Transition to Continuous Integration Practices in ATLAS Software Development

US ATLAS PS,S&C Planning Meeting

August 1, 2016

Alexander Undrus (BNL)

Scope

- WBS items
  - 5.2.1 – Software Validation
  - 5.2.2 – Librarian and Infrastructure Services
- People
  - Alex Undrus (1 FTE)
  - Shuwei Ye (0.5 FTE)
- Key Infrastructure Systems
  - ATLAS Nightly Build System
  - ATN Nightly Testing Framework
  - AtlasSetup build and runtime environment tool
  - LXR code browser
  - U.S. ATLAS user support

Flow Organization Matters

For traffic throughput . . .

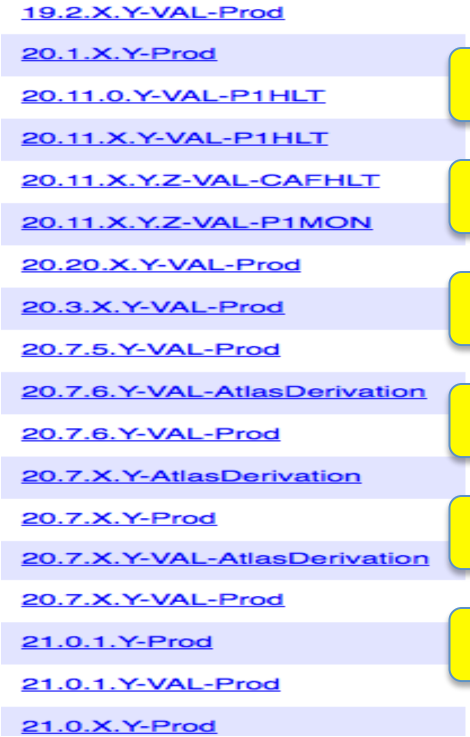
Flow Organization Matters

SVN

TC

AMI

NICOS



19.2.X.Y-VAL-Prod
20.1.X.Y-Prod
20.11.0.Y-VAL-P1HLT
20.11.X.Y-VAL-P1HLT
20.11.X.Y.Z-VAL-CAFHLT
20.11.X.Y.Z-VAL-P1MON
20.20.X.Y-VAL-Prod
20.3.X.Y-VAL-Prod
20.7.5.Y-VAL-Prod
20.7.6.Y-VAL-AtlasDerivation
20.7.6.Y-VAL-Prod
20.7.X.Y-AtlasDerivation
20.7.X.Y-Prod
20.7.X.Y-VAL-AtlasDerivation
20.7.X.Y-VAL-Prod
21.0.1.Y-Prod
21.0.1.Y-VAL-Prod
21.0.X.Y-Prod

