delay the sprint closure | High | Deloitte/<<Client>> team |
| <<Client>> team owns the verification of the integrations with all the legacy systems and confirms the sign-off of the stories from concerned stakeholders | Delay in seeking sign-off from the business stakeholders for INT/UAT will impact the story getting marked to ‘Done’ | High | <<Client>> team |

## Constraints

| CONSTRAINT | | DESCRIPTION |
| --- | --- | --- |
| Collaboration | <<Client>> team and Deloitte testing team needs to collaborate effectively for a successful test planning and execution. | |
| Environmental Constraint | Connectivity to <<Client>> systems should be seamless to ensure Deloitte team don’t run into any latency related issues. | |
| Access Issue | <<Client>> team to designate a point of contact to address all the access related issues for the client orgs to facilitate the Deloitte QA team members to perform the execution tasks in the required Test environments. | |

# Risk and Contingency Planning

This section lists the QA risks that have been identified, including their explanation, their mitigation Strategy, and a contingency plan to minimize the impact of the risk under the likelihood of occurrence of the risk:

|  |  |  |  |
| --- | --- | --- | --- |
| RISK | IMPACT | MITIGATION STRATEGY | CONTINGENCY PLAN |
| Availability of Application for testing on time. Delayed delivery to the test team will impact the Sprint Sign-Off by the test team | High | - Scrum Master to keep a track of Development activity with the help of Scrum team to ensure the build is delivered as per schedule.  - Dev team to define and align on the code drop dates to make the user story available in the QA environment | - QA team to identify the critical testcases to be executed to verify the acceptance criteria are met for the given user story.  - Increase the testing resources for verification of the delayed/complex stories |
| Scope Creep – as Business Unit become more familiar with the application, they will want more functionality | High | Priorities will be set and discussed by stakeholders ahead of time. | Scope of Implementation will be finalized with all the involved stakeholders for MVP release. |
| Poor Code Quality will impact the entire team | High | Identified project metrics will be utilized to track dev code quality | Stringent Code Review process to be implemented. |
| Non-Availability of Mockup Screens during the sprint planning meeting will impact the associated UI testing and a part of the Functional Testing will be affected | High | Work with the Onsite Team/BA to finalize the Mockups before the sprint start so that the testing team has requisite information to create and execute test cases | Approval to be obtained from the PO to spill the part of not tested component of the story to the next iteration, accordingly the story is moved for verification in next sprint. |
| Change in User Stories in the Middle of the Sprint will impact the testing deliverables and test sign off may be delayed | High | Testing team to work with PO and prioritize the stories they pick up for creation/execution to minimize the impact | High level scenarios will be utilized for test case execution due to the time crunch testing team will have in the test case designing |
| Test data collisions and data loss due to one test environment for multiple workstreams. | High | Test Data Management Plan will include segmentation of test data with the purpose of avoiding collision with CRM’s testing tracks for Applications if only one Test Environment is available for use. | A sub-set of data needs to be tagged as a backup data in case other projects use this data. |
| Unplanned Test Environment refresh of the Boundary Systems causing the identified test data loss. | High | Coordinating and Confirming the refresh schedules of the Boundary System Environments with the concerned stakeholders. Might have an impact on the overall schedule. | Rework a sub-set of test scenarios/cases impacted by the Boundary System Environment refresh. |
| Functionality being overwritten due to multiple workstreams deploying the code to 1 QA org | High | Development team to follow the guidelines put across to work on the latest codebase for components, as well as co-ordinate across streams for common components development | Implement the source control versioning to retrieve the correct build for the component version and re-deploy to the QA orgs and facilitate backward promotion in development sandboxes |

# 11 Glossary

| Keyword | Description |
| --- | --- |
| PO | Product Owner |
| UAT | User Acceptance Testing |