Utilizing Atlassian JIRA for Large-Scale Software Development Management

Presentation to

14th International Conference on Accelerator & Large Experimental Physics Control Systems (ICALEPCS)

October 6-11, 2013

John Fisher

NIF Deputy IT Manager

Lawrence Livermore National Laboratory • National Ignition Facility & Photon Science

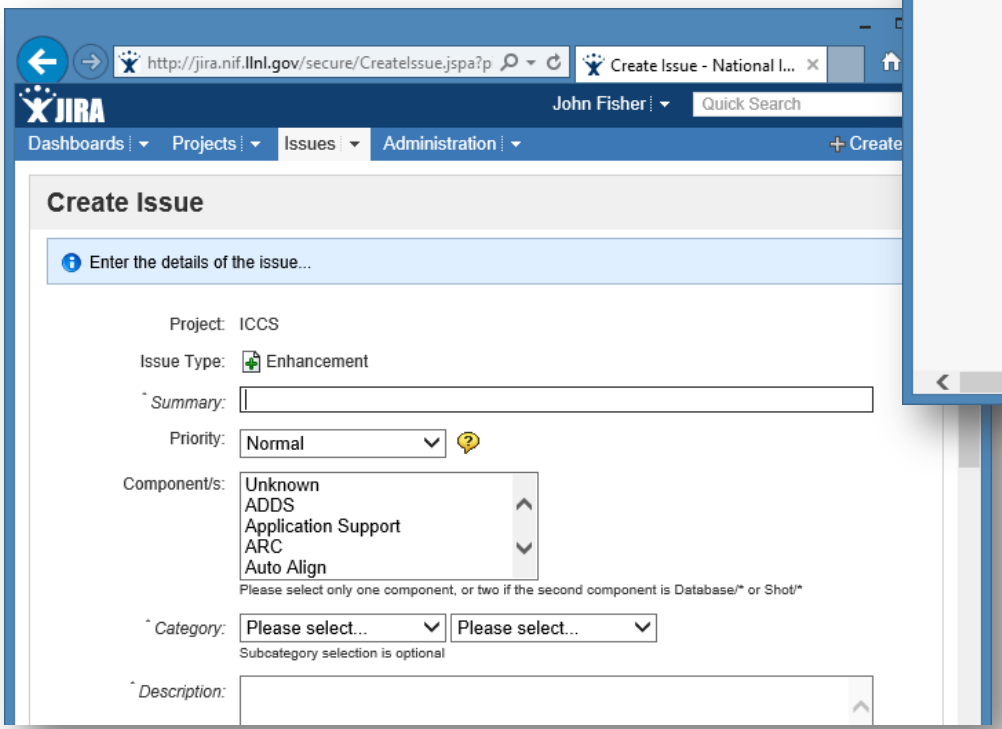
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

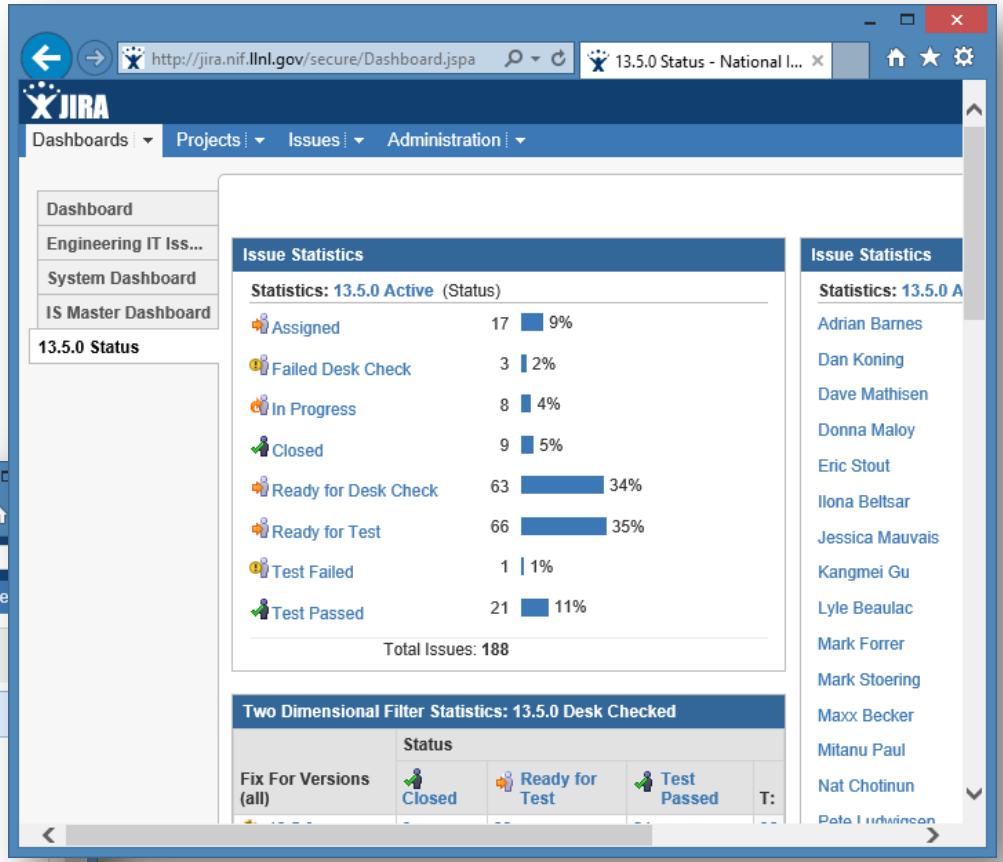
Some background details on ICCS and JIRA

