|  |
| --- |
| C:\Users\lgauthier\Desktop\logo.png |
|  |
| **[Carrier Name & Project]**  **Master** **Test Plan** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact Status:** | | | **[Draft or Released]** |
| **Artifact Approver(s):** |  | **Date:** |  |
|  |  |

|  |  |
| --- | --- |
| **Related Artifacts** | |
| **Ref.** | **Name** |
| 1. | QA Space User Guide – [SharePoint link](https://ipipeline.sharepoint.com/sites/ProfessionalServices/professionalservicesus/quality/_layouts/15/WopiFrame.aspx?sourcedoc=%7BFD73E4B0-1C78-4123-8AC3-F4D05AB0D2B9%7D&file=User%20Guide%20for%20QaSpace.docx&action=default) |
| 2. | iGO Keyword Driven Test Automation Manual – [SharePoint link](https://ipipeline.sharepoint.com/sites/ProfessionalServices/professionalservicesus/quality/_layouts/15/WopiFrame.aspx?sourcedoc=%7b76655a85-7d6d-4638-a3e7-797b6a2c9f32%7d&action=edit) |
| 3. | Jira Manual – [SharePoint link](https://ipipeline.sharepoint.com/:w:/r/sites/ProfessionalServices/professionalservicesus/training/_layouts/15/WopiFrame.aspx?sourcedoc=%7B7DF62B26-D742-4CCD-BB6D-413275532662%7D&file=JIRA%20Manual%20External.doc&action=default&IsList=1&ListId=%7B77D5C4A4-B3DA-40A7-AEC3-F78DB2B78A3E%7D&ListItemId=9) |

|  |  |
| --- | --- |
| **Abbreviations and Acronyms** | |
| *[Below are some of the most common abbreviations that will be used. Insert additional lines as needed]* | |
| BRD | Business Rules Document |
| BA | Business Analyst |
| UAT | User Acceptance Test |

Table of Contents

[1. Artifact Workflow 4](#_Toc48565903)

[2. Introduction 5](#_Toc48565904)

[3. Scope of Work 5](#_Toc48565905)

[3.1. Components and functions to be tested 5](#_Toc48565906)

[3.2. Third-Party Components / Integrations 5](#_Toc48565907)

[4. Testing Approach 5](#_Toc48565908)

[Product Availability 5](#_Toc48565909)

[iGO and Illustrations screens 6](#_Toc48565910)

[PDFs forms Generation/ Mapping 6](#_Toc48565911)

[ACORD submission 7](#_Toc48565912)

[eSignature 7](#_Toc48565913)

[State Specifics (screens, forms, XML) 8](#_Toc48565914)

[eDelivery 8](#_Toc48565915)

[In-Flight Cases 8](#_Toc48565916)

[InsureSight Impact 9](#_Toc48565917)

[Changes to access points 9](#_Toc48565918)

[Integration with SSO 10](#_Toc48565919)

[Case Manager Approval - Integration with Agency Integrator and BGA 10](#_Toc48565920)

[Case Manager Approval – iGo emails (click-wrap) 10](#_Toc48565921)

[Integration with Policy Number Generator 10](#_Toc48565922)

[Integration with Credit Card Information 10](#_Toc48565923)

[Integration with License and Appointment (L&A 10](#_Toc48565924)

[Integration with Doctor Lookup 11](#_Toc48565925)

[Integration with Prescription Lookup 11](#_Toc48565926)

[Integration with Address Validation (Google Address/Smarty Streets) 11](#_Toc48565927)

[Integration with Medical Exam Scheduling 12](#_Toc48565928)

[Integration with Illustration (iGO part) 12](#_Toc48565929)

[UI/CE validation (only for Illustration projects) 12](#_Toc48565930)

[Output (only for Illustration projects) 12](#_Toc48565931)

[Calculation (only for Illustration projects) 12](#_Toc48565932)

[Targeted Risk Based Regression 13](#_Toc48565933)

[Destructive Testing 13](#_Toc48565934)

[Compatibility Testing 14](#_Toc48565935)

[Performance Testing 14](#_Toc48565936)

[5. Resources 14](#_Toc48565937)

[5.1. Test Hardware 14](#_Toc48565938)

[5.2. Test Tools 15](#_Toc48565939)

[5.3. Defect Reporting and Priority Definitions 15](#_Toc48565940)

[6. Quality and Acceptance Criteria 16](#_Toc48565941)

[7. Critical Success Factors 16](#_Toc48565942)

# Artifact Workflow



# Introduction

This document describes the testing strategy, testing scope types and approaches for [Carrier Name] project.

