

ATLAS Nightly System Evolution and Jenkins Evaluation

ATLAS Software Technical Meeting 2015

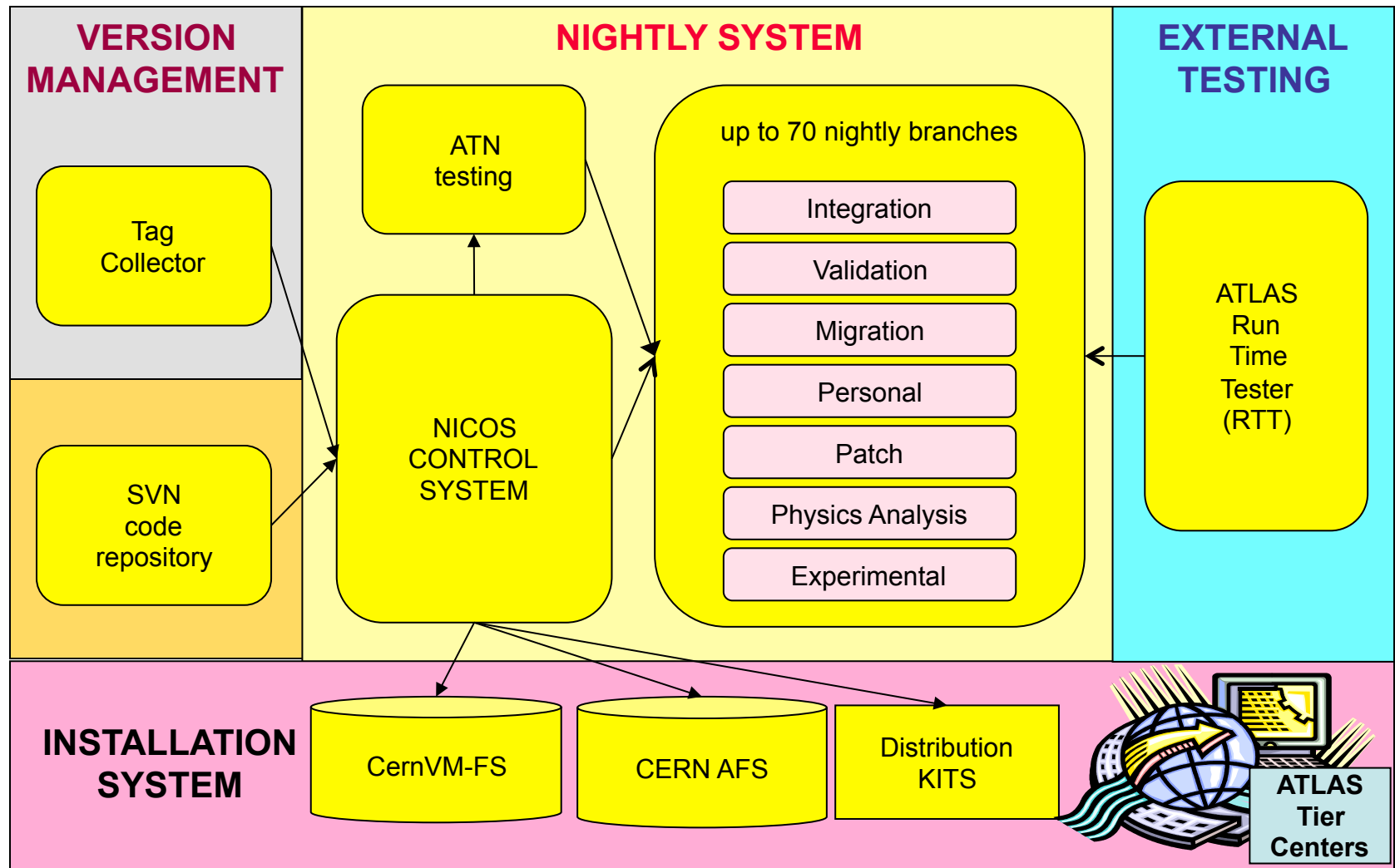
November 11

Alexander Undrus (BNL)

ATLAS Offline Software Facts

- 5.4 million code lines (C++ and python)
- 2200 packages
- 7000 – 8000 monthly commits
- 400 active developers
- Continuous 50 streams of development
- ~10 separately managed code domains
- Working stable releases are delivered frequently (~400 annually)
 - Include all domains (patches and specials are allowed)
- Multi-stage validation