COORDINATORS

SHIFTERS

INSTALLERS

TC/AMI ADMIN

MERGE ADMIN

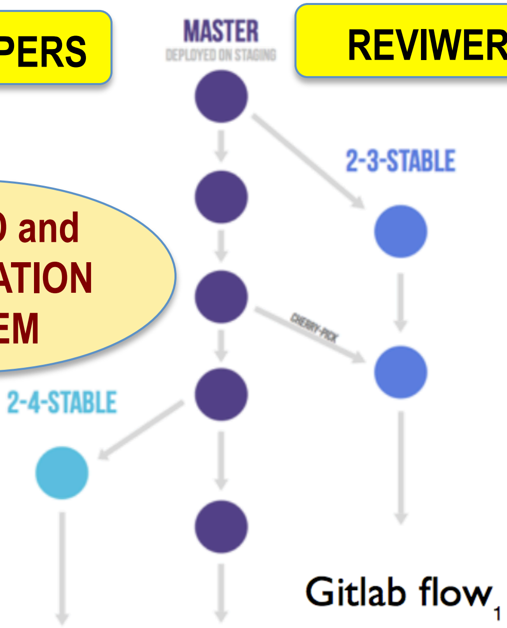
DEVELOPERS

GIT

DEVELOPERS

REVIWERS

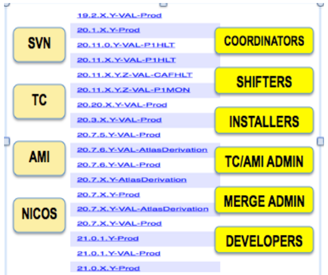
CI BUILD and INSTALLATION SYSTEM

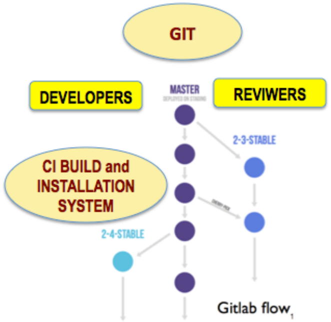


Gitlab flow[1]

. . . and for software development throughput

Flow Organization Matters



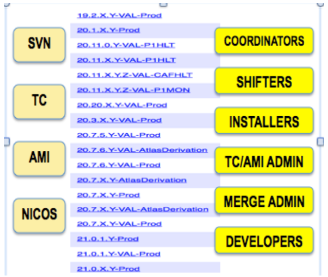


Roundabouts advantage

Pritchard, Michael S.; Harris, Charles E.; and Rabins, Michael J., "Engineering Ethics: Concepts and Cases" (2014)

- Vehicles move non-stop
- Efficient, require less lanes
- Low cost (no signals)

Flow Organization Matters



SVN

TC

AMI

NICOS

19.2.X.Y-VAL-Prod
20.1.X.Y-Prod
20.11.0.Y-VAL-P1HLT
20.11.X.Y-VAL-P1HLT
20.11.X.Y.Z-VAL-CAFHLT
20.11.X.Y.Z-VAL-P1MON
20.20.X.Y-VAL-Prod
20.3.X.Y-VAL-Prod
20.7.5.Y-VAL-Prod
20.7.6.Y-VAL-AtlasDerivation
20.7.6.Y-VAL-Prod
20.7.X.Y-AtlasDerivation
20.7.X.Y-Prod
20.7.X.Y-VAL-AtlasDerivation
20.7.X.Y-VAL-Prod
21.0.1.Y-Prod
21.0.1.Y-VAL-Prod
21.0.X.Y-Prod

COORDINATORS

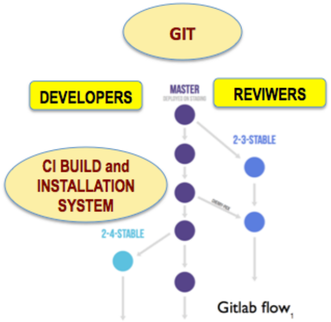
SHIFTERS

INSTALLERS

TC/AMI ADMIN

MERGE ADMIN

DEVELOPERS



GIT

DEVELOPERS

REVIWERS

CI BUILD and INSTALLATION SYSTEM

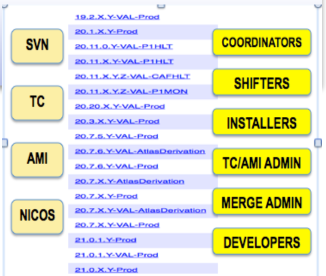
Gitlab flow[1]

Continuous Integration (CI) advantage

(powered by git and Jenkins)

- Teams develop autonomously, non-stop
- Efficient: build/integrate when needed
- Lower cost (less tools, bureaucracy)

Current Software Infrastructure



SVN

TC

AMI

NICOS

19.2.X.Y-VAL-Prod
20.1.X.Y-Prod
20.11.0.Y-VAL-P1HLT
20.11.X.Y-VAL-P1HLT
20.11.X.Y.Z-VAL-CAFHLT
20.11.X.Y.Z-VAL-P1MON
20.20.X.Y-VAL-Prod
20.3.X.Y-VAL-Prod
20.7.5.Y-VAL-Prod
20.7.6.Y-VAL-AtlasDerivation
20.7.6.Y-VAL-Prod
20.7.X.Y-AtlasDerivation
20.7.X.Y-Prod
20.7.X.Y-VAL-AtlasDerivation
20.7.X.Y-VAL-Prod
21.0.1.Y-Prod
21.0.1.Y-VAL-Prod
21.0.X.Y-Prod

