|  |
| --- |
|  |
| Test Plan Document   * Google Search Engine Verification Project |
| August 25, 2023 |

**Document Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Details | Name | Role | Date |
| **Prepared by:** | Siddhartha | QA Lead | 25-08-2023 |
| **Reviewed by:** | Mr.X | Program Manager |  |
| **X Company Approved by:** | Mr.Y | Vertical Leader |  |
| **Customer Approved By:** |  |  |  |

Table - Document Approvals

**Content Header:**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 25-08-2023 | v.2.2 | Google Search Engine Testing | Siddhartha |
|  |  |  |  |

Contents

[1. Purpose 1-3](#_Toc74585739)

[LoUATion of Test Related Documentation 1-3](#_Toc74585740)

[2. Objectives 2-4](#_Toc74585741)

[3. Scope 3-5](#_Toc74585742)

[Activities 3-5](#_Toc74585743)

[Functionality 3-7](#_Toc74585744)

[Non-Functional Testing 3-7](#_Toc74585745)

[Deliverables 3-9](#_Toc74585746)

[4. Test Management 4-10](#_Toc74585747)

[Test Governance 4-10](#_Toc74585748)

[Testing Tools 4-11](#_Toc74585749)

[Microsoft Team Foundation Software (TFS) 4-11](#_Toc74585750)

[Microsoft Team Foundation Software (TFS) 4-11](#_Toc74585751)

[Test Plan Templates 4-12](#_Toc74585752)

[Functional Test Plan Template 4-12](#_Toc74585753)

[Browsers & Devices 4-12](#_Toc74585754)

[Test Phase Management 4-12](#_Toc74585755)

[Test Support and Coordination 4-12](#_Toc74585756)

[Test Entry Criteria 4-13](#_Toc74585757)

[Test Suspension and Resumption Criteria 4-13](#_Toc74585758)

[Test Results 4-13](#_Toc74585759)

[Test Exit Criteria 4-15](#_Toc74585760)

[Metrics Definition 4-16](#_Toc74585761)

[Configuration Management 4-17](#_Toc74585762)

[CommuniUATion Plan 4-17](#_Toc74585763)

[5. Defect Management 5-18](#_Toc74585764)

[Definition of Severity Levels 5-19](#_Toc74585765)

[Defect Submission Guidelines 5-19](#_Toc74585766)

[Root Cause Analysis: 5-19](#_Toc74585767)

[Defect Life Cycle 5-20](#_Toc74585768)

[6. Project Testing Milestones <2023> 6-21](#_Toc74585769)

[7. Assumptions Risks and Issues 7-22](#_Toc74585770)

[Assumptions/Risks 7-22](#_Toc74585771)

[8. Test Roles & Integration with Verticals 8-23](#_Toc74585772)

[Test Roles and Responsibilities 8-23](#_Toc74585773)

[Matrix Engaging Test Phase, Test Activity, and Project Roles 8-24](#_Toc74585774)

[RACI Model 8-25](#_Toc74585775)

[Standard QA Activities and Roles 8-25](#_Toc74585776)

[9. Appendix 9-27](#_Toc74585777)

[Test Phase Definitions 9-27](#_Toc74585778)

# Purpose

The purpose of **software testing** is to ensure:

1. The software solution provided by X Company to the customer meets business needs and effectively supports operational processes.
2. Functional integrity across the project.
3. Minimal or no defect leakage from development to production.
4. Defined customer requirements are met.

The purpose of the **Test Plan** is to provide an overarching plan for how software testing will be managed. The document includes:

1. The overall test objectives and solution planned for this project.
2. A description of the activities, test phases, and features that are in scope and out of scope of the test team for this project.
3. The test management approach including test phase entry, exit, and suspension criteria, metrics definitions, and resource requirements for this project.
4. Tools, points of contact and communication plan.
5. Test related milestones.
6. Any assumptions, risks and issues associated with the testing of this project.
7. Appendices which describe:
   1. Acronyms and Definitions
   2. Test Phase Descriptions, Entry Criteria, Exit Criteria, Ownership and Environments
   3. Project Roles and Responsibilities

## Location of Test Related Documentation

There may be need to use Product Requirement Documents that will be a reference for all clients and would be useful in test scripting.

|  |  |  |
| --- | --- | --- |
| **Requirement Document to be referenced** | **SharePoint Link** | **Scope for Testing** |
| QA Product BRD | Link | Google Search Engine |

# Objectives

This document defines the testing approach and methodology used by X Company to ensure delivery of a high quality testing of Google Search Engine project by ensuring the following:

* Minimization of defects in deployed code
* Functional integrity across the system
* Predefined client quality standards are met
* All the client requirement should cover with the testing phases and tested on appliUATion

This document provides an overview of the approach, test phases, test infrastructure, testing life cycle, metrics and reporting used and the roles and responsibilities associated with testing activities.

The testing methodology is predicated on best practices as well as an ongoing emphasis on incorporating industry standards. The test approach has the following characteristics:

* Uniformity and consistency in the methodology
* Documentation of all test-related activities and outcomes
* Sound planning and utilization of formal work plans
* Tools applied to automate, replicate and support testing activities
* Regression testing throughout the testing life cycle
* Proactive relationships with all domain client teams.

# Scope

This Master Test Plan presents the various testing phases used to ensure that the test plan fulfills the contractual requirements defined in the SOW within JIRA feature. Test phases cover functional, interface, and system integration testing. Successful testing will ensure that the X Company system is ready to offer the services described in the SOW. The collective set of software testing phases will be referred as **Validation Testing**.

