

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Document Version** | **Description** |
| 21-03-2019 | 1.0 | Initial Draft |

Contents

[**Purpose of this Document** 4](#_Toc5009679)

[**1.** **Introduction** 4](#_Toc5009680)

[**2.** **Testing Objective** 4](#_Toc5009681)

[**3.** **Testing Scope** 5](#_Toc5009682)

[**4.** **Test Strategy** 5](#_Toc5009685)

[**4.1** **Testing Stages** 5](#_Toc5009686)

[4.1.1 **System Test** 5](#_Toc5009687)

[4.1.2 **User Acceptance Testing** 8](#_Toc5009695)

[4.1.3 **System Performance Testing** 11](#_Toc5009703)

[**5.** **Test Environment Setup & Build Strategy** 12](#_Toc5009707)

[**5.1** **Test Environment Setup** 12](#_Toc5009708)

[**5.2** **Build Strategy** 13](#_Toc5009709)

[**6.** **Testing Tools** 13](#_Toc5009710)

[**7.** **Defect Management** 13](#_Toc5009711)

[**7.1** **Log, Analyze and Assign defect** 13](#_Toc5009712)

[**7.2** **Defect State** 14](#_Toc5009713)

[**7.3** **Defect Life Cycle** 15](#_Toc5009714)

[**7.4** **Description of the main Defect Life Cycle Statuses** 16](#_Toc5009715)

[**7.5** **Defect Severity and Priority** 16](#_Toc5009716)

[**8.** **Test Schedule** 17](#_Toc5009717)

[**9.** **Risk Contingencies** 18](#_Toc5009718)

[**9.1** **Risks and Mitigation Plans** 18](#_Toc5009719)

[**10.** **Definitions and Acronyms** 19](#_Toc5009720)

# **Purpose of this Document**

This test plan provides a detailed overview of the approach that will be adopted for the testing of CM project. This document will be used as a primary resource by all test professionals to identify what needs to be tested, how to test, the testing process and references that will be needed in order to successfully test the initiative. The sign-off of this document indicates that the signing reviewers approve the stated test plan for CM project. If changes to this plan are required, they will need to be agreed upon and signed off by those identified as signatories in this document.

This document will allow all Department including business, technical and program management, to agree to the proposed testing activities of CM and deliverables prior to the commencement of test activities. Points of contention must be discussed, and refinements agreed by the stakeholders affected before CM test commences.

However, this test plan will be updated and re-issued in the event of a significant re-plan or major scope changes to the project.

The purpose of this document is to:

• Describe the high-level objectives

• Outline the scope of the components required

• Describe the testing approach

• Define the high-level entry and exit criteria for each test phase

• Describe the high-level test responsibilities for the testing team

• Outline the resources, deliverables, risks and tools required to accomplish test success

# **Introduction**

The purpose of this document is to define the test plan that will be adopted for testing within the CM project. This document defines a plan to be implemented for testing the software deliverables, identified in the Sprint 9 project’s scope.

# **Testing Objective**

The primary objective of testing of Case Management is as follows:

1. To verify the incorporation of the current Sprint requirement, as per the application deployment

2. To identify the functional & non-functional requirements for the Sprint and execute the corresponding regression test suite on them.

3. To ensure that data persists in DB.

4. To ensure that under no circumstances, the system degrades or breaks down.

5. To ensure that system meets the user acceptance criteria

Our approach is a hybrid of water fall and agile. Testing activities will start parallel with Dev. phases. Once all the functionalities are completed through design and development, there will be system integration test phase, user acceptance test phase followed by production deployment per state.

# **Testing Scope**

The scope of testing includes the following phases:

* System testing
* Automation Testing
* Performance Testing

|  |  |  |
| --- | --- | --- |
| **S.No** | **Module** | **Feature** |
| 1 | Census Integration with CM |  |
| 2 | HRA | Add HRA |
| 3 |  | View List of HRA |
| 4 | Performance Testing |  |

*Note* – *Testing is being done in Google Chrome only.*

As part of Case Management project, following initiatives will be validated against the agreed requirement:

## **Backlogs**

* Consume data migrated from Census for processing Referral Request from Hospitals
* As a user, I should be able to add HRA for a Member during case management review
* As a user, I should be able to see the list of HRA added for the user

## **Bugs**

* Search member result count is not getting displayed.
* On Scroll members are not coming in the search member result.
* Global queue should not get display in side menu for case managers.
* "Page cannot be displayed" error is getting displayed if user tries to login with Swapna credentials.