COORDINATORS

SHIFTERS

INSTALLERS

TC/AMI ADMIN

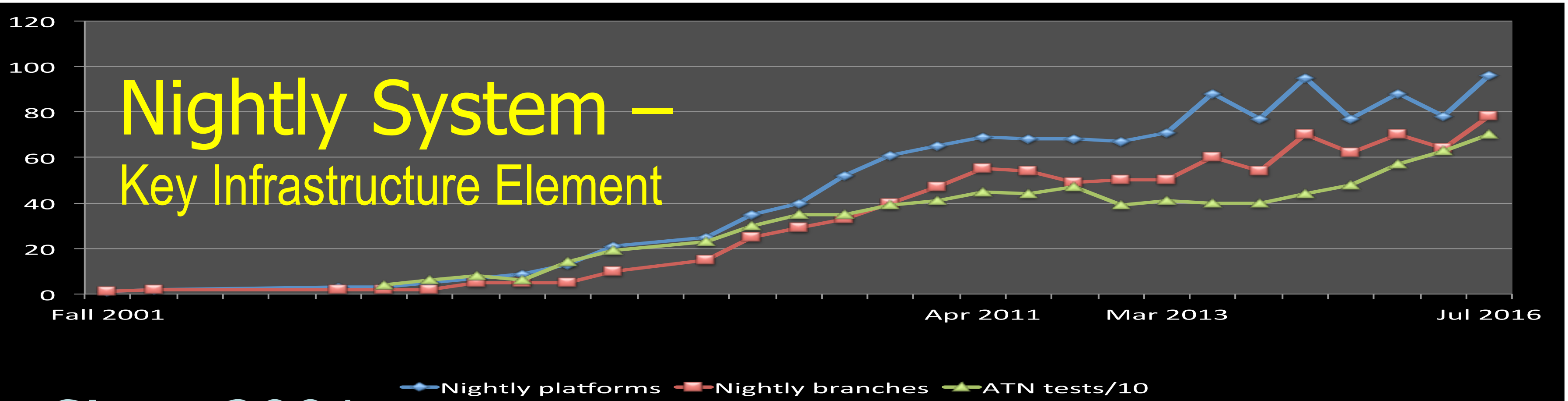
MERGE ADMIN

DEVELOPERS

- Numerous nightly branches (78 in July 2016)
  - Fixed time builds with manual restart option
- Collaborative "work area" based on SVN and TC
  - Complicated tag approval bureaucracy
  - Development line merge is complicated, semi-automatic
- Heavy use of human and hardware resources
  - Substantial SVN and TC admin effort (0.95 SIT FTE)
  - Manual stable release builds by shifters (1.9 SIT FTE)
  - Arduous management of > 2000 packages
  - Fixed job schedule means substantial idle machines time
  - Home-made tools, often without alternative experts

Nightly System –

Key Infrastructure Element



Since 2001

- Developers do integrate their work frequently
  - 78 branches rebuilt daily and/or on-demand
  - Excellent reliability (build failure probability ~ 0.2%)
  - New on-demand functionality is popular, used daily few times
- Oracle-based Web UI (with hot spare backup)
  - Flexible error analysis and monitoring
- 100% home made
  - ~ 60 RH and virtual machines (incl. distscc farm)
- **Frequent updates and modifications: 54 JIRA tickets in 2016 (45 resolved)**

BROOKHAVEN
NATIONAL LABORATORY

AtlasSetup

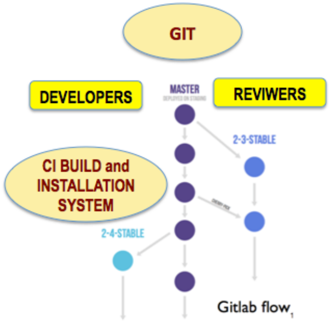
- Build and runtime environment tool
- Used by all developers and in all tools working with ATLAS releases
- Continuous effort of Shuwei Ye to support new tools (CMake), OS (CC7), and compilers (Clang)
- Integral part of ATLASLocalRootBase (ALRB) package that provides site-specific environment at all ATLAS Tier0 – Tier3 sites