Test cases will consider the applicable hardware, software and peripheral equipment required to successfully prove the required functionality. All test results will be provided to customer and discussed during project reviews.

The scope section defines what test related activities and solution functionality are in scope and out of scope for the testing activities of this project. Please see [Appendix B – Test Phase Definitions](#_Appendix_B_–) for a complete list of possible test phases along with description, entrance, and exit criteria.

* In Scope of Test Team activities are the responsibility of the X Company test team to complete as part of this project.
* Out of Scope of Test Team activities are not the responsibility of the X Company test team to complete as part of this project. These activities may be completed by other members of the X Company team, as outlined in the Project Management Plan. The owners of these items are identified as applicable.
* Out of Scope of Project activities will not be undertaken for this project but are listed to avoid ambiguity.

## Activities

The following table lists test related activities associated with this project and whether the activity is in or out of scope of the test team. If it is out of scope of the test team, the last column indiUATes the responsible organization.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | In Scope of Test Team | Out of Scope of Test Team | Out of Scope of Project | Responsible Organization |
| Requirements Management |  | *✓* | Choose an item. | *Requirement Team* |
| Test Planning | ***✓*** |  | Choose an item. | *Testing Team* |
| Data Mining | ***✓*** |  | Choose an item. | *Testing Team/Client Team* |
| Defect Management | ***✓*** |  | Choose an item. | *Testing Team* |
| Test Summary Reporting | ***✓*** |  | Choose an item. | *Testing Team* |
| Unit Testing |  | ***✓*** | Choose an item. | *Dev Team* |
| System Integration Testing | ***✓*** |  | Choose an item. | *Testing Team* |
| Regression Testing | ***✓*** |  | Choose an item. | *Testing Team* |
| Security Testing | ***✓*** |  | Choose an item. | *Specialized testing team* |
| Performance Testing | ***✓*** |  | Choose an item. | *Specialized testing team (Partially)* |
| Single Sign ON | ***✓*** |  | Choose an item. | *Client Team* |
| User Acceptance Testing (UAT) | ***✓*** |  |  | *Client Team* |
| Client Acceptance Testing (UAT) | ***✓*** |  | Choose an item. | *Client* |
| Responsive Web Design (RWD) | ***✓*** |  | Choose an item. | *Testing Team* |

Table 2 - In and Out of Scope Activities

## Functionality

The following table lists test related activities associated with this project and whether the activity is in or out of scope of the test team. If it is out of scope of the test team, the last column indicates the responsible organization.

The following functionality list is in scope of the test team for testing during the project.

|  |  |  |
| --- | --- | --- |
| Component | Function | Feature |
| Web | Transactional Screens | * As per BRD scope |
| Web | Data View Screens | * As per BRD scope |
| Web | Content Screens | * As per BRD scope |
| Web | Landing Pages | * As per BRD scope |
| Web | Common Screens | * As per BRD scope |
| Web | Look & Feel (UX and Branding) | * As per BRD scope |
| Web | Contents | * As per BRD scope |
| Web | RWD | * As per BRD scope |
| Web | Navigations | * As per BRD scope |
| Non-Functional | CertifiUATe | * In scope for Non-Functional Testing |
| Non-Functional | Browser Settings | * In scope for Non-Functional Testing |
| Non-Functional | Layout and Design | * In scope for Non-Functional Testing |
| Non-Functional | Session | * In scope for Non-Functional Testing |

Table 3 - In-Scope Functionality

## Non-Functional Testing

The following table lists the non-functional items that test team will be covering as a part of this project.

|  |  |  |
| --- | --- | --- |
| Component | Function | Feature |
| Responsive Web Design (RWD) | | |
| Browser | Chrome | As per Requirements |
| Browser | Mozilla Firefox | As per Requirements |
| Browser | MS Edge | As per Requirements |
| Mobile Testing | | |
| Device | iPhone | As per Requirements |
| Device | iPad | As per Requirements |
| Device | Android | As per Requirements |

Table 4 – Non-Functional Testing

The following functionality list is out of scope of the test team for testing during the project:

|  |  |  |
| --- | --- | --- |
| Component | Function | Feature |
| Database Testing | All | * Out of Scope |

Table - Out of Scope Functionality

## Deliverables

The following test related collaterals will be delivered as a part of this project:

|  |  |  |
| --- | --- | --- |
| Deliverable | Owner | Description |
| Master Test Plan  *(this Document)* | Test Manager | Describes the approach, methodology, tools, resources, milestones, and test acceptance criteria |
| Test Schedule | Test Manager/Lead | Defines the specific test related tasks, duration, and dependencies, required to achieve the project timeline. This schedule will be incorporated into the overall project schedule. |
| Test Cases | Test Lead | List of tests (execution steps, conditions, data, and expected outcomes) that align to requirements and intended to validate that the solution meets requirements and specifications. |
| Metric Reports | Test Lead | Report summarizing agreed metrics. See [Metrics Definition](#_Metrics_Definition) section for details |
| QA Sign off for each phase | Test Manager/Lead | Report summarizing the QA findings and Go / No go Recommendation for the next test phase |
| Final Test Summary Report | Test Manager/Lead | Report published at the end of all QA testing to summarize approach, findings, defect status, and Go / No go recommendation from QA. |
| Customer Sign-off | Test Manager/Lead | Sign off request for key deliverables including:   * Test Plan * Test Cases |

Table 6 - Deliverables

# Test Management

Testing is a critical step to ensure the integrity and viability of the solution within an environment that closely matching the customer’s own. A common practice of software testing is that testing is performed by an independent group of testers after the functionality is developed and before it is shipped to the customer. X Company utilizes a rigorous test execution approach, by working closely with end users and development teams. This approach includes creation of test cases derived from the requirements and use cases provided. X Company also employs automation, where possible, across test phases to ensure efficiency and repeatability is achieved.