# **Test Strategy**

## **Testing Stages**

An overview of the testing stages to be followed is provided in this section:

Test Stages and Responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| **Stages** | **When** | **Environment** | **Responsibility** |
| System Testing (includes performance testing) | After the completion of construction and unit testing | Test | CM |

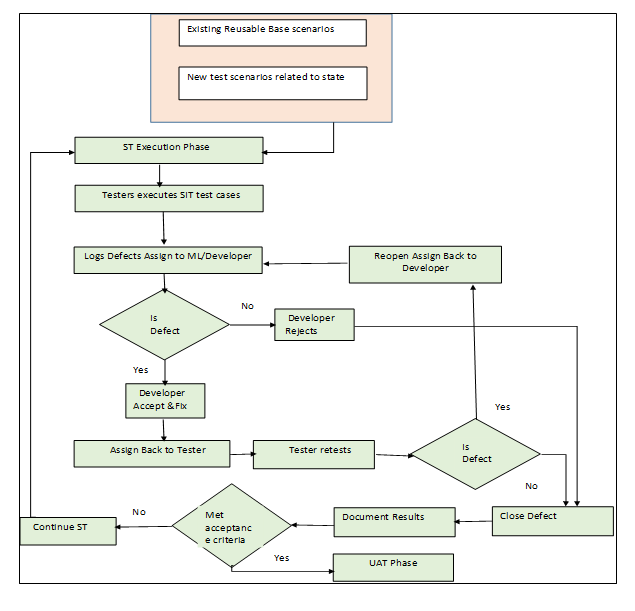
## **System Test**

The objective of system testing is to ensure that the system meets its expected behaviour as per requirement and demonstrate that the CM Team is working as per the signed-off gap documents.

## **Entry Criteria**

|  |  |
| --- | --- |
| **#** | **Criteria** |
| 1 | Code – unit testing Done |
| 2 | System test plan available |
| 3 | System test scenarios available |
| 4 | Master data of the system available |
| 5 | Environment (hardware and software) for the phase available |

## **Approach**



## **Tasks**

|  |  |  |
| --- | --- | --- |
| **#** | **Activities** | **Responsibility** |
| 1 | Prepare System Test Data (ST cycle 1 will be tested on sample manual data. Cycle 2 will be tested on migrated data.) | CM QA Team |
| 2 | Perform System   Testing (Cycle 1) | CM QA Team |
| 3 | Perform Application Performance Testing | CM QA Team |
| 4 | Fix defects of system testing cycle 1 | CM QA Team |
| 5 | Perform System Testing (Cycle 2) | CM QA Team |
| 6 | Fix defects of system testing cycle 2 | CM QA Team |
| 7 | Update System Test cases and test scenarios | CM QA Team |

## **Validation**

|  |  |  |
| --- | --- | --- |
| **#** | **Activities** | **Responsibility** |
| 1 | Share system and Performance test results with entire Team | CM QA Team |
| 2 | Share feedback on UAT test scenarios | CM QA Team |
| 3 | Share the test cases and Traceability Matrix | CM QA Team |

## 

## **Exit**

|  |  |
| --- | --- |
| **#** | **Exit Criteria** |
| 1 | User accepted system test case and results |
| 2 | Report indicating the UAT results as defined in acceptance criteria Priority 1,2,3 defects and severity 1,2 defects are resolved |

## **Deliverables**

List of deliverables for this phase is as follows:

|  |  |
| --- | --- |
| **#** | **Deliverables** |
| 1 | System test results and Traceability Matrix |
| 2 | System, Performance test results |

## **Responsibilities**

|  |  |
| --- | --- |
| **Activities** | **Responsibility** |
| Prepare System Test Plan | Testing team |
| Review System Test Plan | Test Lead |
| Prepare System Test Specifications | Testing team |
| Review System Test Specifications | Test Lead |
| Perform Integration Test | Testing team |
| Perform System Test | Testing team |
| Perform Regression Test | Testing team |
| Perform Performance Test | Testing Team |
| Defect Log | Testing team |
| Defect Resolution | Development team |
| Defect Resolution Verification and Defect Closure | Testing team |