16 JIRA tickets opened in 2016 out of which 13 tickets are resolved

LXR

- ATLAS-wide code browsing service provided by Tier-0 center at BNL
  - 2 servers for stable major releases and AthAnalysisBase releases
- Indispensable tool for software developers:
  - Ability to jump easily to the declaration of any global identifier
  - All references to global identifiers are indexed
- Easy but continuous support of Shuwei Ye (0.05 FTE) and Alex Undrus (0.025 FTE)

2 JIRA tickets
opened and resolved in 2016

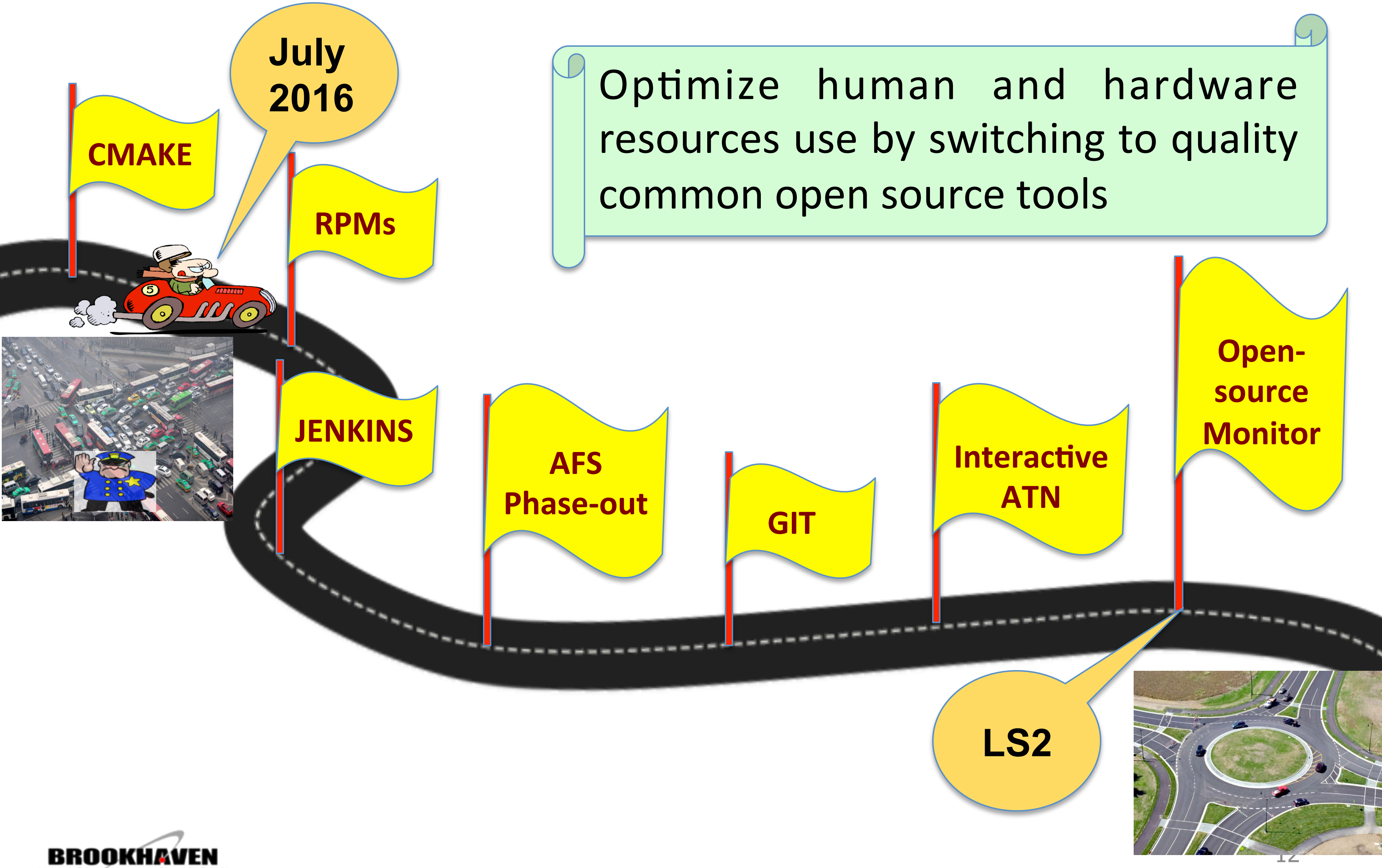


Major Infrastructure Revamp Started

- Inspired by the 2015 Build Review findings:
  - Many current tools are at the End-of-Life (CMT, SVN, AFS, NICOS)
  - Current workflow does not provide modern CI
  - Human and hardware resource use is not optimized
  - Potential of open-source tools is not harnessed
- Among main goals: achieve CI workflow

Roadmap to CI

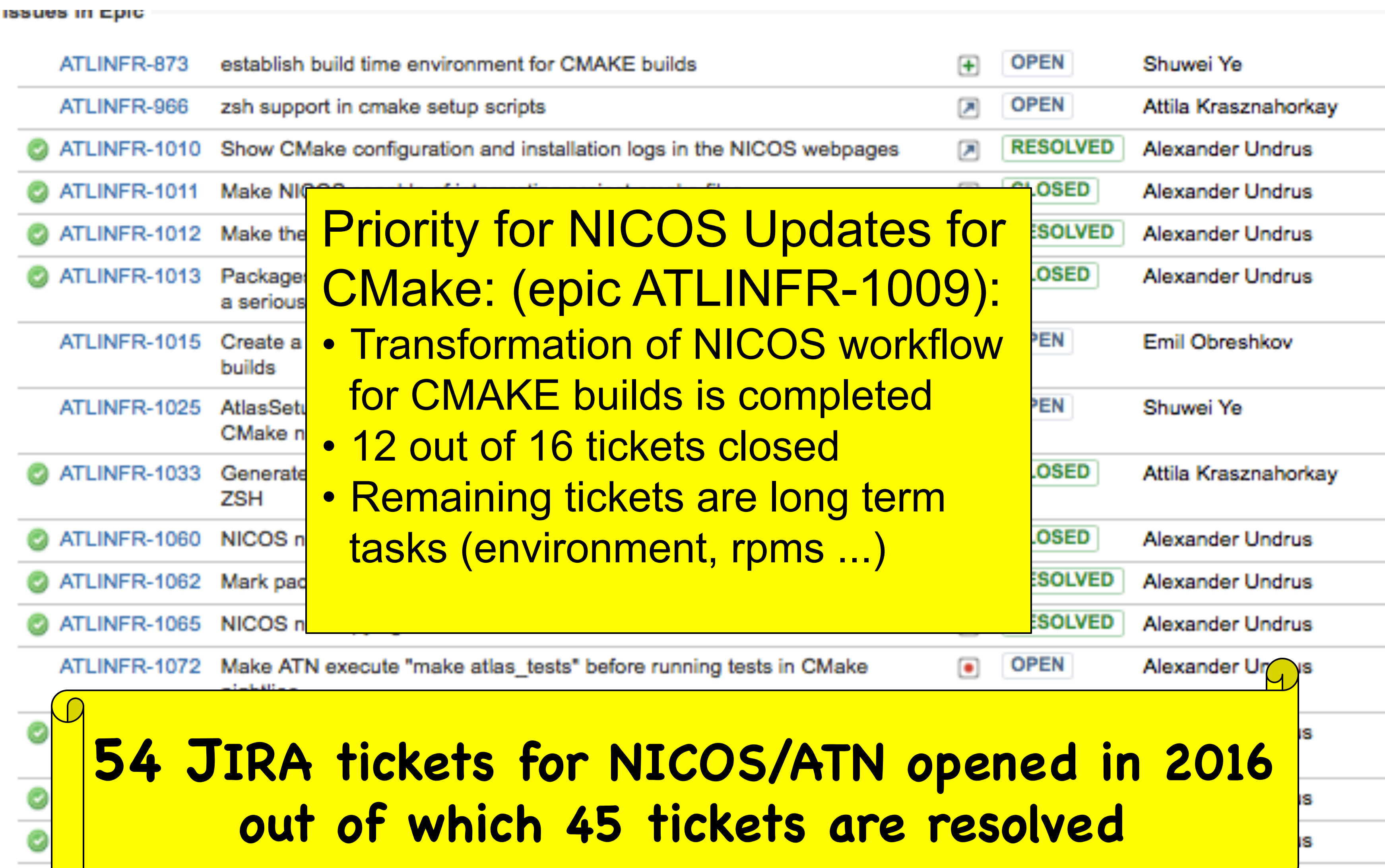
Optimize human and hardware resources use by switching to quality common open source tools