- ICCS is the Integrated Computer Control System for the National Ignition Facility (NIF)
- 3.5+ million source code lines
  - Includes Java, Ada, C, XML, SQL
- NIF regularly receives ICCS software updates
  - ~6 major releases a year, ~200 change requests each
  - Patches about once/week
- ICCS development began in 1997
  - A homegrown system was used
- Atlassian JIRA for issue tracking started in 2006
  - Existing data was transferred with JIRA data migration tools

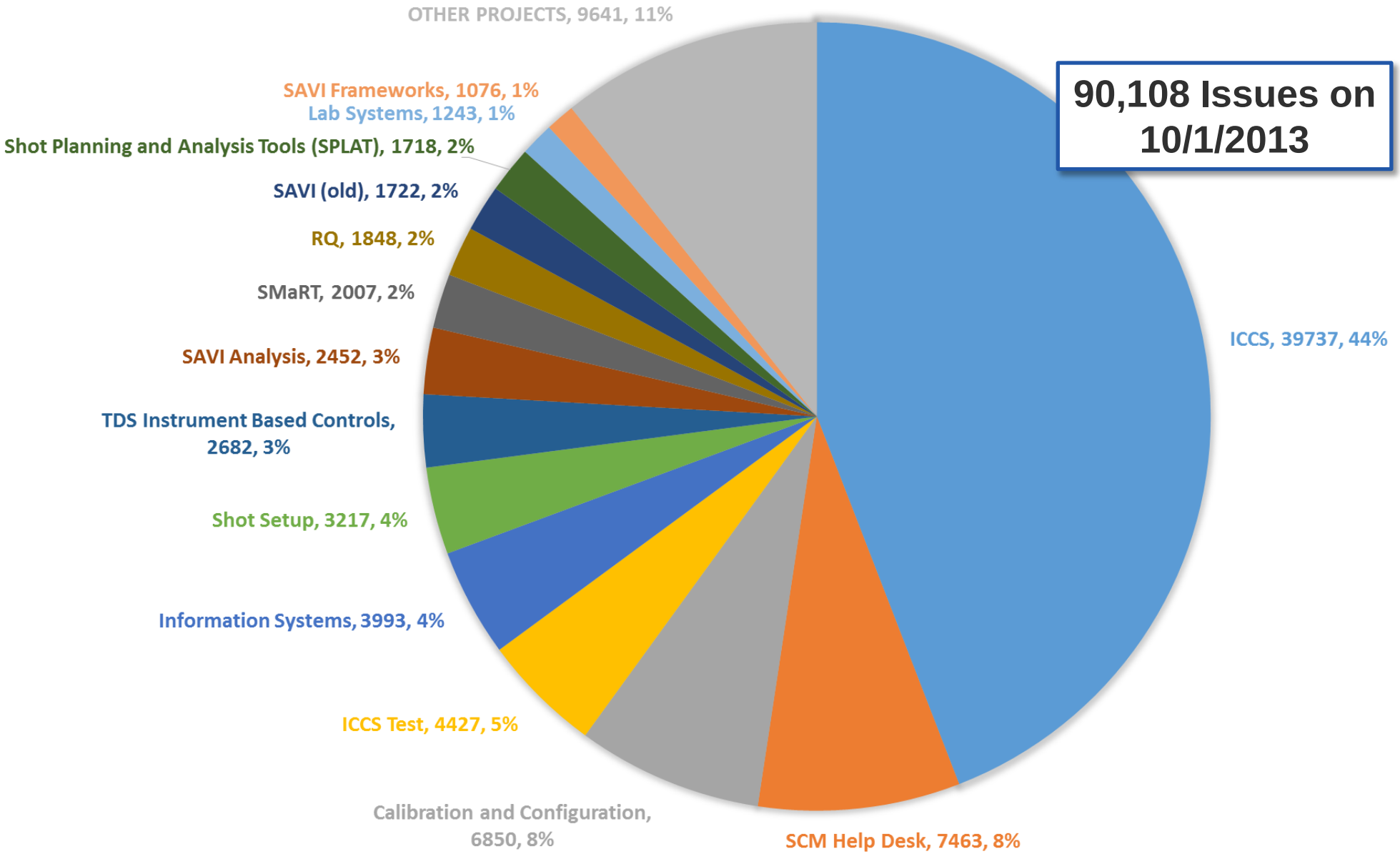
Atlassian JIRA is a flexible issue tracking system