## 

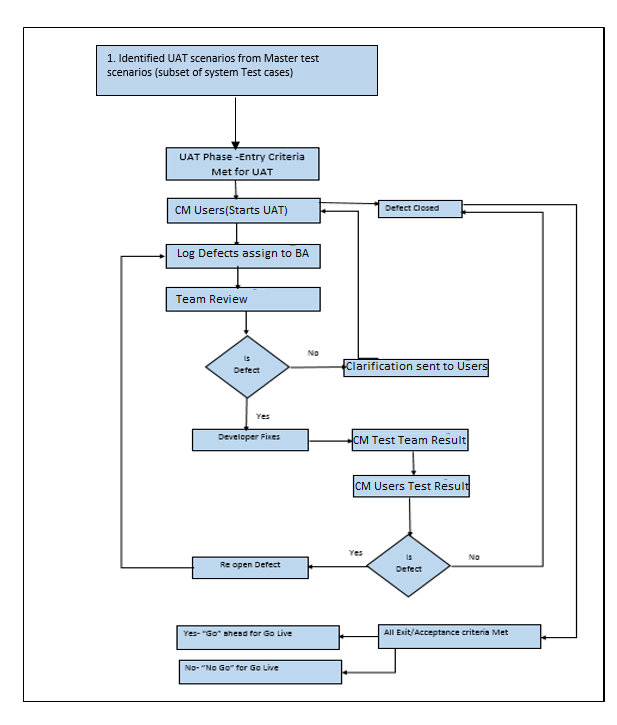
## **User Acceptance Testing**

The User Acceptance Testing (UAT) will be performed by CM users in the test environment using test data to confirm that the application meets the agreed upon requirements.CM Users will perform acceptance testing together with participation and support of the CM development team. CM development team will plan, support and report on acceptance testing to demonstrate that all requirements are met.

## **Entry**

|  |  |
| --- | --- |
| **#** | **Criteria** |
| 1 | System testing completed |
| 2 | Acceptance test scenarios: Priority 1,2,3 defects and Severity 1,2 defects during system test are resolved |
| 3 | System test cases updated as a result of system test execution |

## **Approach**



## **Tasks**

|  |  |  |
| --- | --- | --- |
| **#** | **Activities** | **Responsibility** |
| 1 | Create and identify test data for UAT | CM Users |
| 2 | UAT for common requirements (UAT cycle 1) | CM Users |
| 3 | UAT for common requirements (verification of cycle 1 defects and UAT cycle 2) | CM Users |
| 4 | UAT testing for specific requirements | CM Users |
| 5 | UAT testing for common requirements (UAT cycle 3 verification of cycle 2 defects) | CM Users |

## **Validation**

|  |  |  |
| --- | --- | --- |
| **#** | **Activities** | **Responsibility** |
| 1 | Validate UAT test results | CM Users |
| 2 | Closure of identified UAT defect | CM Users |
| 3 | Validate training surveys | CM Users |

## **Exit**

|  |  |
| --- | --- |
| **#** | **Exit Criteria** |
| 1 | User accepted code |
| 2 | Report indicating the UAT results as defined in acceptance criteria  Priority 1,2,3 defects and severity 1,2 defects are resolved |

## **Deliverables**

|  |  |
| --- | --- |
| **#** | **Deliverables** |
| 1 | UAT test results and Traceability Matrix |

## **Responsibilities**

                                                          UAT and Responsibility

| Activity | Responsibility |
| --- | --- |
| Prepare UAT plan | Testing team |
| Review UAT plan | Test Lead |
| Prepare UAT test specifications | Testing Team |
| Review UAT test specifications | Test lead |
| Perform UAT test | CM Users |
| Defect documentation | CM Users |
| Defect resolution | Development team |
| Defect resolution verification and defect closure | CM Users |

## **System Performance Testing**

As part of performance test for CM application is currently limited to understand the system behaviour with respect to load, with 100 concurrent users trying to access the application. CM team will perform the load test using JMeter tool.