Finally, X Company follows test governance and metric reporting guidelines to ensure all requirements have been delivered as per specifications and within agreed service levels.

## Test Governance

X Company follows a software test phase methodology during software test execution. Each test phase is followed by a report and customer review. The test phases and associated high level testing activities are shown in the following graphic.

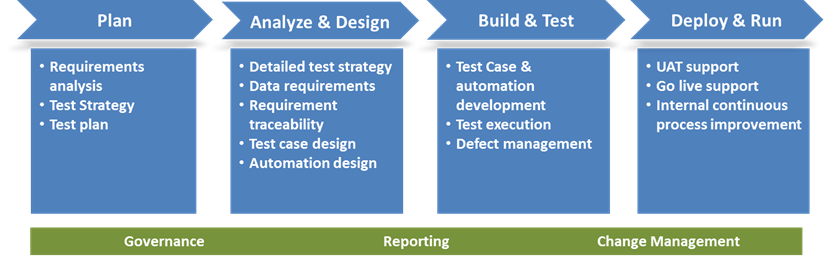


Figure - Test Methodology

The test team provides functional and performance validation of the product to be delivered. The test team will provide input to the overall project milestone review effort.

Note – All testing will be performed in QA Environment & External QA URL only.

## Testing Tools

### Microsoft Team Foundation Software (TFS) or JIRA

X Company uses this tool for management of all test artifacts and test execution status reporting. The tool includes Requirements, Test Plan, Test Lab and Defect Management modules. It offers a consistent and repeatable process for all stages of application quality management.

| Microsoft Team Foundation Software (TFS) | **Description** |
| --- | --- |
| Test Plan | * Test case storage * Tests are organized into unique groups (i.e. based on test phase, platform and/or application). |
| Test Data identification and Execution | * Identifying test data/participants * Execution of test cases for Integration, Regression and UAT * Pass/Fail of test cases recorded |
| Defect Tracking | * Defect tracking and analysis of trends to aid decision-making * Supports entire defect life cycle from detection through resolution * Allows for avoiding duplication of effort on existing defects |
| Dashboard | * Status Reporting * Testing Progress Reporting * Defect Triaging |

Table 7 - TFS

**TFS Access**

To get the access to TFS project please collate the Name with WIN ID and send same to Neil A Rixe / HRS TFS Support <HRS.TFS.Support@X Company.com>

**UAT TP on SharePoint** – The UAT TP, leveraged to facilitate Verizon Dependent Verification UAT execution, resides outside of TFS and includes the following:

UAT scenarios, a test site URL, UAT schedule, UAT Issues Log and Known Issues.

**Note:** UAT Playbook will be created by the client service team and approved by client before sharing with QA team for UAT test data identification

## Test Plan Templates

### Functional Test Plan Template

For BA7 Web Modernization, we are using a standard UAT Test Plan Template that would be in TFS ready to upload format and it is same for all clients.

### Browsers & Devices

It’s recommended to get the following Browsers installed at your desktop and use below mentioned Mobile Devices for RWD testing for Verizon 87 Dependent Eligibility Verification (DEV) Feature –

**Browsers –**

* ~~IE 11~~
* Chrome (v80+)
* Mozilla Firefox (v70+)
* Microsoft Edge (v44+)

**Devices –**

* iPhone (iOS 9 +)
* Android (7 +)
* iPad (iOS 9+)

Note – Latest version of the Browsers and Devices would be preferred.

## Test Phase Management

Testing begins during software development. X Company developers create or modify the code to support the requested features, and perform unit and development integration tests against their code to ensure, at the component level, that the code functions as expected. The X Company test team continues the testing process to ensure quality by understanding requirements, developing test scenarios, test cases, and X Companying various test phases which may include functional, integration, performance, security and other tests. The test lead reviews the results of the developer level testing prior to the start of validation testing to ensure it has been completed and all results are understood. Prior to the start of testing, the test lead reviews all test cases with peers and customers to ensure proper test coverage and quality. These reviews ensure all possible positive, negative, and boundary conditions are tested. Finally, the test lead passes along results of the validation testing to the customer prior to the start of UAT.

### Test Support and Coordination

The X Company test team often supports certain test phases, which are not its responsibility to execute, such as UAT. For support, the X Company test team will work with the responsible organization to:

* Answer questions related to developing a test execution schedule or test plan including questions related to the content of the testing, defect management, communication and so on.
* Provide test cases as used for X Company testing phases, if requested.

### Test Entry Criteria

The X Company test team will develop a comprehensive set of test cases to test all aspects of the solution delivered to the customer. Each test case will map to a requirement (feature) in TFS. All requirements and the features will be reviewed and approved by the customer prior to the start of any test execution activities.

|  |  |
| --- | --- |
| UATegory | Entry Criteria |
| Requirement | All requirement should be in Sharepoint/TFS/JIRA. |
| Unit Testing | The Unit Testing should be completed and Test plan in TFS. |
| Critical Bugs | There would be no critical bugs open, unless agreed upon by the stakeholders. |
| Major/Minor Bugs | All open major/minor bugs have been documented and the stakeholders have agreed to move forward with release of the software. |
| Test Bed | Test Environment should be ready and steady |

Table 8 – Entry Criteria

### Test Suspension and Resumption Criteria

A temporary suspension of testing activities may become warranted during the execution of a testing phase if certain conditions, or combinations of conditions, develop. This suspension may involve certain functional areas, functionality within an area, or the entire system.

