**e\_Treatment Clinical System**

**Software** **QA Test Plan**

**Version: 1.0**

**Date:12/12/2022**

**Prepared By: Group 5**

[1 Document Acceptance and Sign-Off 3](#_Toc121930330)

[2 Revision History 3](#_Toc121930331)

[3 Glossary 4](#_Toc121930332)

[4 Introduction 5](#_Toc121930333)

[5 Testing Objectives 5](#_Toc121930334)

[6 Test Strategy 9](#_Toc121930335)

[6.1 Unit and SIT Testing 9](#_Toc121930336)

[6.2 System Testing and Integration Testing 10](#_Toc121930337)

[6.3 Performance and Stress Testing 10](#_Toc121930338)

[6.4 User Acceptance Testing 11](#_Toc121930339)

[6.5 Automated Regression Testing 11](#_Toc121930340)

[7 Test Environments 12](#_Toc121930341)

[7.1 Test Data 13](#_Toc121930342)

[8 Roles and Responsibilities 13](#_Toc121930343)

[9 Monitoring Testing 14](#_Toc121930344)

[10 Entry and Exit Criteria 15](#_Toc121930345)

[11 In Scope and Out of Scope 16](#_Toc121930346)

[12 Risk/Assumptions and Dependencies 17](#_Toc121930347)

[12.1 Risks/Assumptions 17](#_Toc121930348)

[12.2 Dependencies 18](#_Toc121930349)

[13 Test Schedule 18](#_Toc121930350)

[14 Testing Tools Requirements 19](#_Toc121930351)

[15 Defect Life Cycle Management 19](#_Toc121930352)

[16 Test Deliverables 21](#_Toc121930353)

[17 Appendix 23](#_Toc121930354)

[18 References 23](#_Toc121930355)

# Document Acceptance and Sign-Off

By signing below, I acknowledge that I have read the entire contents of this document and accept the document in this form as reasonably fulfilling the goals described in the section titled Document Purpose. I further agree that this will constitute the document of record and cannot be changed without review and acknowledgement of the groups shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Group / Role** | **Approver Name** | **Approver Signature** | **Date Approved** |
| Group 5/ QA | Rashmi Singh | RS | 12/14/2022 |
| Group 5/QA | Rajiv Ranjan Sahu | Rajiv Sahu | 12/14/2022 |
| Group 5/QA | Divya Patil | Divya P | 12/14/2022 |
| Group 5/QA | Lokesh Balaji | LB | 12/14/2022 |
| Group 5/QA | Sarvesh Sathishkumar | SS | 12/14/2022 |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Document/Department Editor: | | | |
| **Date** | **Revision #** | **Editor** | **Description of Change** |
| 12/08/2022 | eTreatment\_v1 | Rashmi & Lokesh | Incorporated Test Condition Matrix for the e\_Treatment Clinical System V5.0. |
| 12/09/2022 | eTreatment\_v2 | Divya & Sarvesh | Incorporated Test Scenarios/Cases |
| 12/10/2022 | eTreatment\_v3 | Rajiv Ranjan Sahu | Incorporated Traceability Matrix |
| 12/11/2022 | eTreatment\_v4 | Rajiv Ranjan Sahu | Incorporated Introduction & Testing objectives |
| 12/11/2022 | eTreatment\_v5 | Rashmi Singh | Incorporated Test Strategy & Environments details |
| 12/12/2022 | eTreatment\_v6 | Lokesh Balaji | Added details for section 8, 9 & 10 |
| 12/12/2022 | eTreatment\_v7 | Divya Patil | Test Schedule, Testing Tool Requirements, Defect Life Cycle management |
| 12/12/2022 | eTreatment\_v7 | Sarvesh Sathishkumar | Added details for section 11,12 & 16 |

