

APT Process

Following a PR from file to delivery

Process Flow Overview

- PR Prioritization
- Development Process
- Testing
- Delivery Procedures

Priorities

- 150 - Critical
- 250 - Important
- 350 - Medium
- 450 - Low
- 550 - Unimportant
- Developers can change the priority up and down from the 50 in order to provide information on planned work order

States - Development

- NOREC
- OPEN
- ASSIGN
- RTT
- FAILED

States - Testing

- RTT
- TEST
- FAILED
- PASSED
- LIEN

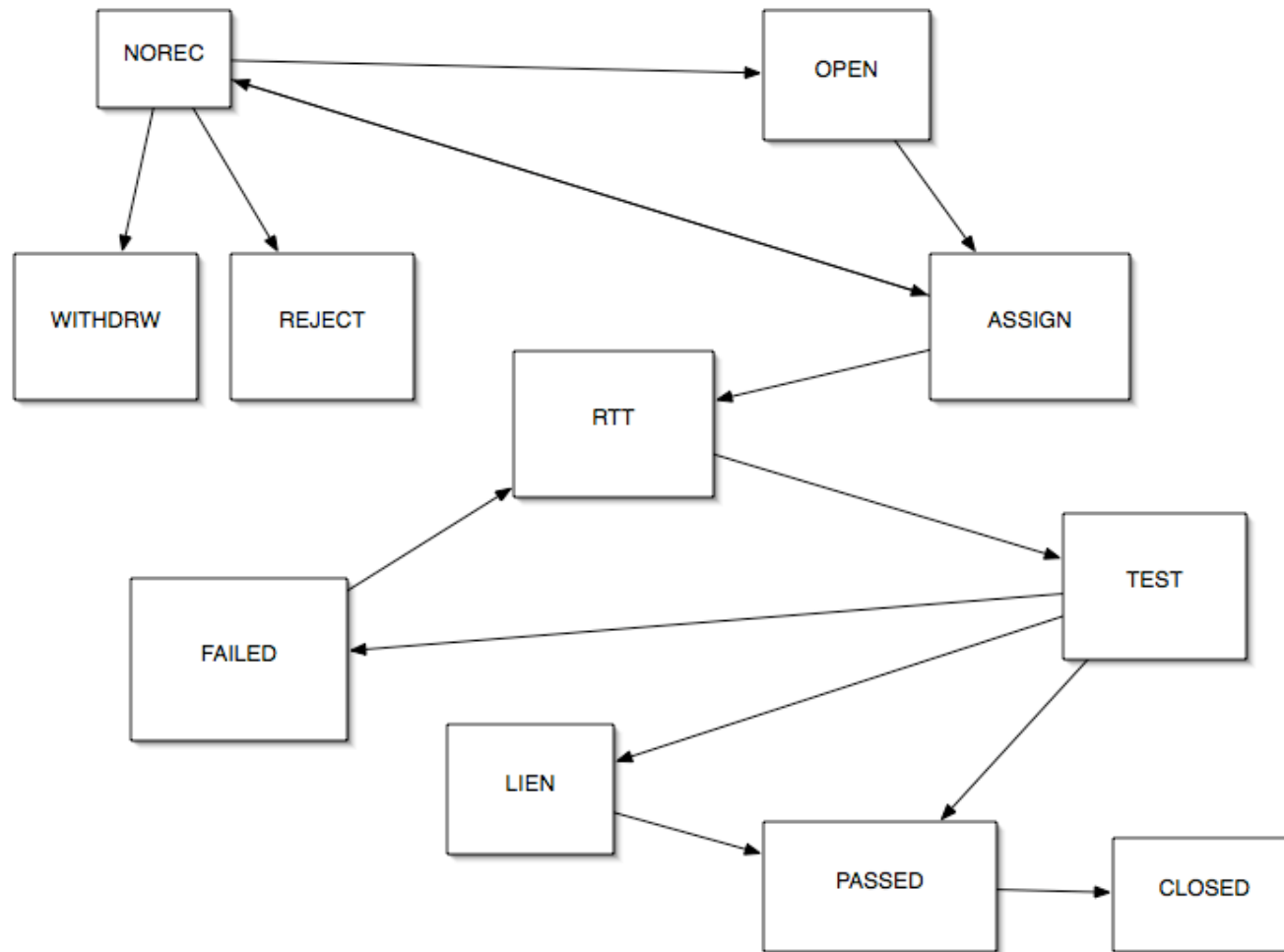
States - Other

- WITHDRW
- REJECT
- DEFERRED
- CLOSED

PR Prioritization Meeting

- Subsystems:
 - PROPINST
 - PCTOOLS
 - PHASEONE
 - APT*
- NOREC vs. In Progress
- User input to Priorities
- Common Keywords
 - Project=?, Effort=trivial, Developer=true, Buglist=yes