Testing resumes when the causes for suspension have been determined, corrective action has been taken, and any appropriate retesting/regression have been performed.

Conditions factoring into test suspension and resumption include the following:

|  |  |
| --- | --- |
| Test Suspension | Test Resumption |
| Incomplete tasks on the critical path | An alternate approach to testing is defined |
| A large volume of defects  (50% of Test Cases failed in module, check the count sheet) | Correction of defects as agreed, and re-plan of test schedules |
| Critical defects | Correction of defects as agreed, and re-plan of test schedules |
| Resource shortage | Additional resource allocation or scope change |
| Incomplete test environments | Establishment of acceptable test environments |

Table 9 - Test Suspension and Resumption Criteria

### Test Results

During each phase of test execution, test execution results and defects found will be tracked to measure test progress and software quality. Defects are defined as any non-conformance to expected test results. All defects will be logged in the applicable defect tracking tool (see [Test Tools](#_Test_Tools) section of this document) as they are found. Defects will be managed as defined in the [Defect Management](#_Defect_Management) section of this document. A final test report outlining all test results will be created and provided to the customer following all test activities. The report will include a summary of the test scope, final test metrics, review of exit criteria, and testing Go / No Go recommendation. Customer will sign off on the final test document prior to release of software to customer.

### Test Exit Criteria

At the end of all X Company related test activities, the X Company test manager will issue a statement indicating whether the solution has met certain exit criteria and is ready for release to production. The following set of exit criteria shall be used to obtain that signoff:

|  |  |
| --- | --- |
| UATegory | ****Exit Criteria**** |
| Content | **GO**: All new features, changes, and defect corrections are implemented, which have been fully tested and are marked as Complete in the TFS.  **NO GO**: One or more features have not been completed, are not functioning, or have not been tested. |
| Performance | **GO**: Performance meets specifications, or is the same or better than the previously released version.  **NO GO**: Performance does not meet specifications or has decreased from the previously released version. |
| Critical Bugs | **GO**: There are no open critical defects, unless agreed upon by the Client Team or customer.  **NO GO**: There are 1 or more critical defects open in the system, not agreed upon by the Client Team or customer. |
| Major Defects | **GO**: All open major defects have been documented and the Client Team has agreed to move forward with release of the software.  **NO GO**: The Client Team not agree to move forward or the test team feel there is not an acceptable workaround for the major defects or there is a risk that the major defects will not be correction in a subsequent release. |

Table 10 - Test Exit Criteria

### Metrics Definition

Metrics are used to judge the progress and effectiveness of the testing activities. These metrics will be reported with normal project status reporting frequency. The following metrics will be collected and reported upon as part of this project:

The Test Manager/Lead runs metrics from the Quality Center tool and prepares a **Test Execution** **Status Report**. The Test Manager/Lead also runs/prepares a **Defect Log** and various **Defect Summary Reports**.

In addition, X Company testers document test results as they execute test case steps. They note inputs and outputs in each test step, compare actual to expected results, document test execution status (i.e. ‘Pass’ or ‘Fail’) and include screen shots and any other related documentation as needed.

X Company testers report results at the test case level and complete the following tasks:

* Execute all test cases/steps successfully
* Check actual and expected results and, if they match, mark all test steps within each test case as ‘Passed’
* Document actual and expected results within the Test Lab in Quality Center
* Account for and verify all report data
* Correct, successfully re-test and close any defects identified while testing

|  |  |
| --- | --- |
| Metric | Description |
| Defects by Severity | * Total number of defects found by severity and phase * Total number of defects corrected by severity and phase * Defect aging for project defects, by severity and phase |
| Test Progress | * Prior to Test Execution: * Total number of test cases created divided by number of test cases planned to be created * During Test Execution: * Total number of test cases executed divided by planned number of test cases executed * Total number of test cases planned, executed, passed and failed |
| Test Schedule Tracking | * Planned versus actual milestone start dates * Planned versus actual milestone end dates |

Table 11 - Metrics Definition

### Configuration Management

X Company closely monitors the migration of code from one environment to another through our configuration management process. This process includes tightly managed controls, testing and approvals to ensure integrity of the code.

* Code freezes are established during testing windows
* New code is migrated on a fixed schedule
* Release dates are assigned during defect tracking

The table below provides detail on the subject of data sources and usage:

| **Data** | **Additional Detail** |
| --- | --- |
| Data Sources | * **Converted Data** – Both automated and manual, data values and formatting is created through this primary source for testing data * **Manipulated Data** – Specific business requirements may require manipulation of data to successfully execute a test * **Created Data** – Some requirements or very unique scenarios may only be able to be tested by creating test data from scratch |
| Data Usage | * **Data Segmentation** – Test data is parceled out by Test Leads to prevent adverse impact of test cases and facilitate multiple groups to test in the same environment |

Table 12 – Config Management

## Communication Plan

To ensure the project team and customer are aware of the test progress, metrics, and issues, the test team will provide the following information as input to the project defined status report:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | CommuniUATions | Executed By | Frequency | Stakeholders | Escalation Path |
| 01 | Test Execution Status | Test Lead | Daily | QA Vertical | Test Manager |
| 02 | Bug Status Report | Test Lead | Daily | QA Vertical | Test Manager |
| 03 | Project Test Status | Test Manager | Weekly | QA Vertical | QA Owner |
| 04 | Defect Triaging | PMO | Daily | All | Test Manager |