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Application | A computer software program that performs a special task |
| Medicare | A federal health insurance program for senior citizen and permanently disabled people |
| Medicaid | A joint and state program that helps low-income families and individuals pay for the cost associated with medical care |
| Medco | Offers comprehensive and affordable insurance plans that minimize coverage gaps to keep your financial foundation strong. |
| United Care Insurance | Health plans are offered by United Healthcare Insurance Company and affiliates. They work with federal and state agencies to provide government-sponsored health insurance. |
| Claim | A request to your health insurance company to pay a bill for health care service |
| Spectra Labs | A lab that delivers testing, analysis and reporting with the reliability and personal service you require to ensure the accurate results possible for your patients |
| Mot-Ran, Xer-Ran, Per-Gan | Drugs used to treat patients. |
| Hemo Dialysis | Procedure where a dialysis and a special filter is used to purify the blood. |
| Peritoneal Dialysis | Procedure to remove unwanted and vestigial products from blood. |

# Introduction

A new e\_Treatment Clinicals system is replacing the legacy TSS application. In the e\_Treatment Clinicals System, patients will get admitted and treated based on their Financial and Clinical Clearance flags (Admit Flag and Treatment Flag), and the Insurance types. Once admitted, the patients are treated based on the Treatment Flag (Yes). During treatments, all the patients' Lab Orders will be sent to the Spectra System Lab and the Lab Results are sent to the e\_Treatment Clinical System. After the end of the treatments, all treatments are collected in the e\_Treatment Clinical System. The e\_Treatment Clinical System will send all Treatment and Lab Charges to the e\_Finance system nightly. The e\_Finance system will create claims and sends them to the insurance companies for getting paid accordingly. The insurance companies then send the payments to the Archival\_Trust Company. The Archival\_Trust Company does a weekly reconciliation with the e\_Finance System for auditing purposes.

# Testing Objectives

The initial phase will include all ‘must have’ requirements aligning with business requirements and functional specifications. This System traverses through patient admission, treatments, financial charges, financial billing, and clinical reports.

**Major new or modified capabilities included in this test**

1. **Patient Admission**

* New and Existing patients can be admitted into the e\_TreatmentSystems.
* The patients are either Financially Cleared or Not Cleared. If the Admit\_Flag is Yes, the patients will be admitted otherwise the patients will be rejected.
* Once the patients get admitted, their demographic information will be sent to the e\_Finance System, the e\_Filing System, and the Spectra System Lab
* For all New patients, new admission folders will be created in the e\_Finance System, the e\_Filing System, and the Spectra System Lab.
* For all Existing patients, the existing folders get updated in the e\_Finance System, the e\_Filing System, and the Spectra System Lab.
* For all New and Existing patients, the e\_Filing system sends their insurance information to the proper insurance company for verification purposes (e.g., if Medicare, to Medicare/Medicaid, etc.)
* If the insurance type is anything different than Medicare/Medicaid, Medco, or United Care, the admission will fail, and the patients will get rejected. In this case, no demographic info will be sent to any system.

1. **Patient Treatments**

* For all the new and existing patients, the patient will get treated only if the Treatment\_Flags is set to Yes otherwise the patient will not get treated (Treatment\_Flags = Yes or No).
* There are 2 treatment locations - At\_Home and In\_Center.
* At\_Home Treatment Location has 2 Modalities - The patients are treated either using Hemo Dialysis under the insurer Medicare/Medicaid with the drug being Mot-Ran or Peritoneal Dialysis and the insurer is a Medco with the drug being Xer-Ran.
* Under In\_Center Treatment the patients are treated using Palliative Dialysis and the insurer is United Care with the drug being Per-Gan.

1. **Treatment and the Lab Charges:**

* For all the new and existing patients, patient Treatments will get stored in the e\_Treatment System’s database after each treatment.
* The patients’ Lab Orders are sent to the Spectra System Lab. The Spectra System lab processes the Lab Orders and sends the Lab Results to the e\_Treatment Clinical System
* All the Treatment and Lab Result Charges are gathered daily in the e\_Treatment Clinical System’s database which sends the Treatment and the Lab Result Charges for all patients to the e\_Finance system for billing purposes.

1. **Financial Billing**

* For all the patients the e\_Finance System sends all claims for the Treatment and Lab Charges to the respective Insurance companies, Medicare/Medicaid, Medco, or United Care for all the treated patients.
* The 3 Insurance companies, Medicare/Medicaid, Medco, and United Care, send their payments to the Archival\_Trust Company based on the claims they received.