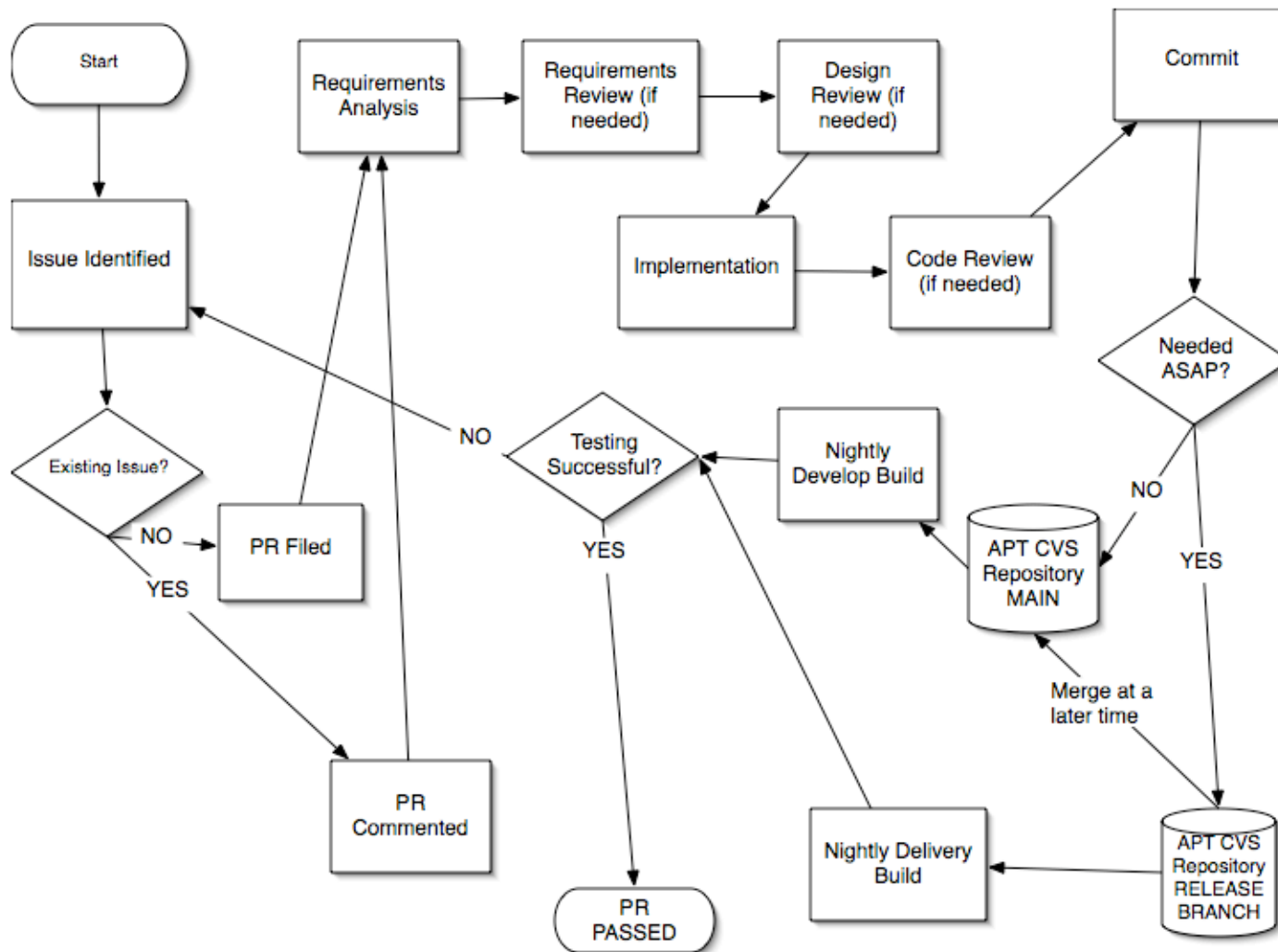
PR Lifecycle



Development Tasks

- Requirements Review/Analysis
- Design
- Implementation
- Developer Testing

Development Process Overview



Requirements Review/Analysis

- Developers are responsible for ensuring requirements are complete, clear, and can be coded.
- PR must be kept up to date with conversations and clarifications that come up - any question worth asking is worth capturing the answer
- Customers to solicit for requirements: Testers, PCs, Phase I PIs, Phase II PIs, ISs, Commandos, Trans/CASM developers, Phaseone manager

Design Process

There are 3 levels of Design:

- Simple
 - No review, implied by implementation comments
- Detailed
 - Design comments are added to PR and reviewed
- Complex
 - full scale design review should be held, including developers and others as appropriate

Implementation

- APT Code is stored in the CVS Repository on RA
 - CVSROOT
 - :ext:<username>@ra.stsci.edu:/usr/ra/cvsroot
 - Module name = APT
 - Can checkout a new world or keep using the same one, but be careful of committing the wrong code, or forgetting to add new files
- Coding Guidelines are here
<http://apst.stsci.edu/documents/CodeConventions.html>

Developer Testing

- Make sure the requirements are met
- Make sure the build runs successfully
 - missing files committed, etc.
- **NEW:** Regression Test mechanism has been set up with ant and JUnit testing
 - Developers still learning to use this mechanism for unit tests