4-phases Plan of the Nightly System CI Upgrade

- Discussed and approved at February 2016 S&C week, https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/AtlasNightliesPlans
  ① Focus on CMake, Jenkins integration (2016)
  ② Put Jenkins on top of the System (2017)
  ③ Focus on new Build Analytics tool (2018)
  ④ Finalization (by LS2)
- Small, incremental changes, iterations
- Detailed plan is essential for a success
  - *Success metrics*
  - *Means of implementation*
  - *Milestones*
  - *Workshops, review panel*

1st Phase Progress (Completion 4Q 2016)



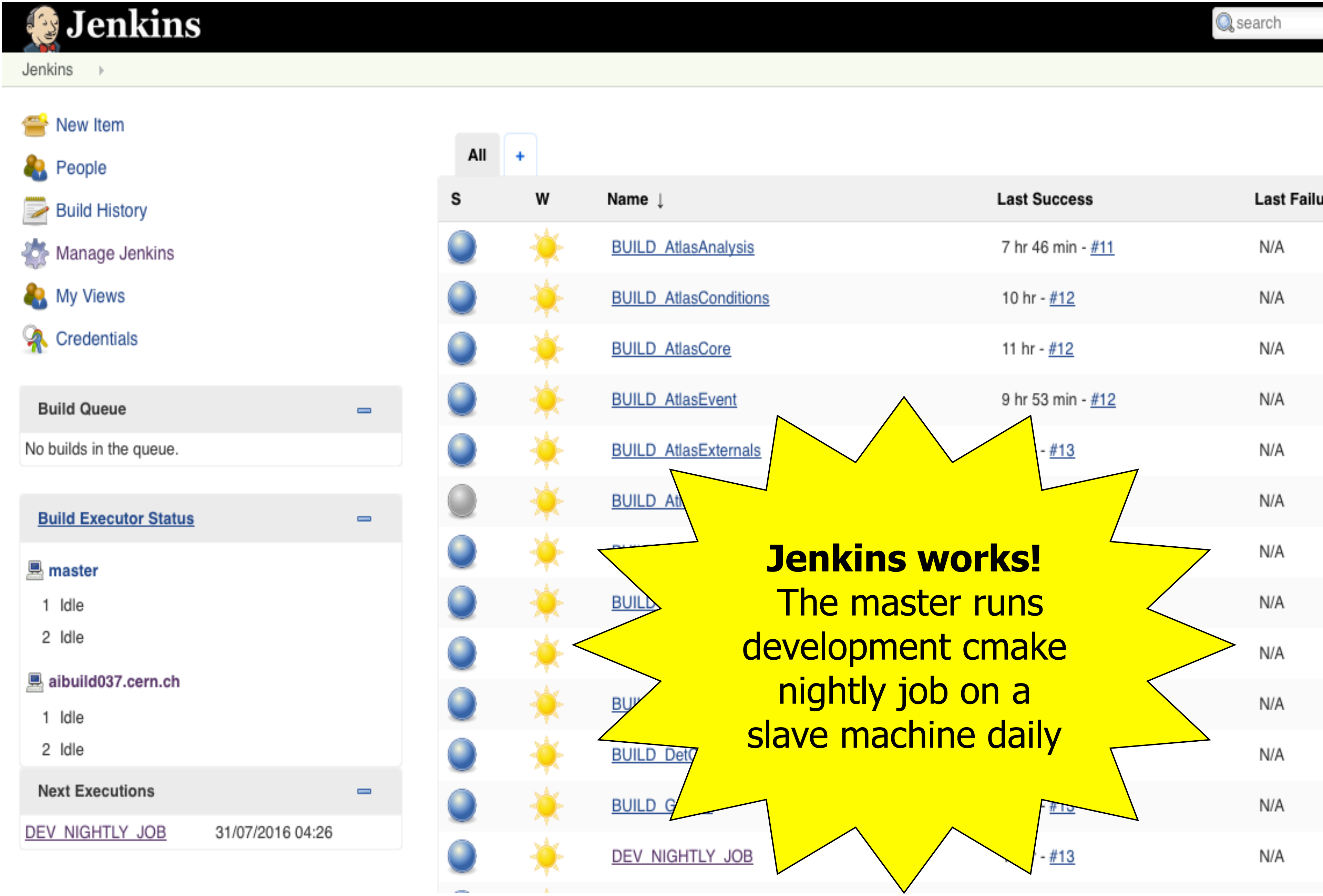
Issues in Epic

ATLINFR-873	establish build time environment for CMAKE builds	⊞	OPEN	Shuwei Ye
ATLINFR-966	zsh support in cmake setup scripts	↗	OPEN	Attila Krasznahorkay
✓ ATLINFR-1010	Show CMake configuration and installation logs in the NICOS webpages	↗	RESOLVED	Alexander Undrus
✓ ATLINFR-1011	Make NIC		CLOSED	Alexander Undrus
✓ ATLINFR-1012	Make the		ESOLVED	Alexander Undrus
✓ ATLINFR-1013	Package a serious		LOSED	Alexander Undrus
ATLINFR-1015	Create a builds		PEN	Emil Obreshkov
ATLINFR-1025	AtlasSetu CMake n		PEN	Shuwei Ye
✓ ATLINFR-1033	Generate ZSH		LOSED	Attila Krasznahorkay