1. **Reconciliation**

* The Archival\_Trust Company does a weekly reconciliation with the e\_Finance System for auditing purposes.

**Differences between current release and the previous phase**

|  |  |  |
| --- | --- | --- |
| **Features/Functionalities** | **e\_Treatment Clinical System** | **TSS Clinical System** |
| Patient Information System (e\_Finance System, the e\_Filing System, and the Spectra System Lab) | R | Q |
| Lab Order & Results (Spectra System Lab 🡪 e\_Treatment Clinical) | R | Q |
| Treatment & Lab Result Charges (e\_Treatment Clinical 🡪 e\_Finance system) | R | Q |
| Insurance – Medicare/Medicaid, Medco, and United Care | R | Q |
| Reconciliation 🡪 Archival\_Trust Company | R | Q |

**Details of the release undergoing testing :**

1. **Interfaces**

* **Patient Admission**:
* The admission of the patient is dependent on the financial clearance and their insurance company.
* Patient admission interfaces with the e\_Finance System, the e\_Filing System, and the Spectra System Lab if Admit\_Flag = Yes
* **Patient Treatment**:
* The treatment of the patient is dependent on the Treatment\_Flags (Yes/No)
* Patient treatment interfaces with treatment location which in turn are interfaced with treatment Modalities along with Insurance and the drug.
* **Treatment and the Lab Charges**:
* Lab Orders are interfaced with Spectra System Lab and the lab results are interfaced with the e\_Treatment Clinical System.
* **Financial Billing & Reconciliation:**
* Treatment and Lab Charges interface with the e\_Finance system. e\_Finance system is interfaced to the 3 Insurance companies which is turn are interfaced to the Archival\_Trust Company for reconciliation.

1. **Roles & Responsibility:**

* **Patient Admission:** Gets admitted based on the Admit\_Flag and insurance company.
* **Patient Treatment:** Patients are treated, and lab orders are sent to Spectra System Lab.
* **Spectra System Lab:** It is responsible for processing the Lab Orders and sending the Lab Results to the e\_Treatment Clinical System
* **e\_Treatment Clinical System:** It is responsible for sending the Treatment and the Lab Result charges for all patients to the e\_Finance system for billing purposes
* **e\_Finance system:** e\_Finance System sends all claims for the Treatment and Lab Charges to the respective Insurance companies, Medicare/Medicaid, Medco, or United Care for all the treated patients.
* **Insurance companies:** The 3 Insurance companies, Medicare/Medicaid, Medco, and United Care, send their payments to the Archival\_Trust Company based on the claims they received.
* **Patient Demographics:** The patient’s information is stored in various systems like OnBase System, eProcessing system and Avalon Systems Lab
* **The Archival\_Trust Company:** It is responsible for weekly reconciliation with the e\_Finance System for auditing purposes.

**Context diagram of product within its environment showing interfaces**

Diagram

Description automatically generated

**Hardware (Physical/VM/Cloud):**

* CPU: Intel i7
* RAM: 16 GB
* Storage: 1TB
* Internet: LAN/Wi-Fi/VPN

**Software**

* OS: Windows 7,10,11, Mac OSX
* Browsers: Google Chrome, Firefox, Safari, Edge
* Security: Norton 360, McAfee Total Protection
* Software: Selenium, UFT, JMeter, LoadRunner, ALM

**Risks**

* **Data security:** The e\_Treatment Clinicals system has various information regarding patient’s demographics, Lab results and End Treatments which needs to be secured.
* The applications need to be running 24/7 as it is a health care system.

**Environment requirement**

* Environment that will be used for testing should be stable and should have configurations like that of the production environment.

# Test Strategy

This section gives the overall details of the test effort and the approach on how the testing will be accomplished. This helps in communicating to the project team and the customers/users the scope of testing.

## Unit and SIT Testing

**Unit Testing:** Individual units or components of the software are tested. The purpose is to validate that each unit of the software code performs as expected.

Unit Testing is done during the development (coding phase) of an application by the developers.

**SIT Testing:** System Integration Testing (SIT) is performed to verify the interactions between the modules of a software system. It deals with the verification of the high and low-level software requirements specified in the Software Requirements Specification/Data and the Software Design Document.