- **JIRA is highly customizable:**
  - **Issue types**
  - **Workflows**
  - **Custom fields**
  - **Notifications**
  - **User entry screens**





JIRA is widely used at NIF, and not just for software



Fields used for each JIRA issue

Triaging

- Project
- Type (Change, Problem, Enhancement)
- Components
- Environment
- Environment
- Category
- Origin
- Priority
- Keywords
- Applicable DB Environments
- Rank

Work Execution

- Assignee
- Due Date
- Original Estimate
- Fix Version

Quality Control

- Final Verification Environment
- Final Verification Type
- QA Verifier
- Test Database
- Test Instance
- Test Release

Documentation

- Summary
- Description
- Reporter
- Recommendation
- Affects Versions
- LoCoS #
- Release Notes
- Resolution Notes
- Root Cause of Defect
- Process Restart Required
- Process Restart Details
- End-User Release Notes
- End-User Description

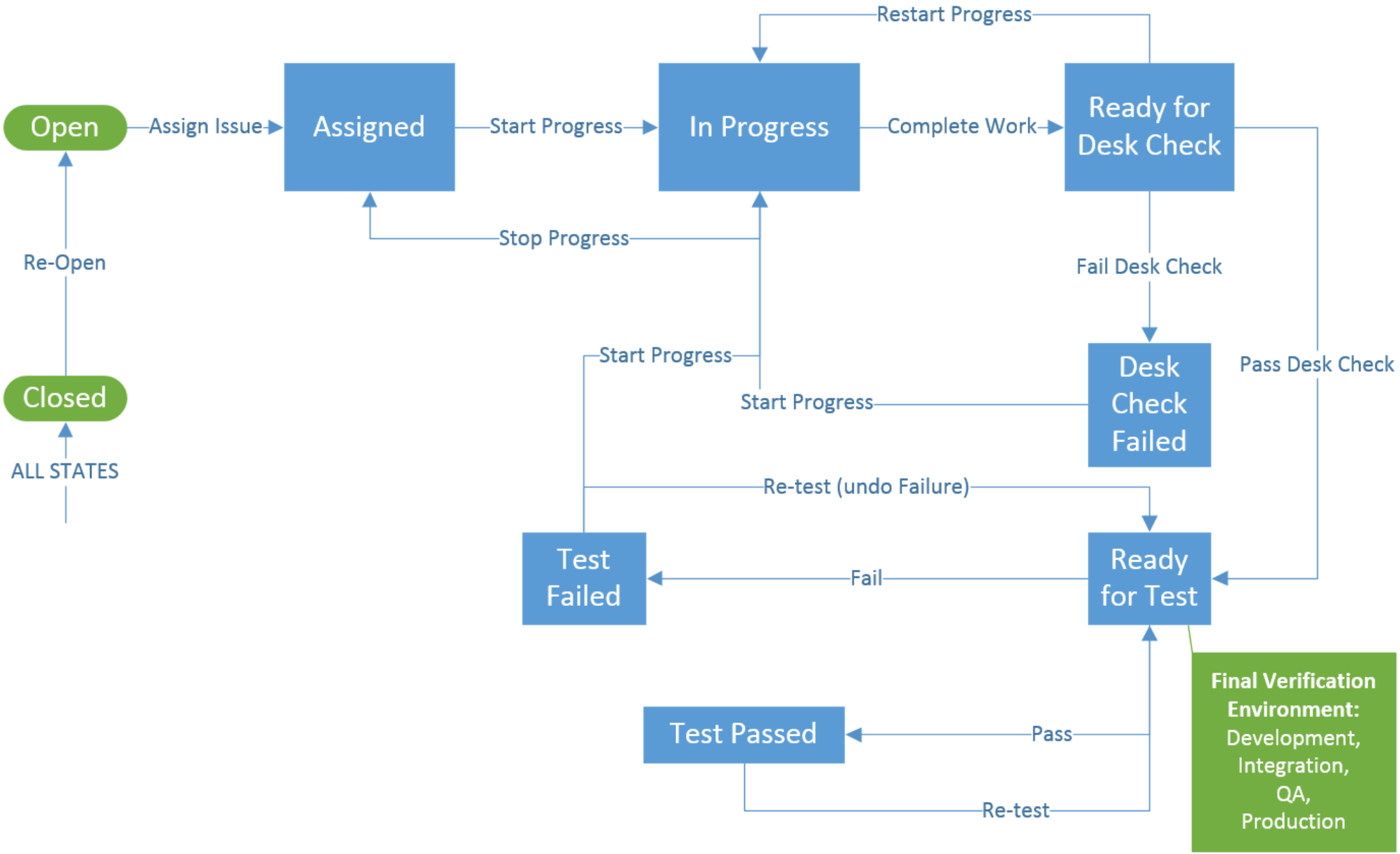
Review

- Requirements Review
- Desk Checker
- Integration Requirement
- Integration Checked
- SCCB Approval Date

Configuration Management

- Wrap-around
- CM Release
- CM Snapshot

The NIF software workflow implemented in JIRA



Quality Assurance uses a risk-based approach

- IPRB (Integrated Product Review Board) Reviews
  - The formal review process for large NIF Engineering projects

- Formal Software Design and Code Reviews
  - The need for a review is assessed by higher level management
  - Reviews are expected for:
    - Fundamentally new software design or functionality