✓ ATLINFR-1060	NICOS n		LOSED	Alexander Undrus
✓ ATLINFR-1062	Mark pac		ESOLVED	Alexander Undrus
✓ ATLINFR-1065	NICOS n		ESOLVED	Alexander Undrus
ATLINFR-1072	Make ATN execute "make atlas_tests" before running tests in CMake nightlies	●	OPEN	Alexander Un s

Priority for NICOS Updates for CMake: (epic ATLINFR-1009):

- Transformation of NICOS workflow for CMAKE builds is completed
- 12 out of 16 tickets closed
- Remaining tickets are long term tasks (environment, rpms ...)

54 JIRA tickets for NICOS/ATN opened in 2016 out of which 45 tickets are resolved



Jenkins

search

Jenkins ›

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

master

1 Idle

2 Idle

aibuild037.cern.ch

1 Idle

2 Idle

Next Executions

DEV_NIGHTLY_JOB	31/07/2016 04:26

All +

S	W	Name ↓	Last Success	Last Failu
●	☀	BUILD_AtlasAnalysis	7 hr 46 min - #11	N/A
●	☀	BUILD_AtlasConditions	10 hr - #12	N/A
●	☀	BUILD_AtlasCore	11 hr - #12	N/A
●	☀	BUILD_AtlasEvent	9 hr 53 min - #12	N/A
●	☀	BUILD_AtlasExternals	- #13	N/A
●	☀	BUILD_Atl		N/A
●	☀			N/A
●	☀	BUILD		N/A
●	☀			N/A
●	☀	BU		N/A
●	☀	BUILD_DetC		N/A
●	☀	BUILD_G	#13	N/A
●	☀	DEV_NIGHTLY_JOB	- #13	N/A

Jenkins works!
The master runs development cmake nightly job on a slave machine daily