It is conducted after unit testing and will be done each time a new module is added to the system. This is performed by testers and developers.

## System Testing and Integration Testing

**System Testing involves testing the** **complete system.** All the modules/components are integrated to verify if the system works as expected or not.

System Testing is done **after Integration Testing.** This plays a vital role in delivering a high-quality product. System testing is done by the developers and testers.

**Integration Testing:** In this testing, software modules are integrated logically and tested as a group. This testing exposes defects in the interaction between these software modules when they are integrated.

After Unit testing and before System testing, Integration testing is performed. QA is responsible for integration testing.

## Performance and Stress Testing

**Performance Testing:** Performance Testing is used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under a particular workload. The main purpose of this testing is to identify and eliminate the performance bottlenecks in the software application.

Yes, performance testing is included as e\_Treatment Clinical System is a healthcare system, and performance testing will help us to ensure the quality, reliability, performance, safety, and efficiency of the application.

Monitoring of the application performance is crucial when various patients are trying to register for the treatment at the same time. During the admission procedure, the patient admission module should always be available (stable). By loading many patient records, the system is put to the load test.

The healthcare providers (Doctors/Hospitals) provide care 24/7, so the patient check-in software needs to be always available. Also, it needs to communicate with insurance companies to validate policy information and send claims.

**Stress Testing:** The goal of Stress testing is to measure software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn’t crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

Stress testing is required for the e\_Treatment Clinical System to ensure that the system does not crash if there is sudden increase in number of patients. It will help us interpret the system's performance during its failures and get to know the behavior of the software.

## User Acceptance Testing

User Acceptance Testing (UAT), also known as beta or end-user testing, is carried out by the user or client to determine whether it can be accepted or not.

The main purpose of this testing is to validate the software against business requirements.

This is typically the last step before the product goes live or before the delivery of the product is accepted. This is performed after the product itself is thoroughly tested i.e., after System Testing.

The UAT team is responsible for User Acceptance testing. The team can be comprised of beta testers, or the customer should select UAT members internally from every group of the organization so that every user role can be tested accordingly.

* The UAT process is extensively focused on developing quality products and seeking feedback regularly from customers. QA teams here must ensure that each method and strategy is executed to improve the quality and usability of the product while making it user-friendly.
* QA teams must streamline the testing processes that define the outcome of UAT and leverage the UAT execution with far better efficiency.
* Testers need to work on iterating the product functions based on improving the quality of a product. Through various inspections and process checklists, QA teams can improvise their processes and contribute to developing a top-quality product.

## Automated Regression Testing

Automated regression testing is a technique for ensuring that new software upgrades do not break the functioning of an existing software application. It includes re-running functional and non-functional tests to ensure that a software application works as intended after any code changes, updates, revisions, improvements, or optimizations.

Yes, automation regression testing is required in this process. This would help us to:

* An effective regression strategy saves organizations both time and money.
* The greatest benefit of automated regression testing is that it frees up resources.
* With an automation tool that creates visual recording and logs, we can easily go in and detect why tests fail. These speeds up feedback loops, and bug fixes.
* Automated regression tests can run 24/7, 365 days a week.
* With the ability to test faster comes the ability to test more.
* As new features are added to a product, new tests must follow, causing the regression suite to grow over time. Eventually, it’ll reach a point where it’s no longer feasible to manage the regression test cases manually. Automated regression testing will make it easy for us to keep a clear overview of our testing suite and allow us to adjust and maintain it without requiring us to rewrite or change code.

# Test Environments

Testing environment is an arrangement of all the required software and hardware settings essential for testing. The setting of the test environment depends upon the requirement or the build of the applications that needs to be tested.

The test environment requires setting up of various number of distinct areas like: -

* **Setup of Test Server** – Each test may or may not be executed on the local machine itself. Hence, we need to establish a test server.
* **Setting up a network** – Setting up of Internet, LAN/WiFi
* **Test PC setup**
* **Creating Test Data for the Test Environment**
* **Check the Bug Reporting** Tools

|  |  |
| --- | --- |
| **Test Type** | **Environment** |
| SIT Testing | SIT |
| System Testing | ST |
| Performance Testing | Pre- Prod (Env similar to Production) |
| User Acceptance Testing | Staging |

All production deployments follow a path from the Development to Production and beyond from the environment’s perspective.