Table 13 – Communication

# Defect Management

Defects will be raised during every phase of testing and will be logged into the Defect Management system at the time they are found. Note that defects found and corrected during developer unit and integration testing will not be logged unless they are not corrected prior to hand-off to the test team. Defects will be tracked closely to assess the impact on the end-user and business process and ensure that quality of the overall solution is validated. The following defect management process will be followed:

* All defects will be logged into the defined defect management tool and given a severity based on the following definition:
* **Severity**
* The extent to which the defect can impact the operation or software.
* Defined by the tester based on impact and understanding.
* Defects may also be reviewed and, if necessary, given a priority based on the following definition:
* **Priority**
* The order in which a defect should be resolved.
* Defined by tester and confirmed by stakeholders as appropriate considering the following:
* Business need for fixing the defect
* Severity/Impact
* Probability/Visibility
* Defects will be corrected by the development team based on that assigned severity and priority. Blocker and Critical defects are worked first.
* All corrected defects will be added to a software build for re-test by the test team prior to release to the customer.

## Definition of Severity Levels

The following table defines the base set of Severity levels that may be assigned to a defect. Note that the term enhancement is used in the event a defect becomes an enhancement or vice versa.

|  |  |
| --- | --- |
| Severity Level | Description |
| (1) Critical | * Turnaround Time - Rapid response * Issue requires immediate attention and testing cannot proceed without resolution or tool(s) cannot go live with issue pending |
| (2) High (No work-around) | * Turnaround Time - Target response in 24-48 hours * Issue prevents a feature of the tool from usage. Feature is considered essential for users |
| (3) Medium (work-around) | * Turnaround Time - Target response within 72 hours * Issue is significant but a feature can be used with alternate path. However usability of the feature work around is not ideal. |
| (4) Low | * Issue does not prevent the usability of the tool or feature. Average user may not identify the issue. |

Table 14 - Defect Severity Levels

## Defect Submission Guidelines

* [**Defect Reporting Guidelines**](https://sp.services.conduent.com/sites/InteRetr/doc/Product%20Testing/HPALM%20Documents/LW%206.0%202018_19%20Defect%20Reporting%20Guidelines.docx)

## Root Cause Analysis:

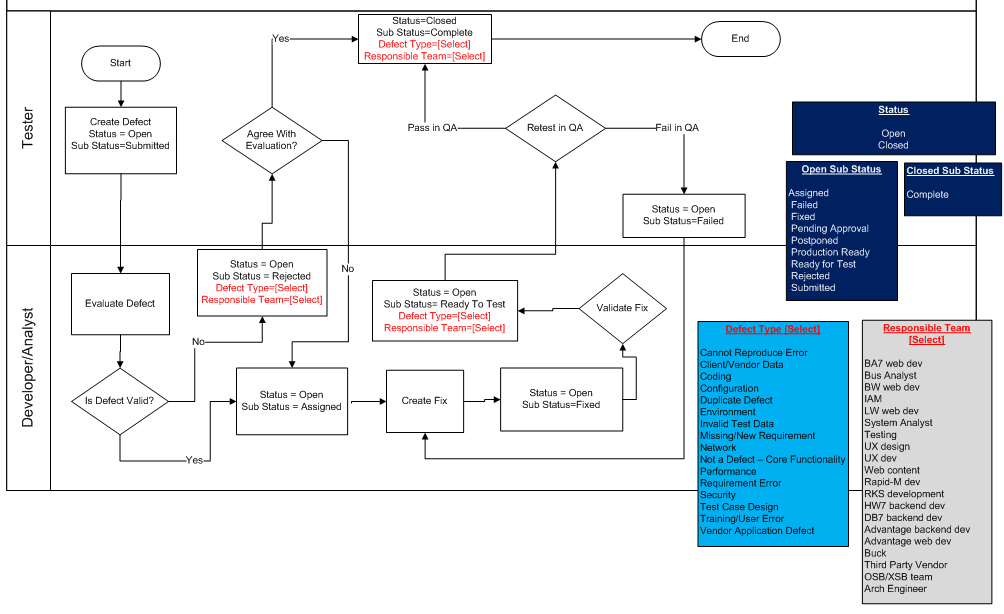
RCA (Root cause analysis) is a mechanism of analyzing the defects, to identify its cause. Below are some of the factors responsible for defect occurrence.

* Incomplete Requirement
* Coding Issue
* Configuration Issue
* Test Environment Issue
* Test Design Issue
* Test Data Issue
* Hardware Issue
* Support Software Issue
* Vendor Issue
* Network Issue