## **Objective**

The main objective of the performance testing is to:

* Evaluate the application scalability as intended
* Identify performance bottlenecks, and provide recommendations
* Evaluate the capability of the underlying infrastructure to host the given application
* Proof of performance of architectural components or web services
* Stress test the application to verify the system threshold

## **Approach**

* Design the test scenarios and test cases
* Build scripts for the identified test cases using JMeter
* Configure test scenarios using JMeter, with all test settings
* Execute the designed test scenarios
* Monitor and collect metrics
* Communicate any issues encountered during the run to all groups involved (If there are major issues noticed during the test, abort or stop test execution on mutual consent and reschedule the test).
* Consolidate the results from the JMeter
* Analyze the logs to identify bottlenecks
* Discuss with the application team for complete analysis and identification of bottlenecks.
* Log defects in TFS wherever applicable.
* Retest the defect after fix
* Share the results with all stakeholders

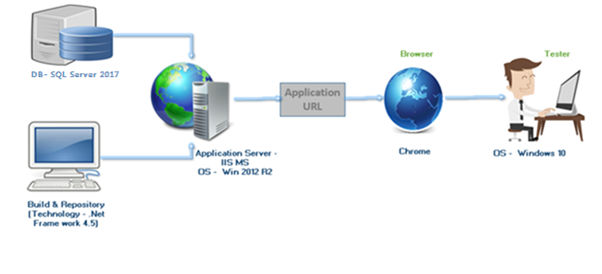
## **Scope**

|  |  |  |
| --- | --- | --- |
| **S No** | **Module** | **Feature** |
| 1 | Login and Logout |  |
|  | Episode Management | Create Episode |
| 2 |  | Edit Episode |
| 3 |  | View Episode |
| 4 | Queue Management | Own Queue |
| 5 |  | Referral Tracker |
| 6 | Episode History | View Episode from History |
| 7 | CM Referral Queue | Create Referral Requests |

# **Test Environment Setup & Build Strategy**

The test environment will be provided by the configuration management team and will be deploying the builds for system testing. The high-level overview of the test environment is as mentioned below.

## **Test Environment Setup**



**Technology:** Dot net Framework 4.5

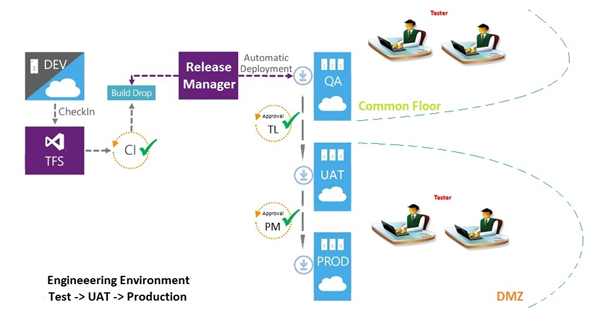
**Database:** SQL Server 2017

**Application Server**: IIS MS

**OS:** Windows 10

**Browsers:** Chrome 70

## **Build Strategy**



# **Testing Tools**

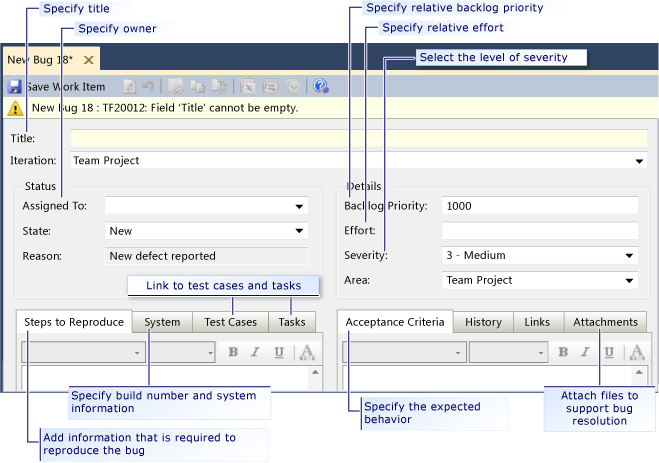
|  |  |
| --- | --- |
| **Activity** | **Tool** |
| Test Defect Tracking | TFS |
| Test Cases document | TFS |
| Test Results Logging | TFS |
| Test Plan & Test Summary Report | TFS/ Word |
| Web Browsers | Chrome |
| Database | SQL Management Studio |
| Performance Testing | JMeter |

# **Defect Management**

## **Log, Analyze and Assign defect**

After the defect has been found by the QA it needs to be reported. Note that found defect can be reported by anyone in the project and logged into TFS. There should be enough information about the case so that the QA could retest it later.