**DEVEVELOPMENT -> QA -> STAGING -> PRODUCTION**

Graphical user interface, text, application, chat or text message

Description automatically generated

## Test Data

Like test environment, test data will depend upon the application that is tested by the QA. After carefully reviewing the requirements of the application, the testing team can create appropriate test cases and data based upon the test scenarios which are to be tested.

Test Data can be generated –

* Manually
* Mass copy of data from production to testing environment
* Mass copy of test data from legacy client systems
* Automated Test Data Generation Tools

Test data that would be required are:

* URL of the application/ Desktop application installed on the system if it a desktop application
* Username & Password to login to the system

# Roles and Responsibilities

**Tester**

* The tester gathers the information related to the testing and tests the software at various levels and records the result.
* Working with software developers and project support teams.
* meet with system users to understand the scope of projects.
* The tester is also responsible for tracking the defects and reporting them.

**QA Lead**

* Supervises the entire team and software testing processes - developing testing strategy, planning the use of technical resources, and performing quality control.
* Sometimes they can act as a negotiator with the client, in order to bring all his wishes to the project team.

**Test Manager**

* In addition to preparing test strategies, test budgets and test cycles, the test manager is also responsible for evaluating the test outcome.
* It is also the responsibility of the Testing Manager to guide the testing teams in designing testing approaches and techniques.

**Test Leaders**

* Responsibilities of Test leaders tend to include involvement in the planning, monitoring, and control of the testing activities and tasks.
* The test Leaders are also responsible for assigning tasks to the testers and reviewing the test documents and preparing test plans at each level according to the objective.

**Test Architect**

* This specialist conducts project analysis and develops a global product validation system.
* He/she must have experience performing automated testing and must understand software architecture and system design.

**QA Analyst**

* This position is kind of a combination of software testing skills with the skills to build long-term web product validation strategies.
* The QA Analyst position includes work with business logic and matches the target audience.

**Test Automation Engineer.**

* They are responsible for creating test scripts, which can be used to test the operation of the system and implement an automation environment.

**Manual Test Engineer**

* Manual Test Engineer gets to know the product manually.
* He/She mostly works with the client part of the software. Web software is tested from the point of view of the end user group, to find the maximum number of bugs that should not get into the release version and to users.

**UI/UX Test Engineer**

* Some companies make a UI/UX department in order to find the best user experience.

# 9 Monitoring Testing

Test Monitoring is the process of evaluating and providing feedback on the test proceedings that are currently in progress.

The most used **Test monitoring metrics include**:

* Test Coverage Metrics
* Test Execution Metrics (Number of test cases pass, fail, blocked, on hold)
* Requirement Traceability Metrics
* Percentage of scheduled test cases prepared – Percentage of test environment preparation finished (including tests executed or not executed, failed, or passed)
* Defect information (discovered, resolved or unresolved, failure rate, retest outcomes)
* Test Deadlines
* Test project cost, based on a cost-benefit analysis of detecting the next problem or running the next test.
* Requirement Tracking
* Consumption of Resources in the Project.

**Test Monitoring includes:**

* Track and report defect status as per the defect management process.
* Establish, document, and publish test status reports as defined by implemented TS test methodology.
* Testing schedule slippage recovery plans (including code rework) for testing should be considered and documented accordingly.
* Ensure that any approved changes can be incorporated into the testing, referencing the change control process as appropriate.
* Schedule test meetings for monitoring progress as appropriate.

# 10 Entry and Exit Criteria

**Entry Criteria** gives the prerequisite items that must be completed before testing can begin.

* Test plan should be appropriately defined and approved.
* QA has a thorough understanding of the functional requirements.
* Readiness of test cases for testing each step of the process must be there.
* Appropriately requirements regarding Patient Admission, Patient Treatments, Treatment and Lab Charges and Financial Billing should be defined and approved.
* Partial or Complete code for each requirement must be available for testing.
* Patients admitted, Patient’s treatment type, charges for each patient, types of treatment, and financial information about each patient’s data must be available and accessible for testing.
* Test Environment should be set up with necessary resources like tools and devices.
* Tester should have sound knowledge of the test environment and the testing tools.