[Provide A Brief Summary Of The Goal Of The Project Being Delivered]

# Scope of Work

## Components and functions to be tested

[List the features (e.g., screen, forms, XML) that are in scope of testing and link to BRD/JIRA user stories if existing]

## Third-Party Components / Integrations

[List the features (e.g., MRAS, Lexus Nexus, carrier integrations, Illustrations) that are in scope of testing and link to BRD/JIRA user stories if existing]

# Testing Approach

| **Feature/Functionality to be tested** | **Validation** | | **In scope of this release** |
| --- | --- | --- | --- |
| *[List the feature/ functionality to be tested.]* | *[Description how the feature/functionality being delivered is planned to be tested, specify which verifications will be done]* | |  |
| Product Availability | 1. Product type is correct 2. State approvals correspond to the latest specifications and that products are available in the correct states; 3. Snippet is correct (e.g., correct screen displays when selected) | | Y/N |
| Product testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| iGO and Illustrations screens | 1. Screens are triggered according to BRD rules; 2. field format is correct; 3. Labels are correct; 4. Appropriate default values are set correctly; 5. Rules on field functionality is working as designed (e.g., dependent fields triggering) 6. Fields accept the permissible values and guides user or prevents user when a non-permissible value is entered; 7. Required fields are completed before screen/Illustration generation; 8. Boundary and destructive condition Testing (describe.   Note: State specific testing criteria is identified in the **State Specifics** section below. | | Y/N |
| Screens testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| PDFs forms Generation/ Mapping | 1. Correct forms are generated; 2. Data elements map to the specific form, including overflow; 3. Data changes in IGO are reflected properly on the form; 4. Data is cleared after hiding the fields/screens; 5. State-specific forms are generated and all data from the screens map to their respective fields on the forms; 6. Corresponding forms are included in the signature package for every party and the forms which are not signed by particular signer are absent; 7. Corresponding forms are included into Role70 package; 8. PDF packages for different signature types (Face-to-Face, ClickWrap, Print and Wet sign, etc.) is correct; 9. Signatures/dates are situated in the right places and forms; 10. Order of forms in PDF package is correct. 11. Before/After Comparison   Note: State specific testing criteria is identified in the **State Specifics** section below. | | Y/N |
| PDF testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| ACORD submission | 1. Submission is successful (validation passed); 2. Confirm ACORD Version 3. ACORD paths and properties match BRD/NGSD screen definition in delivered outputs:  * xPath in ACORD is the same as in BRD, if applicable * Value is the same as entered in the application * TC codes and tc value are the same as in ACORD\_Standards, if applicable * Before/After Comparison   Note: State specific testing criteria is identified in the **State Specifics** section below. | | Y/N |
| ACORD submission testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| eSignature | e-Signatures will be verified manually against provided e-Signature requirements. Testing will be conducted in order to confirm:   1. Identify electronic signature option (e.g., AlphaTrust, DocuSign) 2. Signature process is working properly (list type of signature tested) for all signing parties (e.g., PI, AI, all Owners, Payor, guardian …) 3. eSignature restrictions are correct 4. Finger signing on applicable devices working properly for projects with finger signing deployed 5. e-signature screens display and capture of the signatures on the forms match; 6. Face-to-face, remote, and mixed e-signature for scenarios are working properly as defined in the BRD; 7. e-signature by voice; 8. Legal person/guardian signs the case if the party is a minor; 9. Links in e-sig email expire properly,abandoned app 10. Signature is present on 103 and PDF forms;   Verify that the DocuSign Listener correctly supports Agency Integrator, Paramed, and eSignature;   1. Confirm post-submission forms are sent (not sent until submitted) 2. Notification emails are sent, when applicable | | Y/N |
| eSignature testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| State Specifics (screens, forms, XML) | 1. State approvals correspond to the latest specifications and that products are available in the correct states; 2. Correct state specific screen modifications have been made (labels, values, etc.); 3. Correct state specific validations are in place; 4. Correct state specific forms are generated and all data from the screens map to their respective fields on the forms; 5. State specific data is being passed in integrations (e.g. Illustrations integration, eSubmission). | | Y/N |
