**QA Release Process**

For every release, the following testing aspects are followed:

1. **JIRA Dashboard used for tracking JIRA ticket closures for that specific release**
   1. JIRA tickets can be of the following types:
      1. Customer tickets
      2. Regression tickets
      3. Any other ticket assigned to specific QA engineers
2. **Regression testing**
   1. Maintained in testrail.
      1. QA Lead to create testrun, review and get it approved.
      2. Reviewers/Review date, approvers/approve date all tracked in testrail testplan in "Description" section. (example: 4.10 plan <http://probe.cliqr.com/index.php?/plans/view/268> )
   2. **Prior to 4.10 release:** For each release, regression test plan is tracked in an excel spreadsheet in <http://docs.cisco.com>
      1. <https://docs.cisco.com/share/page/site/nextgen-edcs/folder-details?nodeRef=workspace://SpacesStore/2cdc5812-047a-486c-bd05-292b20cdf3e2>
         1. Add the reviewers, the date reviewed and the approval info in the testplan and copy this into [docs.cisco.com](http://docs.cisco.com)
         2. Reviewers/Approvers: QA Leads (Fayas/Vel)
         3. smartsheet is used by QA engineers during the testing to update on a regular basis (as work-in-progress document)
         4. Once the testing is completed, the final 4.9.1 regression plan with test results updated will also be copied to docs.cisco.com (by QA Lead)
         5. After the plan is uploaded, it must be approved by Gautam
            1. Select ...More, then Start Workflow
            2. Workflow = Document Approval
            3. add Gautam as approver
            4. Click Start Workflow
3. **New feature testing**
   1. New Features are tracked in Rally User Stories. The following can apply even for User Stories that don't require any code changes (i.e. testing only feature)
   2. Testplan/Testcases for new features
      1. Create tasks  for User Stories in Rally for the following:
         1. Rally Task - Create test cases in test rail and create a test run in test rail.
            1. Make sure to include testrail testsuite link in this task. And then mark the task complete
            2. Include Reviewers/approvers, Review/approval date in the "description" field in the testsuite.

Reviewer/approver - dev engineer or dev lead or scrum lead

* + - * 1. Include User story number in the description of the testrun
        2. Test cases may be user story specific or end-to-end feature testcases (based on the phase of the release cycle dev or qa or talk to your scrum lead/qa lead)
        3. Scrum teams OR QA Leads (Fayas/Vel) to decide which if any of these should be included in Regression testsuite (for future release testing). This has to be appropriately labeled in Testrail. QA engineers can provide the input.
      1. Rally Task - Execute test cases that have been created for this User Story
         1. Make sure to include testrail testrun link in the notes section of the User Story.
         2. And then mark the task complete when all testcases are executed.
      2. Rally Task - track automation for this user story - one of the options below
         1. Automation as part of the user Story

Add task in Rally.

* + - * 1. Automation tracked separately as part of Bhaskar/Automation team's tasks.

Open ticket in JIRA automation project

Put the link to the automation JIRA ticket in the rally task and mark the Rally task complete.

Automation may not be done/completed during the user story timeframe.

* + - * 1. if automation already exists

Add task in Rally - put link/pointers to the automation suite/code and mark the task done.

* + 1. Engineers may choose to create the first draft in Excel and import into testrail
  1. Defects tracking
     1. Defects which are fixed in the Sprint should be tracked in Rally.
     2. Defects which are not going to be fixed in the sprint should be tracked in JIRA (because not relevant to US or corner case and needs more time to fix)
        1. QA engineer/Submitter to put the Fix version in JIRA as the current release being worked on (unless the submitter is sure that this does not need to be fixed in the current release)
        2. Priority definitions - see link in table below.

1. **Automation testing**
   1. Jenkins jobs exists for various kinds of automation suites
   2. Cron jobs setup for automation runs for daily builds
   3. JaCoCo code coverage is generated for main branch on a daily basis
   4. PR Validation exists for all dev commits. (enabled automatically for main branch; other branches added on demand basis)
   5. Triage Tool - to automatically triage the daily results, track historical results
2. **QA Phase (Milestone)**
   1. Entrance Criteria
      1. Dev Complete
         1. All User Stories in Rally marked Accepted
         2. (Any exceptions approved by release manager/scrum lead)
         3. All Feature branches merged to the release branch.
      2. All Smoke/regression suites run and passing with no blocker/critical issues
         1. Exceptions to the above will have to be agreed upon by the QA Leads.
      3. Include Build number in QA Readiness ticket in JIRA.
      4. QA Phase readiness checklist ticket in JIRA (Rob to create for each release)
   2. Exit Criteria
      1. Feature testing - 100% executed
         1. All Rally User stories in Accepted state
      2. Manual Regression - 100% executed
      3. Automated Regression/Smoke - 100% executed
      4. All Resolved tickets in JIRA have been closed (exceptions needs to be approved by Release manager/QA leads)
      5. 0 Blocker, 0 Critical, < 30 Major bugs - unresolved at the end of the release (before posting images to software.cisco.com)
      6. Release readiness checklist ticket in JIRA created and tracked by Rob for each release
3. **Bug scrubs**
   1. During the dev cycle and early part of the QA cycle
      1. Release manager scrubs on a periodic basis by himself/herself
   2. Toward the end of the QA cycle (typically ~4 weeks before FCS)
      1. Bug scrub meetings may be scheduled on a regular basis (release manager to decide)
4. **Documentation review**
   1. Customer documentation is reviewed by QA (and/or Dev Engineers), leads and managers
      1. Feedback, corrections given back to doc writer using JiRA tickets and/or emails.
5. **Automation development**
   1. Automation Project (AUT) in JIRA used for tracking the following:
      1. new feature, enhancements, code coverage gaps/bugs fixes
      2. Automation Tickets for issues found with automation code in daily regressions
6. **QA testbeds**
   1. Central QA environment
      1. Nightly build is used to upgrade
      2. Always on the current major/minor release being worked on
      3. Most manual regression is run on this testbed
   2. Individual testbeds
      1. Depends on issue being worked on - setup related issue/specific env requirement
      2. Cloud used depends on issue type - QA engineer to decide and check with lead/manager when in doubt
   3. Scale testing
      1. AWS
      2. VMware
      3. Openstack
7. **SaaS site testing**
   * 1. Pre-upgrade testing - done a day (or close to) the upgrade
     2. Upgrade done by DevOps team
     3. Post-upgrade testing - done right after upgrade
8. **Tools/Utilities available for QA to use**
   1. Cumulus app repositor, Triage tool
9. **Training**
   * 1. [New Joinee Training Program (QA) - 30days](https://confluence-eng-sjc3.cisco.com/conf/display/qa/New+Joinee+Training+Program+%28QA%29+-+30days)