    - Software with significant risk or impact
    - Existing software that has encountered a significant failure

- Desk check
  - A secondary developer reviews code changes and JIRA content
  - Required for all JIRA issues

Formal reviews are tracked in JIRA

- Formal reviews are tracked in JIRA
  - Issues are assigned to software releases
- JIRA issue content includes
  - A pointer to the review materials
  - A list of participants
  - Results of the review
- Action items are tracked in JIRA
  - Resolution Notes (minor rework)
  - JIRA subtasks
    - Future work
    - External scope
    - Major problems
- The scribe for the review is the QA Verifier in the JIRA workflow

Possible Review Participants
Design/code originator
Scribe
Experienced technical reviewers
NOT experienced technical reviewers
Customers
QA Manager
Tester

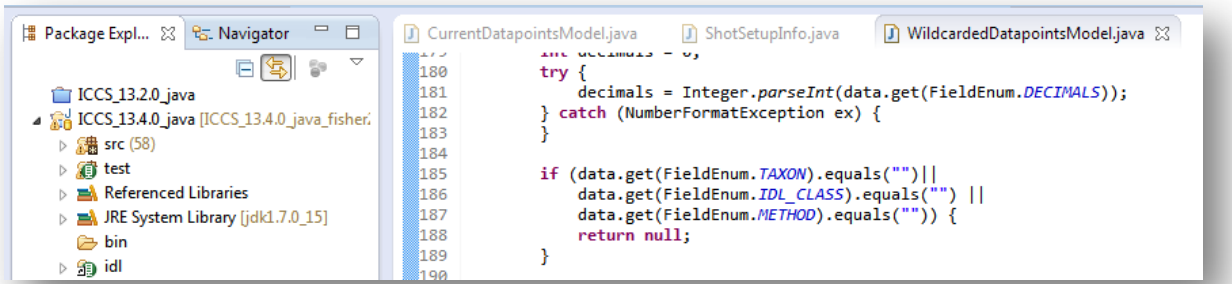
Code Review Items
Modules to external interfaces
Significant algorithms
Reports from FindBugs

Design Review Items
Software requirements
External interfaces
Design details (UML, data flow, etc)
Offline test plan
Online test plan
Configuration data
Scheduling, resources, costs

A desk check is required for every code change

- If the desk checker approves changes
  - Transition issue to “Ready for Test”
- If significant problems were found
  - Add explanation comment to JIRA issue
  - Transition issue to “Desk Check Failed”
- If only minor problems were found
  - Add explanation comment to JIRA issue
  - Transition issue to “Ready for Test”