| State Specifics testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| eDelivery | 1. Emails are delivered; 2. Access to appropriate forms is working; 3. Correct forms are triggered for all parties; 4. The right forms are included in the right packages (e.g., leave behind packages, party specific packages (owner1 vs owner2)) 5. Email with link to view signed forms works correctly | | Y/N |
| eDelivery testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| In-Flight Cases **Inflight testing scenarios**   * Started but not locked * Unlocked and then locked * Locked and then unlocked * Locked and sent out for signatures   **Inflight testing needed**   * Product replacement * Form replacement * Screen changes * Fund changes   **Inflight testing not needed**   * New product * New feature * New rider | * + *[Define the specifc scenarios to be tested] e.g.:*  1. *lock case and then unlock,* 2. *case ready to be signed, finish case and lock,* 3. *Disconnected case not signed, not synched then sync* 4. *Disconnected case signed, not synched then sync*    * Connected and disconnected mode are working properly    * 103 changes have been reflected properly    * Signature are working properly | | Y/N |
| In-flight cases testing approach:  Conditions to perform:   * Screen/Forms Mapping/Form Triggering changes which may affect current In-Flight cases in the PROD system * Fundamental changes, I.e. new face amount requirements, new signing options etc., to a product that has already been released to PROD  1. BEFORE starting In-Flight cases testing:   1.1 Validate In-Flight test cases exist in the QaSpace Project Library (if not, copy from Common Library [Common In-Flight Test Cases](https://jira.ipipeline.com/secure/TestManagementAction.jspa?projectKey=CTCL&viewTab=Work&testRunId=9234&folderId=9323))  1.2 Create Test Runs in the QaSpace Project, include In-Flight test cases that are planned to be executed   1. When and Where In-Flight cases should be created:   2.1 In-Flight cases should be created in the QD/UAT environments used by the project BEFORE relevant project changes are deployed to the environments  2.2 In-Flight cases should be created in the PROD Stg. Environment BEFORE related project changes deployed to environment   1. When and Where In-Flight cases should be executed:   3.1 AFTER the project build(s) with the changes that are deployed to QD/UAT environment. In UAT the cases should be executed as a part of Smoke Testing after the build is deployed but before being released to the customer.  **Note:** The results of in-flight testing will be communicated to the team during daily standups. Any defects discovered will be opened in the Jira system and will be noted in the Test Status report.  3.2 AFTER any build with relevant changes have deployed to the PROD Stg. environment. The cases should be executed as a part of Smoke Testing after the build is deployed.  **Note:** The results of in-flight testing will be communicated to the team during daily standups. Any defects discovered will be opened in the Jira system and will be noted in the Test Status report. | | | |
| InsureSight Impact | 1. Test Set preparation (cases with various input, store CaseID numbers) 2. Cases included into InsureSight report 3. Appropriate case parameters are captured in GoodData report and valid (i.e. Case Status, Submission type, etc.) | | Y/N |
| Good Data testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Changes to access points (e.g., carrier portal, distributor portal, LifePipe Forms pipe, PHS Widget,) | 1. Login changes 2. SAML or SSO update verifications 3. Other API or integration changes or updates | | Y/N |
| Changes to access point testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with SSO | 1. Data is prefilled correctly on IGO screens 2. Screens are validated and case locked 3. Mapping of integrated data on PDF form is confirmed 4. Illustration form is included in the PDF package 5. Mapping and e-signatures on the illustration forms are correct 6. Specific changes are reflected in ACORD | | Y/N |
| SSO integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Case Manager Approval - Integration with Agency Integrator and BGA | 1. Case creation for particular BGA 2. Lock, sign and submit the case to a Case Manager 3. Case Manager can approve or decline case submission to agency. | | Y/N |
| Agency integrator and BGA integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Case Manager Approval – iGo emails (click-wrap) | 1. Case creation for particular BGA 2. Lock, sign and submit the case to a Case Manager 3. Case Manager receives email with link to approval process 4. Correct forms display for review 5. Case Manager can approve or decline case | | Y/N |
