Code Management in the ATLAS Collaboration

DPF 2015
August 6

Alexander Undrus (BNL)
on behalf of the ATLAS Collaboration
The ATLAS Collaboration at LHC

- 38 countries
- 200 universities and institutions
- 3000 scientific authors
- With up to 3 billion events per year

ATLAS records petabytes of data
ATLAS Offline Computing

ATLAS Offline software is available, used, and developed in hundreds of sites. Main centers are located in 3 continents.

~11 Tier-1 centers

100s Tier-2 and Tier-3 centers
ATLAS Offline Code

- 5.4 million code lines (C++ and python)
- 2200 packages
- 7000 – 8000 repository commits per month
- Most members of the ATLAS Collaboration contributed at least a line

ATLAS code developers in June 2015

New authors (not active 1 year ago)
Authors active 1 year ago
ATLAS Code Management Principles

- Continuous, 50-stream development
  - Supported by the ATLAS Nightly System
- ~10 separately manageable code domains
- Working stable releases are delivered frequently (~400 annually)
  - Include all domains (patches and specials are allowed)
- Multi-stage validation

Key areas for success:

COORDINATION  INFRASTRUCTURE
ATLAS Software Releases

- **Full releases** include all domains
- **Patch releases** contain fixes and updates of packages
- **Analysis releases** provide additions and updates for a specific physics analysis
- **Special releases** (testing new technologies and tools or narrow-purpose releases)
- Patch and Analysis releases are always used with the corresponding base full release

- **Nightly releases** lead up to, and are the basis of, stable releases (~400 per year)
  - 67 parallel nightly branches (as of 07/2015)
Development Streams and Cycles
(a simplified schema)

Each development stream is supported by 1 or 2 nightly release branches

migration stream
(development of new technology)

new technology
is added to main stream

End of stream, reintegration with main stream

main stream of development

bug-fix

releases of patches

special purpose

End of stream, reintegration with main stream

main stream of development

bug-fix

releases of patches

Monthly stable releases

Stable releases every 2-10 days

merges

stable release cut
ATLAS Software Infrastructure

- Includes tools and services for
  - Software storage
  - Code approval
  - Builds
  - Validation
  - Developed mostly by ATLAS members

- Keeps modernizing, abreast with modern database and technologies

- In responsibility of ATLAS Software Infrastructure Team (SIT)
  - Also provides platforms and compilers support, documentation
ATLAS Software Infrastructure Components

**VERSION MANAGEMENT**
- Tag Collector
- SVN code repository

**NIGHTLY SYSTEM**
- ATN testing
- NICOS CONTROL SYSTEM
- up to 70 nightly branches
  - Integration
  - Validation
  - Migration
  - Personal
  - Patch
  - Physics Analysis
  - Experimental

**EXTERNAL TESTING**
- ATLAS Run Time Tester (RTT)

**INSTALLATION SYSTEM**
- CernVM-FS
- CERN AFS
- Distribution KITS

**ATLAS Tier Centers**
ATLAS Tag Collector

- Web interfaced database application
- Allows developers to apply for new SVN tags in developing releases
- Coordinators control all changes
- Works in close cooperation with the ATLAS Nightly System and SVN repository
ATLAS Nightly System

- Capable to support ~100 branches of multi-platform nightly builds
- Runs ~500 tests in the integrated ATN test framework
- Dynamic web interfaces, database driven
- Uses the ATLAS Nightly Build farm with 50 multi-core nodes
- Works in close cooperation with the ATLAS Tag Collector, RTT test framework, and SVN repository
Components of the ATLAS Nightly System

Nightlies Oracle Database

Nightlies Web Server (based on PanDA Web Platform, uses JSON, Ajax, WCGI, JQuery)

Web UI for nightly releases tag content comparison

NICOS Control System \textit{(runs nightly jobs)}

System Status Web UI \textit{(provides dynamic nightly jobs status)}

Nightly Farm Status Web UI \textit{(displays build machines information)}

Web UI for build results research \textit{(compilation, tests, installation)}

Components of the ATLAS Nightly System

DPF 2015 - Alexander Undrus
ATLAS Software Validation

- Two-stage validation in the Nightly System

  - Main Nightly Branch
  - Software Patches
    - Validation Nightly Branch

- Two testing frameworks for nightly releases

  - ATN framework runs ~500 “smoke” tests, results in 3 hours after compilations
  - RTT framework runs longer tests, data processing jobs on a batch farm, results in 12-24 hours after compilations
ATLAS Package Update Workflow

Package Developer

SVN Repository

Tag Collector

Nightly Builds and Tests

Release Coordinator

Stable Release (cut from successful nightly release)

Applies for validation

Inspects results with Web UI

Validates in nightlies, accept or reject

Gets release content

Inspects results with Web UI
Future Development Plans

- Foster collaboration with other HEP experiments
  - ATLAS infrastructure uses mostly home-made tools
  - HEP Software Foundation provides an opportunity to share experience and develop common tools
    - Interfaces to CMAKE
    - GIT code repository
    - Jenkins continuous integration tool
Conclusions

• ATLAS Software Infrastructure allows effective utilization of the knowledge and experience of numerous software developers working around the world and around the clock

• ATLAS developed unique web-based, database-driven collaborative tools for the offline software development