Code Assessment Criteria
Are requirements and design implemented
Does JIRA documentation reflect changes
Robustness
Conformance to architecture/frameworks
Extendibility, maintainability, simplicity
Proper commenting
Exception handling
Concurrency



Quality CM tools enable reliable deployments

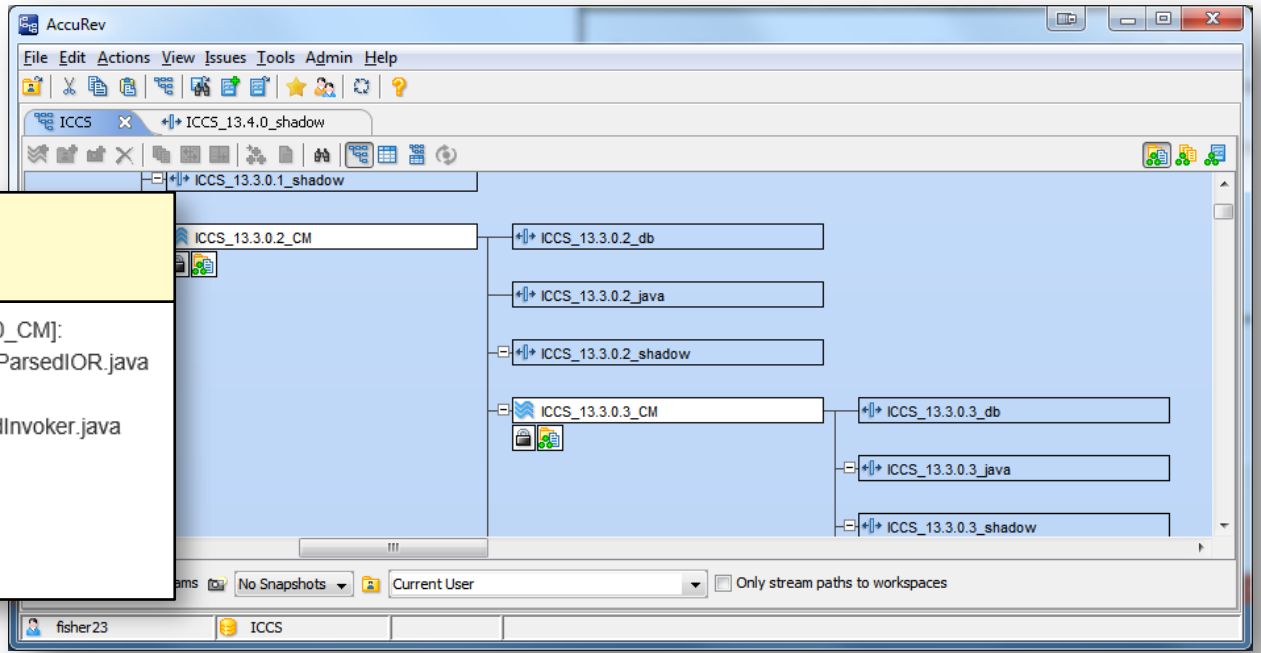
- Most all NIF projects use AccuRev for a version control system
- AccuRev provides high-end capabilities
  - Stream-based architecture
  - Automatic merging and inheriting between code streams
  - Advanced GUI to visualize codebase
- CM Team uses AccuBridge and in-house tools for connecting AccuRev and JIRA

AccuRev details are Stored in JIRA issues

Release Notes:

Modified Java sources [ICCS_13.4.0_CM]:
\.\src\java\src\iccs\base\corba\ICCSParsedIOR.java
ICCS_win32_working_stout6/1
\.\src\java\src\iccs\base\corba\TimedInvoker.java
ICCS_win32_working_stout6/2