| Case manager approval via iGo emails testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with Policy Number Generator | 1. Policy Number was generated and mapped into the form | | Y/N |
| **Policy number generator integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]*** | | | |
| Integration with Credit Card Information | 1. Credit Card validation is in ClickWrap email if Voice signature is not activated 2. Credit Card validation is in ClickWrap email if Voice signature is activated 3. 'Enter Credit Card Information' popup appears on appropriate screen; No error message appears if valid Card number added 4. Appropriate warning message appears if invalid Card number entered | | Y/N |
| Credit card integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with License and Appointment (L&A**)** | 1. Application is unavailable without a correct agent ID 2. Appropriate error message is displayed for an incorrect ID 3. No error messages for correct ID 4. Correct ID triggers additional options 5. ID is correctly mapped to the appropriate forms 6. ID is prepopulated at the Agent's screens | | Y/N |
| L&A integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with Doctor Lookup | 1. Necessary fields are prefilled after one of the search results is selected (Last Name, First Name, M.I., Suffix, Hospital Name, Address (Street, City, Country, State, ZIP), Phone Number) 2. These fields are changed after new search result was selected 3. Max Fields length isn't exceeded by search results 4. Screen is iGO and can be locked after case is complete 5. Prefilled fields are mapped to PDF forms and submitted to ACORD | | Y/N |
| Doctor lookup integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with Prescription Lookup | 1. Necessary field is prefilled after one of the search results is selected (Prescription name, dosage) 2. These fields are changed after new search result was selected 3. Max field length isn't exceeded by search results 4. Screen is iGO and can be locked after case is complete 5. Prefilled fields are mapped to PDF forms and submitted to ACORD | | Y/N |
| Prescription lookup integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with Address Validation (Google Address/Smarty Streets) | 1. Integration starts working after first symbol is entered 2. Search results correspond per requirements 3. After one of the search results is selected, corresponding fields are pre-filled (Streets (2,3), City, Country, States, ZIP) 4. These fields are changed after new search result was selected 5. Max Fields length isn't exceeded by search results 6. Screen is iGO and can be locked 7. Prefilled fields are mapped to PDF forms, submitted to ACORD 8. Non-US Address can be validated only by Google Address service 9. Smarty Streets supports Coding Accuracy Support System (CASS) | | Y/N |
| Address validation integration testing approach:   1. Define Address Validation service (Google Address/Smarty Streets) that is used on the carrier 2. Define screens where Address Validation service is used 3. Copy to Project QaSpace test cases applicable for your project from the QaSpace Common Library ([Common Google Address/Smarty Streets Validation Test Cases](https://jira.ipipeline.com/secure/TestManagementAction.jspa?projectKey=CTCL&viewTab=Work&testRunId=9234&folderId=10093)) 4. Perform testing on previously defined screens. | | | |
| Integration with Medical Exam Scheduling | 1. The information (vendor -> dates) is transmitted to the exam vendor via an ACORD 121 | | Y/N |
| Medical exam scheduling integration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Integration with Illustration (iGO part) | 1. The screens are filled with the necessary information about the insured(s) and insurance product(s) and then the case should be locked. In logs, all values should correspond the specification: paths, values, type codes 2. The responses should be verified for different states/products/parties combinations | | Y/N |
| Integration with illustration testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| UI/CE validation (only for Illustration projects) | 1. Boundary conditions testing is successful; 2. Correct wording and triggering; | | Y/N |
| UI/CE validation testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Output (only for Illustration projects) | 1. Static text is displayed in accordance with BRD, has correct format and page position; 2. Dynamic text according to State, Age, Gender and any other UI triggers specific is accurate. | | Y/N |