1st Phase Items under Development

- ATLAS Jenkins master server
  - Configuration is a **BIG TASK**
  - Frequent Jenkins&plugins updates pose a challenge
  - 3 instances tested at BNL and CERN
  - Instabilities analyzed and tackled
  - Disruptive puppet management was a major problem – solved with CSOPS help
  - Works OK now, behind CERN SSO
- CMake build jobs run on a slave machine daily
  - Project builds are synchronized by Jenkins
  - 100% success on the newest server

Jenkins Caveats

- Key functionalities are provided by plugins
  - ~ 50 active plugins on the ATLAS master
  - Essential "multijob" and "multiplatform" plugins
- Stability is not granted automatically
  - Automatic updates kill all slave jobs
    - *New Jenkins versions ~ twice a week*
    - *Plugins backward compatibility is not guaranteed*
  - Jenkins slave agent crashed occasionally with Jenkins 1.* but is very stable with Jenkins 2.*
  - Build timeout plugin is needed for stopping stalled jobs
- Jenkins puppet module (from CERN IT) is disruptive
  - The latest version is much better

Jenkins: Next Steps

- Security scan, opening ports in the firewall
  - Help of ATLAS CSOPS is essential
- Achieve high stability for the Jenkins-managed test nightly jobs
  - Less than 1 Jenkins hiccup per month
- Assure optimum synchronization of different build steps (no idle periods)
- Switch few CMake-based 'regular' nightly branches to Jenkins scheduling by December 2016
  - Optionally: new nightly release names 'YYYY-MM-DDTTTT'

Moving Nightlies Installation from AFS

- CVMFS seems like a natural choice
- Existing ATLAS nightlies CVMFS server:
  - Can not be regarded as the model for AFS replacement
  - Holds few nightly branches used by few users
  - Data size and load on it is incomparable with that on AFS
- CVMFS and AFS use cases are different
  - CVMFS is a distribution file system
  - Better solution is needed to allow several users trigger installations and publications on a CVMFS server
- Solutions for CVMFS nightlies installations are explored in ATLINFR-1050
  - Several servers with a common base path
  - **NOT FREE: additional support effort seems inevitable** (Brinick Simmons [UK] offered help in testing new CVMFS server)

2nd Phase (completion 4Q 2017)

- Continue to add cmake-based builds to Jenkins
- Retire RH build machines
- Use VM SSD machines for builds
  - . . . Or 32-core RH supercomputers with superfast SSD
  - Reduce number of CPUs ~ 30% at the build farm
- Automate creation of nightly jobs for new nightly branches (possibly with a special Jenkins plugin)
- Keep the Nightly System in sync with ATLAS transition from SVN to git
  - Transition schedule with dated milestones would be very helpful
  - Evaluate Jenkins plugin for git SCM
- Evaluate incremental capabilities of CMake builds

3rd Phase (completion 4Q 2018)

- Evaluate and choose an open source tool for nightlies monitoring
- Move cmt and RootCore based builds to Jenkins
- Support git SCM as needed
- Deliver a beta version of the new ATN framework allowing testing on remote farms and in individual developers sessions

4th Phase (completion by LS2)

- ✧ Remove dependence of the ATLAS Nightly System on AFS
- ✧ Replace the NICOS Web UI with open source monitoring tool
- ✧ Finalize the support of the workflow based on git SCM
- ✧ Deliver beta-version of the new ATN framework

Conclusions

- U.S. ATLAS provides crucial services for the ATLAS software infrastructure (nightly build and testing system, code browsing, environment setup) and quality support for U.S.-based physicists
- These critical support and development services are efficiently provided at 1.5 FTE cost (compare with ATLAS SIT effort 8 FTE)
- Good progress achieved in the ongoing 1st phase of the CI Nightly System upgrade

Backup Slides

Continuous Integration (CI) Definitions

- **Continuous Integration** is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible*(http://www.martinfowler.com/articles/continuousIntegration.html, 2006)*
- **Integration** is the act of submitting your personal work (modified code) to the common work area ("Learning Continuous Integration with Jenkins", by Nikhil Pathania, 2016)
- CI was first named and proposed by Grady Booch in "*Object Oriented Design: With Applications*" (1991), although Booch did not advocate integrating several times a day (*https://en.wikipedia.org/wiki/Continuous_integration*)