**Exit Criteria** defines the items that must be completed before testing can be concluded.

* Achieving complete coverage of each functional requirement.
* Spotting and fixing the high priority defects and errors. For example, errors in charges, billings and transferring the lab charges should be fixed.
* Fixing all the “Show Stopper Defects” or “Blockers” which are blocking the final functioning of the application and making sure that no critical defects and errors are in open status.
* Retesting and closing all the high priority defects and errors to execute the regression scenarios successfully.
* All the critical test cases for admission of patient, treatment of patients, charges and billing are passed.

# 11 In Scope and Out of Scope

The **In-Scope** functions that will be tested:

* Check Patient Admission & Treatment based on admit flags
* Check the Treatment location for each patient
* Check Test Modalities based on Location
* Check Insurance for different Modalities
* Check Drug type based on location, insurance & modalities
* Check if Spectra System lab sends the Lab Results to the e\_Treatment Clinical System
* Check if Lab Result Charges for all patients are sent to the e\_Finance system
* Check if e\_Finance system sends all claims to the respective Insurance companies
* Check if Archival\_Trust does reconciliation with the e\_Finance System

The **Out-of-Scope** functions of the application:

* Maximizing Patients Benefits
* Improvisations on machines for Home and In-Clinic treatment
* Checking Annual Agreements with Insurance Companies
* Payment terms and condition with Insurance Companies
* Validation of the total claims received by the Holding Trust Company

# 12 Risk/Assumptions and Dependencies

## 12.1 Risks/Assumptions

|  |  |
| --- | --- |
| **Risk** | **Mitigation** |
| The test environment should be stable | All major codes fixes should have been delivered before the start of the testing. |
| All the home treatment equipment’s gives unaltered output | Verifying with the factual data of other in clinic patients |
| Transfer of the bills from e-treatment to financial system will be done flawlessly | Verification of before and after numbers of the bills |
| Lack of resources | By outsourcing resources, the costs can be reduced, and the requirements can be met |
| Testing schedules are met | There will be a dedicated product manager who will keep track of the progress and manage the schedules |
| Corruption of files | Backup of all the files should be taken |
| Teamwork and communication | To track the progress of the team, stand-ups should be conducted everyday |

|  |  |
| --- | --- |
| **Assumptions** | **Mitigation** |
| Production data will be available for seeding the test database before testing. | Ensure production data is backed up and restored in the Test DB. |
| Patient details should be proper | Accurate patient records |

## 12.2 Dependencies

|  |  |
| --- | --- |
| **Dependencies** | **Comments** |
| Test Data Availability | Test data & database should also be made available to the testers for use during testing. |
| Software Availability | Selenium, E-treatment system, JIRA, Load Runner |
| Resource Availability | Testing must be conducted with an adequate number of resources |
| Budget Constraint | Testing should be funded adequately |

# 13 Test Schedule

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Start Date** | **End Date** |
| Design and Functional Requirements Review | May 1, 2022 | June. 2, 2022 |
| User Story Reviews | September 12, 2022 | September 19, 2022 |
| Sprint 1 to Sprint 5 | October 10, 2022 | February 28, 2023 |
| Test Plan Review | October 10, 2022 | - |
| Test Cases Review, Before each Sprint | October 10, 2022 | October 30, 2022 |
| Performance & Load Testing, | January 15, 2023 | February 28, 2023 |
| Pilot Release | April 1, 2023 | - |
| Full Deployment | May 1, 2023 | - |
| Project Retrospective | June 1, 2023 | - |

# Testing Tools Requirements

|  |  |
| --- | --- |
| **Tool** | **Objective** |
| Load runner & Performance Centre | Performance testing |
| Selenium, UFT | Automation tool |
| ALM, JIRA | Defect Management tool |
| Junit, NUnit | Unit testing tool |
| Citrus, Tessy | Integration testing tools |
| Test Management Tools | TestRail, Test pad |

# Defect Life Cycle Management

A defect is an error in coding or logic that causes a program to malfunction or to produce incorrect/unexpected results.