| Output testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| Calculation (only for Illustration projects) | 1. All Calculation changes will be tested against calculations tools which contain expected values for the Output. A calculation tool is an Excel spreadsheet with policy calculation according to unique UI input and product rates. Calculation testing covers all combinations starting from the simplest cases. | Y/N | |
| Calculation testing approach: *[Describe how above functionality will be tested (functional targeted testing, end to end, regression…)]* | | | |
| 1. Link to the test results summary must be provided in the UAT DoD.  |  |  |  | | --- | --- | --- | | Targeted Risk Based Regression | 1. This approach is based on identifying the most risky areas/features/external issues of the application and performing targeted risk based regression testing on these areas of the application under test. Risk based targeted regression may often be executed as a substitute for full application regression testing towards the end of a release cycle. | Y/N | | Targeted Regression testing approach:   1. Collaborate with your PM/BA to review this type of testing and identify when it should be performed (per Sprint, per Release, etc.) 2. Analyze and identify the most risky areas and features of your application, analyze external issues 3. Review Common Test Cases stored in QaSpace | Common Test Library [Targeted Risk Based RegressionTest Cases](https://jira.ipipeline.com/secure/TestManagementAction.jspa?projectKey=CTCL&viewTab=Work&testRunId=9234&folderId=13839) 4. Add applicable Test Cases to your team’s QaSpace Project and adapt as needed 5. Perform testing on current QA/UATenvironment and report results 6. Regression testing must be completed for   In-Flight Cases  TX103  eSignature testing  PDFs forms Generation/ Mapping | | | | | | |
| |  |  |  |  | | --- | --- | --- | --- | | Destructive Testing |  | Y/N |  | | Destructive testing approach: • Destructive testing scenarios are designed to simulate more realistic business use cases that end users may execute in the Production environment as part of their use of the system in a regular business day setting.  1. BEFORE starting Destructive test planning:  1.1 The team will collaborate on the need and extent for carrying out Destructive testing  - Collaboration with client on most common Business Case scenarios should take place as part of test design process during the project. 1.2 Ensure Destructive test cases exist in the QaSpace Project Library (if not, copy from Common Library [Destructive Testing Cases](https://jira.ipipeline.com/secure/TestManagementAction.jspa?projectKey=CTCL&viewTab=Work&testRunId=9234&folderId=9235)) 1.3 Create Test Runs in QaSpace Project, include Destructive test cases that are planned to be executed  2. When and Where Destructive testing should be performed: - Destructive Test cases should be executed in the UAT environment after the final - or Prod candidate - planned project build has been deployed  Note: The results of destructive testing will be communicated to the team during daily standups. Any defects discovered will be opened in the Jira system and will be noted in the Test Status report. | | |  | | | | |
| Compatibility Testing | For existing customers; request a report from iPipeline IT Support detailing the types of devices agents are using to access the iGO Production Environment. To request this report, send an email request to [itsupport@ipipeline.com](mailto:itsupport@ipipeline.com) asking for the following data points from Production Active Environment:   * The GAID(s) for customer * The timeframe of login activity, i.e. past 60 days, previous month, etc. * What data to report. Available device data includes:   + Login Device Type   + Operating System   + Browser     - Browser Type     - Browser version | |  |
| *[Define what is applicable to the project after agreeing on the affinity platform with the carrier and reviewing Production login data from IT Support. If the customer is deploying to Production for the first time, the project team should get agreement from the customer on what the affinity and platforms of interest are in order to plan compatibility testing coverage.]*  *For additional detail on what platforms iGO & Illustrations are officially supported on please refer to the* [*iGO System Requirements*](https://www.customerportal.ipipeline.com/knowledgebase/igo-system-requirements/) *entry in the on-line Customer Portal Knowledgebase*   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | IE11/Edge | Chrome | FF | Safari | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | | Win 10 | Y/N | Y/N | Y/N | N/A | | MacOS | N/A | N/A | N/A | Y/N | | Android | N/A | Y/N | N/A | N/A | | iPhone | N/A | N/A | N/A | Y/N | | | | |
| Compatibility testing approach: *[Describe how above functionalities will be tested (functional targeted testing, end to end, regression…)]* | | | |
| |  |  |  | | --- | --- | --- | | Performance Testing |  | Y/N | | | | |
|  | | | |