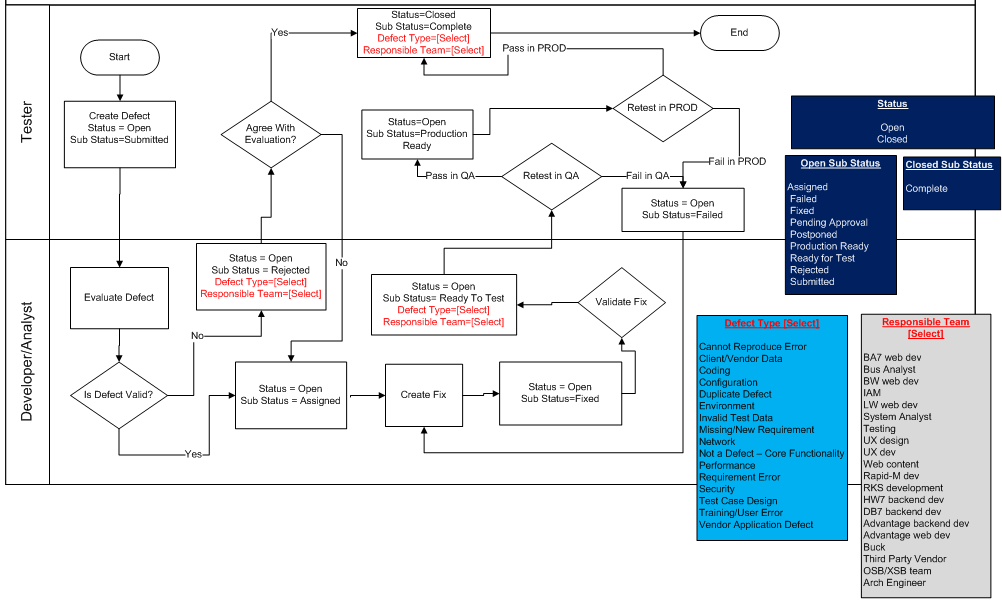
## Defect Life Cycle

Defect Life Cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it’s not reproduced. Defect life cycle is related to the bug found during testing. The defect has different states in the Life Cycle. The Life cycle of the defect can be shown diagrammatically as follows:

**Defect flow before Go Live:**

1. 

**Defect Flow Post Go Live:**



# Project Testing Milestones **<2023>**

The following testing milestones will be tracked to ensure on time completion of testing activities to support this project. The full test schedule, along with these milestones, is contained in, and will be tracked through, the overall project schedule. These milestones are subject to change based on the overall project schedule.

|  |  |
| --- | --- |
| Milestone | Completion Dates |
| Requirements Sign off | TBD |
| Test Plan sign off | TBD |
| Test Case Creation Complete | TBD |
| Test Start (SIT, Automation, Performance, Security) | TBD |
| Test End (SIT, Automation, Performance, Security) | TBD |
| Test Readiness Review Go / No Go | TBD |
| UAT Start & End Date | TBD |
| UAT Start & End Date | TBD |
| Production Release - Go Live | TBD |

Table 15 - Test Milestones

# Assumptions Risks and Issues

X Company will track and manage all assumptions, risks, and issues that are raised in the execution of this project. The test lead will review test related risks and issues with the team on a regular basis to be sure required actions are being taken to close issues in a timely manner. Test specific assumptions, risks, and issues will be provided to the project manager to be maintained at the project level (see the [Test Tools](#_Test_Tools) section for a hyperlink to the project site). The following tables provide the test related Assumptions, Risks, and Issues identified at this point.

## Assumptions/Risks

All test related assumptions will be provided to the project manager and will be maintained on the project site. The following assumptions have been made to date on this test effort:

|  |  |
| --- | --- |
| Assumptions | Notes |
| Schedule dependencies identified in the project scheduled are achieved. |  |
| Test resources are on-boarded to the project per the schedule. |  |
| All hardware and software, including appropriate licensing, is in place prior to the start of testing activities. |  |
| Customer sign off on applicable deliverables is received in a timely manner. |  |
| There will be impact in project timeline if there is major scope change | Any frequent BRD change will impact the project deliverable at risk |

Table 16 - Test Assumptions

# Test Roles & Integration with Verticals

## Test Roles and Responsibilities

The following table provides a list of various roles and responsibilities held on a project. Note: One person may hold multiple roles on the project

|  |  |
| --- | --- |
| Role | Responsibilities |
| Test/QA Manager | * Overall responsibility for the software testing activities within the scope of the project. * Manages the test team to ensure test deliverables are met and assists with resolution of issues. * Reviews test schedule, progress reports, metrics, and issues with test lead to ensure the test effort remains on track and issues are resolved or escalated quickly. * Attends kick off, requirements reviews, and status meetings to ensure effective communiUATion and issue resolution. |
| Test/ QA Lead(s) | **Defect Triaging/Management**   * Manages the day-to-day test activities and test team to ensure tasks are completed efficiently and correctly. * Reviews the work of testers to ensure high quality is achieved, and provides coaching where needed. * Provides status reports including test progress, health, metrics, and issues. * Develops test cases from available use cases. * Executes test cases, raise defects based on test results, raise issues to in a timely manner. |
| Test/QA Analyst | * Develops test cases from available use cases. * Executes test cases, raise defects based on test results, raise issues to Test lead in a timely manner. |
| Test Automation Developer | * Develops automated test scripts from test cases provided by the test lead. * Develops and executes performance and security testing as required. |

Table 17 - Test Roles and Responsibilities

## Matrix Engaging Test Phase, Test Activity, and Project Roles

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Phase | Environment | BA7 Platform | Test Planning | Test Plan Review & Sign-off | Test Data | Test Execution | Test Execution Sign-off |
| Unit | DEV | Client Server | Programmer/ Analyst | Programmer/ Analyst | Programmer/ Analyst | Programmer/ Analyst | Programmer or Analyst Lead |
| System Integration Testing | QA | Client Server | Testing Team | Testing Team | Testing Team | Testing Team | Testing Team |
| Non-Functional Testing | QA | Client Server | Testing Team | Testing Team | Testing Team | Testing Team | Testing Team |
| Non-Functional (Compatibility, Performance and Security testing where ever required)\* | QA | Client Server | Specialized Testing Team | SWAT/ Business Team | Specialized Testing Team | Specialized Testing Team | Specialized Testing Team |
| Regression | QA | Client Server | Testing Team | Testing Team | Testing Team | Testing Team | Testing Team |
| Client Acceptance  (UAT/UAT) | QA | Client Server | Client Team | SWAT/ Business Team | Client Team | Client Team | Client Team |
| Post-Production Verification | PROD | Client Server | N/A | SWAT/  Business Team | Client Team | Client Team | Business Team |

Table 18 - Test Ownership

\*Security Testing/ Performance Testing - Security Testing performed at product level. However in case there is a requirement of performance and security testing at individual client level, then only it will be included as part of non-functional testing scope.