1. **Defect Fields**

|  |  |
| --- | --- |
| **Defect Fields** | **Defect Fields Description** |
| Defect ID | It’s a unique identification number |
| Reported Date | This is the date when a defect/bug is reported. |
| Reported By | It contains information about the individual who reported the defect, such as their name and ID number. |
| Status | Defect statuses include New, Assigned, Open, Retest, Verification, Closed, Failed, and Deferred, etc |
| Version Found In | It’s an Application’s Product version where the defect is found |
| Defect Description | It includes a detailed step-by-step description of the defect |
| Fixed by | It contains info about the developer who fixed the defect. |
| Date Closed | The date on which the defect was resolved. |
| Severity | Shows the impact of the defect or bug in the software application. |
| Priority | The order in which defects are fixed - High, Medium, and Low. |

1. **Defect Life Cycle**

Diagram

Description automatically generated

1. **Defect Severity:**

|  |  |
| --- | --- |
| **Defect Severity** | **Defect Description** |
| Critical | This defect indicates complete shut-down of the process, nothing can proceed further |
| Major | It is a highly severe defect and collapses the system. However, certain parts of the system remain functional |
| Medium | It causes some undesirable behaviour, but the system is still functional |
| Low | It won’t cause any major break-down of the system |

1. **Defect Priority:**

|  |  |
| --- | --- |
| **Defect Priority** | **Defect Description** |
| Low | The Defect is an irritant, but repair can be done once the more serious Defect has been fixed |
| Medium | During the normal course of the development activities defect should be resolved. It can wait until a new version is created |
| High | The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed |

# Test Deliverables

**Test Scenarios/Test Cases:** The Test Scenarios/Cases can be found in [Appendix](#_Appendix).

**Test Coverage Report and the Traceability Matrix:** Listing the mapping of the requirements to the test cases to ensure full coverage.Test Coverage Report and the Traceability Matrix can be found in[Appendix](#_Appendix).

**Test Status Report:** The list of tests that were executed and the status of the testing, including Defect report. Test status report can be found in[Appendix](#_Appendix).

**Change Control and Change Management**:

1. **Change Control & its implementation during testing:**

Change Control is the process that a company uses to **document, identify, and authorize changes**to an IT environment. It reduces the chances of unauthorized alterations, disruption, and errors in the system.

Whenever any **new or different changes are requested for the system**, especially by stakeholders, it is neither optional nor ignorable. It has to be implemented without affecting other components of the system. This is when the change control comes handy. It **helps** project teams to **modify the scope of the project using specified controls and policies**. Change Control is practiced whenever a project is not progressing as planned.

It is mandatory that a formal document for change request is completed and reviewed in order to keep control of change requests.

**Process of Change Control**

Diagram

Description automatically generated

1. **Change Management Process:**

It is the quality control process that sets the stage ready by assessing, planning, and getting the right approvals for deployment of one or multiple changes to ensure minimal disruption to live environment.

Diagram

Description automatically generated

# Appendix

|  |  |
| --- | --- |
| **Document** | **Links** |
| Test Scenarios/Cases | <https://northeastern-my.sharepoint.com/:x:/g/personal/singh_ras_northeastern_edu/EQVnDnomAWlNl0S6ImsUntgBq2s5gXqq3v1fz3rVMDNR2w> |
| Traceability Matrix | <https://northeastern-my.sharepoint.com/:x:/g/personal/singh_ras_northeastern_edu/EQVnDnomAWlNl0S6ImsUntgBq2s5gXqq3v1fz3rVMDNR2w> |
| Test status report | <https://northeastern-my.sharepoint.com/personal/singh_ras_northeastern_edu/Documents/Test%20Status%20Report.xlsx?web=1> |

# References

* <https://www.softwaretestinghelp.com/types-of-software-testing/>
* <https://www.guru99.com/defect-life-cycle.html>
* <https://www.techtarget.com/searchcio/definition/change-management>
* <https://www.comeet.com/resources/job-descriptions/qa-tester>
* <https://northeastern.instructure.com/courses/124324/assignments/1635577>