Deliverables:
ICCS_FW.jar



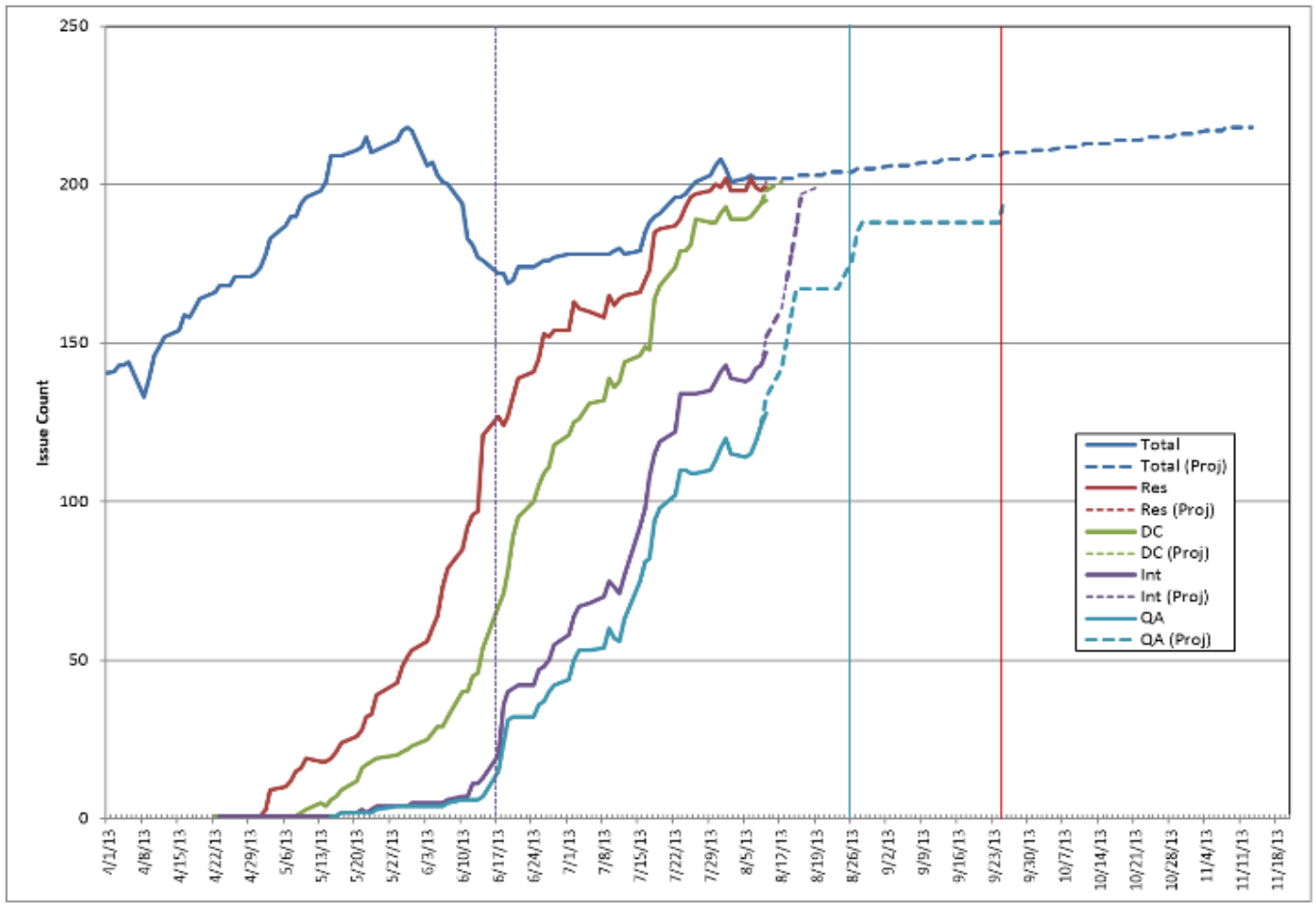
Limitations and Workarounds

- No support for field-level permissions
  - Some fields are exclusively for CM Team, Integration Leads, etc
  - Workarounds:
    - Control access to certain screens through workflow permissions
    - Field-level history of all changes

- Limited tools for managing workload and tracking releases
  - No way to visually trend software release progress over time
  - Limited charting capabilities
  - Workarounds:
    - ICCS developed homegrown tools to address JIRA limitations

ICCS tracks software releases using historical data

- JIRA data is periodically exported to an Excel file for historical trending
- The chart below tracks issue resolution, desk checks, integration testing, and QA for a specific software release



ICCS Managers can browse developer workload

- ICCS leverages Splunk (a “big data” analysis tool) to mine JIRA’s MySQL database
- This dashboard allows interactive investigation of upcoming work
  - Open issues for all active releases
  - Remaining days work by developer
  - All work currently assigned to developer
- See poster THPPC082 for much more on ICCS and Splunk

splunk> listen to your data™

Future Plans with JIRA for NIF

- Upgrade from JIRA 4.1.1 to 6.1
  - Streamlined interface
  - Support for mobile devices
  - Compatibility with newest plug-ins

- Other Upgrades
  - Migrate database from MySQL to Oracle
  - Migrate from bare metal server to an Oracle Virtual Machine
  - AccuRev's AccuSync JIRA server and AccuRev to 6.0


NIF