## RACI Model

The RACI model brings structure and clarity to describing the roles that stakeholders play within a project. It is a matrix to clarify responsibilities and accountabilities of certain tasks and ensure that every task in the project is taken care and have an achiever.

**Responsible**: People or stakeholders who are the "doers" of the work. They must complete the task or objective or make the decision. Several people can be jointly responsible.

**Accountable**: Person or stakeholder who is the "owner" of the work. He or she must sign off or approve when the task, objective or decision is complete.

**Consulted**: Stakeholders who need to give input before the work can be done and signed-off on.

**Informed**: Stakeholders who need to be kept "in the picture." They need updates on progress or decisions, but they do not need to be formally consulted, nor do they contribute directly to the task or decision.

### Standard QA Activities and Roles

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **QA Project RACI** | | | | | | Output Work Product |
| QA Project Activities | | | **Test Manager** | **Test Lead** | **Test Analyst** | **Business System Analyst** | **Dev Technical** | **Project Manager**  **(PMO)**  **for Conversion** |
| **Phase 1: Test Initiation** | | | | | | | | | |
| 1. X Company Testing Activities Initiation | | | **R** | **I** |  |  |  | **A** | Request for testing services in the project |
| 2. X Company the Kick off Meeting | | | **A** | **R** |  | **C** |  | **C** | Understanding of project by test team |
| 3. Perform review and analysis of requirements for testability | | | **A** | **R** |  | **C** | **C** | **I** | Test Requirements template |
| 4. Prepare the High Level Scope & Estimation | | | **A** | **R** |  | **C** | **C** | **C** | Test Effort Estimation template (initial estimation on resourcing, effort, schedule and cost for testing) |
| 5. Prepare the High Level Test Schedule | | | **A** | **R** |  | **C** |  | **C** | Test Plan template (initial test schedule aligned to project schedule) |
| **Phase 2: Test Planning (includes UAT/UAT)** | | | | | | | | | |
| 1.Prepare the Detailed Test Estimation | | **A** | | **R** |  | **I** | **I** | **C** | Test Effort Estimation template(test estimates for testing activities) |
| 2.Develop the Test Strategy & Plan for each testing type | | **A** | | **R** |  | **I** | **C** | **I** | Test Strategy & Plan template |
| 3.Prepare the Defect Management Plan | | **A** | | **R** |  | **I** | **I** | **I** | Defect Management Plan template |
| 4. Prepare the Test Plan/Schedule | | **A** | | **R** |  | **C** | **I** | **C** | Detailed Test Plan template (test schedule aligned to project schedule) |
| 5. Plan Test Environment(s) | | **A** | | **R** |  | **I** | **I** | **C** | Plan for test environment setup and procurement of hardware, software, network, and/or tools |
| **Phase 3: Test Design & Preparation (includes UAT/UAT)** | | | | | | | | | |
| 1.Test Scenario Preparation & Review | | **I** | | **A** | **R** | **C** | **C** | **I** | Test Scenario & Test Case Design template, Test Scenario Details tab |
| 2. Test Scenario Defect Reporting, Tracking & Closure | | **A** | | **R** | **I** | **C** | **C** | **I** | Test Scenario & Test Case Design template, Test Summary Details tab |
| 3.Requirement Traceability Matrix Completion | | **A** | | **R** | **C** | **C** | **C** | **I** | Test Requirements template update |
| 4.Test Case Preparation & Review | | **I** | | **A** | **R** | **C** | **C** | **I** | Test Scenario & Test Case Design template, Test Case Details tab |
| 5.Test Environment Setup | | **I** | | **A** | **R** | **I** | **I** | **I** | Environment setup for testing |
| 6.Test Data Setup | | **I** | | **A** | **R** | **I** | **I** | **I** | Test Data Management Process Workflow |
| 7.Requirements Definition | | **A** | | **R** | **C** | **C** | **C** | **I** | Test Scenario & Test Case Design template, Test Scenario Details tab |
| 8.Test Estimation Update | | **A** | | **R** | **C** | **I** | **I** | **I** | Test Effort Estimation template (update) |
| **Phase 4: Test Execution & Monitoring (UAT Execution/ UAT Monitoring)**  **# *Client Facing Team will interact with Client and communicate with UAT defects.*** | | | | | | | | | |
| 1.Test Execution | | **I** | | **A** | **R** | **C** | **I** | **I** | Test Execution results |
| 2.Defect Tracking/Management | | **A** | | **R** | **R** | **C** | **I** | **I** | Test Progress Report template |
| 3.Test Progress Reporting | | **A** | | **R** | **C** | **C** | **I** | **I** | Test Progress Report template |
| 4.Test Summary Reporting | | **A** | | **R** | **C** | **C** | **I** | **I** | Test Summary Report template |
| **Phase 5: Test Closure & Reporting** | | | | | | | | | |
| 1.Test Closure & Reporting | | **A** | | **R** | **C** | **C** | **C** | **C** | Lessons Learned, Final Test Progress and Summary Reports |
| 2.Post Production Defect Analysis | | **A** | | **R** | **C** | **C** | **C** | **C** | Post production defect analysis |
| 3.Post Implementation Review | | **C** | | **C** | **C** | **C** | **C** | **R** | Post Implementation analysis |