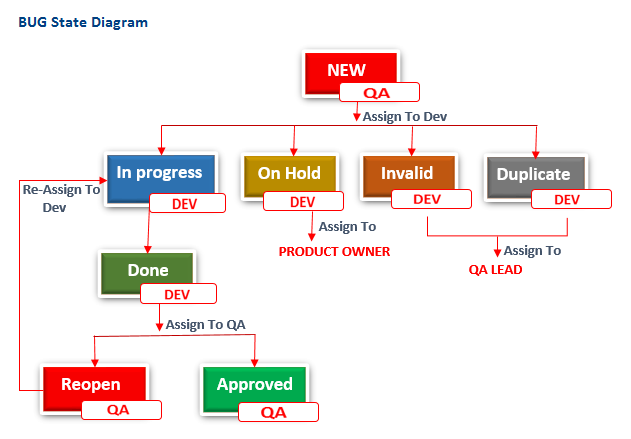
The work item form for a bug contains the fields and tabs in the following illustration:



## **Defect State**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. No | State | Actors | Description | Rules |
| 1 | New | QA | Tester finds a defect and posts it with the status New | QA will assign to DEV |
| 2 | In Progress | DEV | Once the developer starts working on it then state should be, In Progress | Dev will accept the BUG, which has been assigned by QA |
| 3 | Invalid | DEV | IF the bug is invalid, then DEV will give the State, Invalid | 1. Dev will assign to QA Lead with status as Invalid  2. QA Lead will review the bug and then assign it to QA |
| 4 | Duplicate | DEV | If the BUG is duplicate, then DEV will give the state, Duplicate | 1. Dev will assign to QA Lead with status as Duplicate  2. QA Lead will review the bug and then assign it to QA |
| 5 | On Hold | DEV | If a valid defect is decided to be fixed in upcoming releases instead of the current release, then state will be On hold | 1. Dev will assign to Product Owner with status as On Hold  2. QA Lead will review the bug and then assign it to DEV |
| 6 | Done | DEV | Once the developer fix the bug then state will be, Done | DEV will assign to BUG owner who is logged that bug |
| 7 | Approved | QA | After Retesting, the bug is not produced again then the state is Approved | QA will Approve, once bug has been fixed. |
| 8 | Reopen | QA | After Retesting, the bug is produced again then the state is Reopen | QA will reassign to DEV |

## **Defect Life Cycle**



## **Description of the main Defect Life Cycle Statuses**

|  |  |
| --- | --- |
| **State Changes** | **When to use** |
| Usual Workflow | |
| From **New** to **In Progress** | When the developer will accept the BUG and start working on it. |
| From **In Progress** to **Done** | When the developer fix the bug then state will be, Done |
| From **Done** to **Approved** | After Retesting, the bug is not produced again then the state is Approved |
| From **Done** to **Reopen** | After Retesting, the bug is produced again then the state is Reopen |
| From **Reopen** to **New** | When the team reconsiders fixing the bug. |
| From **New** to **On Hold** | When a valid defect is decided to be fixed in upcoming releases instead of the current release, then state will be On hold |
| From **New** to **Invalid** | When the bug is invalid, then DEV will give the State, Invalid and assign it to QA Lead. |
| From **New** to **Duplicate** | If the BUG is duplicate, then DEV will give the state, Duplicate and assign it to QA lead |
| From **Invalid** to **New** | IF QA lead reconsiders fixing the bug then state will be NEW. |
| From **Duplicate** to **New** | IF QA lead reconsiders fixing the bug then state will be NEW. |

## **Defect Severity and Priority**

Severity (priority) level shall refer to the circumstances in which the state business users assign the severity level and prioritize (rank) the defects, errors, or issues discovered during the testing or warranty periods when reporting a particular defect, error, or issue to the Dev for the purpose of classifying the urgency or criticalness to the agency’s business operations. The severity levels shall be defined as follows:

• Severity Level 1 (Critical) shall be defined as a critical system component(s) that is unable to be used by internal and/or external users. Failure causes loss of function or data, and there is not a mutually agreed-upon workaround.

• Severity Level 2 (High) shall be defined as a critical system component(s) that has significant outages and/or failures precluding its successful operation, or producing incorrect results, or possibly endangering the agency’s environment. The system may operate but is severely restricted. Failure causes a loss of function or data, but there is a mutually agreed-upon workaround.

• Severity Level 3 (Medium) shall be defined as a minor problem that exists with the system, but the majority of the functions are still usable, and some circumvention may be required to provide service. Failure causes a partial loss of function, but users can accomplish tasks with a mutually agreed-upon workaround.