# Resources

## Test Hardware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Role** | **Resource** | **Hardware configuration** | **Software configuration** |
| 1. | QA environments | *[List all QD environments]* | TBD | TBD |
| 2. | UAT environments | *[List all TD environments]* | TBD | TBD |
| 3. | Prod environment | *[Insert Prod environment]* | TBD | TBD |

## Test Tools

|  |  |  |
| --- | --- | --- |
| **#** | **Tool** | **Comment** |
| 1. | iPipeline JIRA | Defect Tracking System. Agile Management Tool |
| 2. | QA Space | JIRA testing plug-in for storage and management of manual and automated test scenarios |
| 3. | Team Foundation Server (TFS) | Project code, Illustration test results |
| 4. | SharePoint | Artifact Repository |
| 5. | Jenkins | Continuous Integration server for running iGO automated tests by schedule |
| 6. | Automated Calculation Engine Batch Testing Framework | Running Illustrations automated regression tests |
| 7. | Beyond Compare | Data comparison utility |
| 8. | SauceLabs | Cloud-based platform for running tests on requested browsers and operating systems |
| 9. | Chatter | Messaging Communication with team members |
| 10. | Microsoft Office applications (Word / Excel / PowerPoint) | Test Plan, Test Summary Reports, presentation materials, other project documentation creation |

## Defect Reporting and Priority Definitions

iPipeline JIRA will be utilized for defect reporting and tracking.

The following priority definitions will be utilized when processing defects:

**Critical**

* A major issue where a large piece of functionality or major system component is completely broken
* Defect that causes total failure of the software system or subsystem or unrecoverable data loss or severe impact to data integrity
* There is no workaround
* In general, a critical defect would prevent the product from being released or block testing of significant portions of functionality
* Examples are login errors, screens not displayed, PDFs not generating, screens not locking keeping the user from starting the signature process

**High**

* Functional or visual issues that materially impact system functionality
* Defect results in severely impaired functionality
* A work around may exist, but its use is unsatisfactory
* This would include issues like incorrectly worded questions (not typos), incorrect field or screen availability, incorrect edits, incorrect PDF mapping

**Medium**

* Functional issues that do not materially impact system functionality
* Defect causes failure of non-critical aspects of the system, or produce incorrect, incomplete or inconsistent results
* There is a reasonably satisfactory work-around
* The product may be released if the defect is documented, but the existence of the defect may cause customer dissatisfaction
* Examples are incorrect field minimum or maximum limits or dropdown contents out of order

**Low**

* Visual issues that do not materially impact system functionality
* A work-around exists or, if not, the impairment is slight
* Generally, the product could be released, and most customers would be unaware of the defect’s existence or only slightly dissatisfied
* Examples are minor typos or minor field alignment problems

The proper JIRA workflow along with Base and New Screen Designer issues workflow are outlined in the JIRA Manual artifact (see ref. 3).

# Quality and Acceptance Criteria

1. Application should work according to the requirements and functional specification.
2. All planned tests are added to QA Space and executed.
3. Automated regression tests are run.
4. All found defects are reported to JIRA.
5. There are no open Critical / High severity defects.
6. Defect Containment Rate is 90% and higher.
7. Root Cause Analysis is conducted for external defects if necessary.
8. Test artifacts such as Test Plan, Test Summary Report, etc. are updated and uploaded to SharePoint storage.
9. Before UAT deployment all outstanding issues are described in Release Notes that are shared with team

# Critical Success Factors

1. Scope of work is defined, BRD and functional specification are signed-off. Requirement changes that are identified after sign off should be reviewed to determine if change management process must be followed to accommodate.
2. Schedule announced in the Release Plan is followed. Development and testing of all functionality is completed in time.
3. There are no known Critical / High severity defects at the time of Release.