Table 19 - RACI

# Appendix

## Test Phase Definitions

The following table lists each test phase along with a description of the phase, entry and exit criteria for the phase. Collectively, these test phases are referred to as validation testing:

|  |  |  |  |
| --- | --- | --- | --- |
| ****Test Type**** | Description | Entrance Criteria | Exit Criteria |
| **Unit Testing**  Owner: <Client Team > | Ensure each developed component correctly implements the design specification.   * Performed against the smallest testable elements of the system including custom programs, configuration, and security profiles * Performed by Developers in a development environment | * Code Complete * Unit Test Cases written, reviewed and ready for execution | * Unit testing passed * Bugs found are reported, fixed and closed * Test results produced and approved by Development Lead / Project Lead |
| **System Integration Testing (SIT)**  Owner: < Testing Team > | Ensure functionality operates as defined in the requirements.   * Test cases written for each function, user interface component, and work flow * Includes positive, negative, and boundary conditions * Performed by Test team in test environment | * Successful completion and validation of developer unit and integration testing * Test cases written, reviewed and ready for execution * **Customer sign off of requirements, Test Cases and TFS features** | * Completion of all functional test cases. * No open critical defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer |
| **Regression Testing**  Owner: < Testing Team > | Ensure base functionality continues to function as expected after changes made to support new functionality.   * Re-run regression test cases * Performed by Test team in test environment | * Successful completion and validation of functional testing * Test cases written, reviewed and ready for execution * **Customer sign off of requirements, Test Cases and TFS features** | * Completion of all manual and automated regression tests. * No open critical defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer |
| **Non-Functional Testing- Compatibility Testing**  Owner: < Testing Team > | Non-functional testing X Companied on the application to evaluate the application's compatibility with the computing environment such as operating systems, internet browsers, data bases and so on. | * Successful completion and validation of functional, regression, and integration testing * Test cases written, reviewed and ready for execution * Customer sign off of requirements, Test Cases and **TFS features** | * Completion of all Compatibility test cases with no open critical findings |
| **Non-Functional- Security / Vulnerability Testing**  Owner: < Specialized Testing Team > | Ensure the solution complies with access control requirements of X Company as well as legal and legislative regulations.  Identify and communicate level of risk associated with the confidentiality, integrity, and availability of the data and assets the application protects.  Uses CVSS to score application vulnerability risk. | * **SAST Entry criteria:** Code has been checked into source code repository and contains all library references required for the code to compile successfully * **DAST Entry criteria:** Code has been deployed to UAT or other pre-production environment and one or more user accounts have been provided. | * Completion of appropriate Dynamic Application Security Tests (DAST) and Static Application Security Tests (SAST) with a CVSS score within acceptable limits. |
| **Non-Functional- Security / Vulnerability Testing**  Owner: < Specialized Testing Team > | **FOSS:**   * Identifying any open source software contained in an externally distributed product * Reviewing and approving the intended use of FOSS * Satisfying FOSS license obligations | * Source code containing packages, libraries etc. in precompiled form needs to be shared via secure medium * Application Distribution model needs to be communicated | * Complaint with all open source license obligations and vulnerability free open-source components in X Company products |
| **Non-Functional - Performance Testing**  Owner: < Specialized Testing Team > | Ensure the solution meets the response or processing times defined in the specifications, under the data flow and connection volumes expected.   * Stress Testing: Establish the limits of an application or infrastructure. Identify the “breaking point” of the application in terms of throughput, number of users, or other criteria. * Load Testing: Determine how system will perform under peak usage and normal usage. Identify the throughput, number of users, or other criteria under which the system will operate effectively. * Performed by Test team in test environment | * Successful completion and validation of developer testing. * Test cases written, reviewed and ready for execution * Customer sign off of requirements, Test Cases and **TFS features**   Note: Performance tests could be run during development, once application is stable, to get an early indication of issues. | * Completion of all performance test cases and the system response and processing time meets specifications as documented in the requirements. * No open critical defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer |
| **Smoke Testing**  Owner: Dev Team | A non-exhaustive software test meant to ascertain that the most crucial functions of a program work, before taking the build for more rigorous testing.  Executed when solution handed off from development to the initial testing phases and again when solution released from test to UAT and production environments | * Upon any successful build. | * Successful completion of test cases * No open critical data defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer |
| **User Acceptance Testing (UAT) / Client Acceptance Testing (UAT)**  Owner: Client Team | Ensure the end to end functionality of the system operates as expected in an environment that closely resembles production.   * These tests ensure that the system as a whole functions properly. * Performed by operational or business support teams in UAT environment. | * Successful completion and validation of functional, regression, and integration testing. * Customer sign off of requirements, Test Cases and **TFS features**. | * Completion of all data validation tests. * No open critical data defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer. |
| **Post Production Validation (PPV)**  Owner: Client Team | Ensure the critical business functions of the operation are supported by the system.   * Validate system is ready for production. * Execute high level functions that characterize the operation; validate the system’s availability, functionality, and accuracy. * Owner: Performed by client, operational and/or business support teams in the production environment | * Successful completion UAT | * Completion of all data validation tests. * No open critical data defects, unless agreed by customer. * No open major defects above the threshold agreed with the customer. |

Table 20 - Test Phase Definitions