• Severity Level 4 (Low) shall be defined as cosmetic and minor errors (for example, grammar errors, color changes, layout, and so on) all the user tasks can still be accomplished.

• Priority 1 (High) - The defect must be resolved as soon as possible. The defect has a severe impact on testing or the production environment. An immediate, emergency patch should be provided to resolve the defect.

• Priority 2 (Medium) - Medium priority defects are resolved after critical priority defects. The defect has a major impact on testing or the production environment. A resolution needs to be delivered in the first upcoming planned build.

• Priority 3 (Low) - The defect should be resolved in the normal course of development activities. It can wait until subsequent upcoming builds (not necessarily the first upcoming build).

• Priority 4 (Others) - A defect with low priority indicates that the system does not meet the specifications or is an irritant to the user (that is, affects the user experience). Resolution can be deferred until after more serious defect have been fixed.

# **Test Schedule**

The testing schedule for Sprint-9 is mentioned below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story** | **Task** | **Owner** | **Start Date** | **End Date** |
| Consume data migrated from Census for processing Referral Request from Hospitals | Execute Test cases | Lavanya | 26/03/2019 | 27/03/2019 |
|  | Validating the logged bugs | Lavanya | 28/03/2019 | 28/03/2019 |
|  | Retesting the impacted area of bugs | Lavanya | 28/03/2019 | 28/03/2019 |
| As a user, I should be able to add HRA for a Member during case management review | Understanding the requirements | Saritha | 22/03/2019 | 22/03/2019 |
|  | Collaborate with BA and Dev and Analyses the impacted Areas and Loop holes | Saritha | 22/03/2019 | 22/03/2019 |
|  | Identifying Test Scenarios | Saritha | 25/03/2019 | 26/03/2019 |
|  | Identifying Test Cases and Test data | Saritha | 25/03/2019 | 26/03/2019 |
|  | Execute Test cases | Saritha | 27/03/2019 | 28/03/2019 |
|  | Validating the logged bugs | Saritha | 29/03/2019 | 29/03/2019 |
|  | Retesting the impacted area of bugs | Saritha | 29/03/2019 | 29/03/2019 |
| As a user, I should be able to see the list of HRA added for the user | Understanding the requirements | Saritha | 25/03/2019 | 25/03/2019 |
|  | Collaborate with BA and Dev and Analyse the impacted Areas and Loop holes | Saritha | 25/03/2019 | 25/03/2019 |
|  | Identifying Test Scenarios | Saritha | 25/03/2019 | 26/03/2019 |
|  | Identifying Test Cases and Test data | Saritha | 25/03/2019 | 26/03/2019 |
|  | Execute Test cases | Saritha | 27/03/2019 | 28/03/2019 |
|  | Validating the logged bugs | Saritha | 29/03/2019 | 29/03/2019 |
|  | Retesting the impacted area of bugs | Saritha | 29/03/2019 | 29/03/2019 |
| Performance Testing | Recording Scripts | Lavanya | 19/03/2019 | 28/03/2019 |
|  | Running the Scripts | Lavanya | 19/03/2019 | 28/03/2019 |
|  | Generating the Report | Lavanya | 19/03/2019 | 28/03/2019 |

# **Risk Contingencies**

## **Risks and Mitigation Plans**

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Mitigation Plan** |
| 1 | Delay in requirements for sprint | High | BA Team needs to work closely with product owner and close the requirement minimum 1 week before the first release |
| 2 | Scope Creep – Change in the scope during test preparation phase. | High | Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out and Key documents such as detailed requirements should be signed-off by all stake holder. |
| 3 | Changes to the functionality may negate the tests already written and we may lose test cases already written | High – to schedule and quality | Export data prior to any upgrade, massage as necessary and re-import after upgrade. |
| 4 | The development process may be delayed or make insufficient progress, which would delay associated test activities | Medium | Project Manager/Scrum Master to work closely with the team to monitor the progress of development. Contingency to be worked into the project plan. BA to plan UAT in such a way that would allow for test schedule to be adjusted compensating for delays. |
| 5 | QA Resource Availability | Medium | QA Lead needs to monitor test activities closely and if needed involve other resource for testing. |

# **Definitions and Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| UAT | User Acceptance Testing |
| NFR | Non-Functional Requirements |
| UI | User Interface |
| CM | Case Management |
| HRA | Health Risk Assessment |