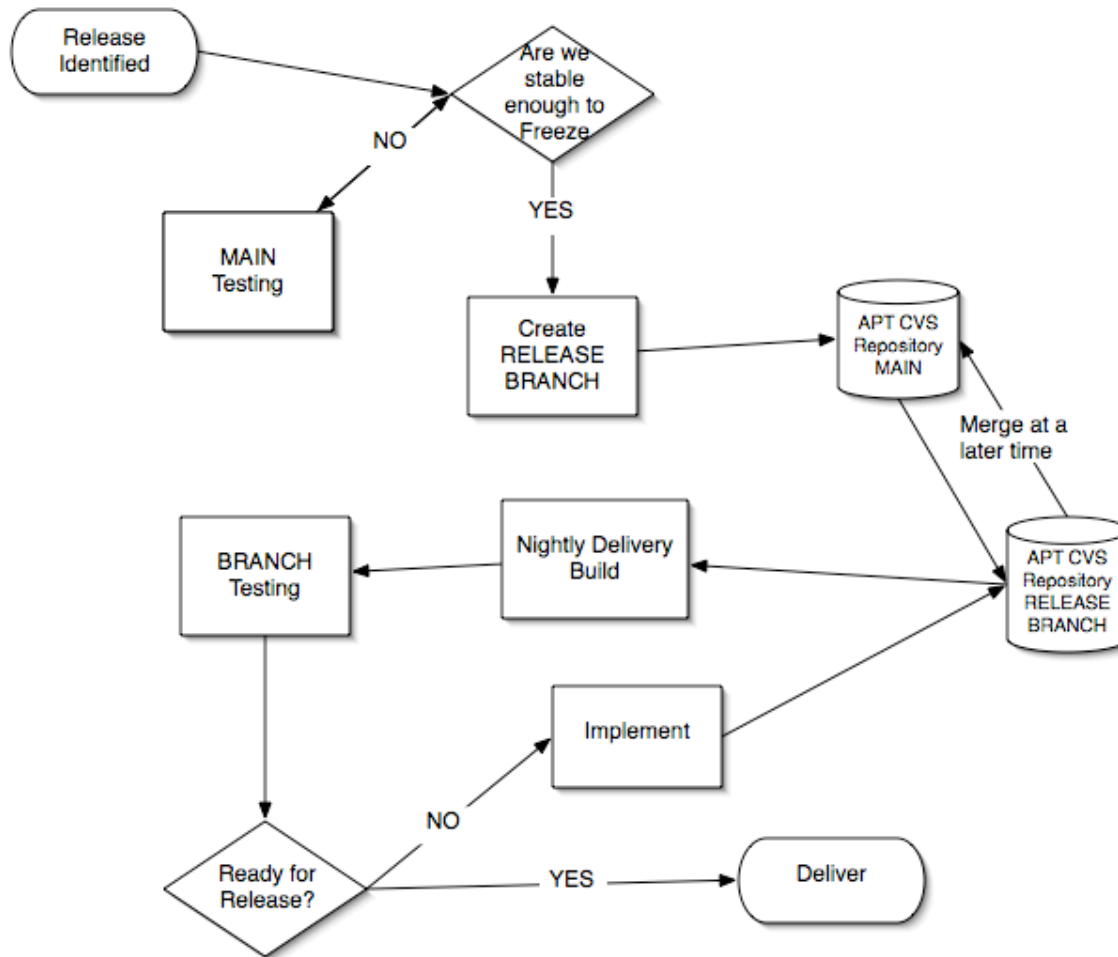
Code Commit

- Code Review needed?
- To commit to Branch or Main
 - Decision made before implementation usually

Testing

- Once the PR is complete - the developer comments it with implementation notes and notes for the tester
- PR Testing assigned by APT Testing Coordinator (Karla) based on expertise and availability
 - Current testers are Karla Peterson, Ron Downes, Chris Ritchie, and Kate Henshaw
- Interface Regression is done just before release, to try out all the pieces and see if it has changed
- A Regression suite maintained by Christine is run periodically, especially prior to release, to catch unintended side-effects

APT Release Process



APT Deliveries

- We have Crons that run nightly to create
 - Nightly Develop
 - Nightly Delivery (aka Beta branch)
- We keep a week's worth of builds, changing a tag each day of the week
- Latest nightly builds are available on Solaris from a standard link
- Testers may download their own copy for baseline testing

Handling LISP

- CASM and Trans are included in APT
- Both are delivered to a staging area whenever a new version is available for APT
 - /project/ra5/apt/lispPackages/HEAD
- When we branch, we copy the Lisp to a new directory
- If a Lisp program must be redelivered, it still goes to HEAD, then we copy it to the Branch directory for build into the Nightly Delivery

APT Delivery

- When we decide to release a version of APT
 - Start new build process from last Nightly Delivery
 - Tag that version of the code
 - CVS checkout and build
 - Use Install Anywhere to create the build bundles
 - Install on Solaris with links:
 - hot-apt - internal releases only
 - apt - external releases
 - Inform apt-users and apt-list and others as needed
- After Delivery, Merge Branch changes back onto Main

Release Schedule

Current Release is 17.2.3

- Major release is by cycle: 17
- Minor releases every 2-4 months:
 - 17.0, 17.1, 17.2, 17.3, 17.4
 - Odd releases are for internal use, but may not used
 - Even releases are for Phase I, Phase II, and Calibration
 - Any planned release may be cancelled if not needed
- Fixes off the branch can be turned around in 24 hours
 - Checkout branch, and commit to branch, test after overnight build, then deliver after incrementing release

Things not covered

- APT Servers
- Documentation
- Schedulability