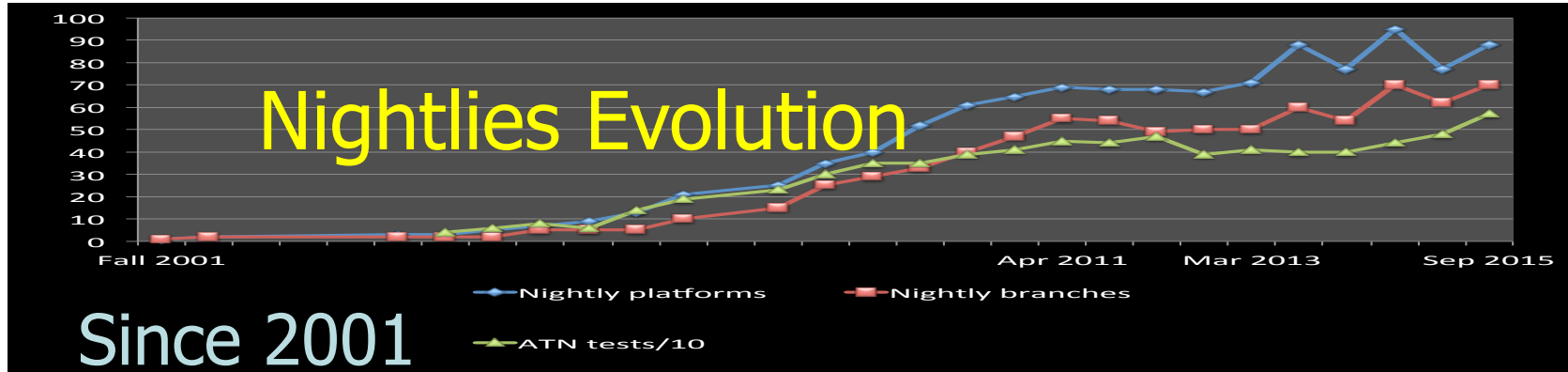
ATLAS Software Infrastructure Components



Branch	Rel	# Platf/ Proj	TC Rel and Tags	TC	Build	on AFS	KIT INST RPM KV	CVMFS KV	Ave. Comp. Err. (w/warn)	Ave. Test OK (no warn),%
DEVELOPMENT GROUP:										
20.X.Y	rel 1	2/12	20.9.0							
20.X.Y-VAL	rel 1	2/12	20.9.0							
ROOT6 INTEGRATION GROUP:										
20.6.X	rel 1	2/12	20.6.1							
20.6.X-VAL	rel 1	1/12	20.6.1							
20.7.X	rel 1	1/12	20.7.0							
20.7.X-VAL	rel 1	2/12	20.7.0							
TIER0 GROUP:										
20.1.X	rel 1	2/11	20.1.7							
20.1.X-VAL	rel 1	2/11	20.1.7							
TRIGGER_RELEASE GROUP:										
20.2.X	rel 1	2/12	20.2.3							
20.2.X-VAL	rel 2	1/4	20.2.3							
SIMULATION GROUP:										
20.3.X-VAL	rel 1	1/11	20.3.4							
19.2.X_BUGFIX GROUP:										
19.2.X-VAL	rel 1	1/11	19.2.5							
PATCH GROUP:										
17.2.X.Y-VAL-Prod	rel 2	1/1	17.2.14.12							
19.2.X.Y-VAL-Prod	rel 1	1/1	19.2.4.8							
20.1.5.Y-VAL-Prod	rel 1	1/1	20.1.5.13							
20.1.X.Y-AtlasDerivation	rel 2	1/1	20.1.6.1							
20.1.X.Y-Prod	rel 2	1/1	20.1.6.2							
20.1.X.Y-VAL-AtlasDerivation	rel 1	1/1	20.1.6.1							
20.1.X.Y-VAL-Prod	rel 1	1/1	20.1.6.2							
20.2.X.Y-VAL-P1HLT	rel 1	2/1	20.2.1.7							
20.2.X.Y.Z-VAL-CAFHLT	rel 1	2/1	20.2.1.6.2		03-AUG 05:44	03-AUG 05:54			0 (2.0)	83 (22)
20.2.X.Y.Z-VAL-P1MON	rel 1	1/1	20.2.1.5.2		02-AUG 23:43	03-AUG 00:02			0 (23)	20 (0)
20.2.X.Y.Z-VAL2-CAFHLT	rel 1	2/1	20.2.1.6.2		03-AUG 14:01	03-AUG 14:07			0 (2.0)	83 (22)
20.3.0.Y-VAL-Prod	rel 1	1/1	20.3.0.4		03-AUG 07:03	03-AUG 10:48			0 (28)	61 (60)

ATLAS Nightly System

- **Key part** of software organization
- Includes job control (NICOS), testing framework (ATN), Web server, Database (Oracle based), CVMFS server
- 70 branches of multi-platform nightlies
- Run 4900 synchronized ops streams daily
- ~500 ATN tests
- Dynamic web interfaces, database driven
- Log mining mechanisms
- Mail facility
- Uses the ATLAS Nightly Build farm with 50 multi-core nodes
- Closely linked with the ATLAS Tag Collector, RTT test framework, and SVN repository
- **100% home-made**



- Defined by 'traditional' athena development
 - Single OS, single build site
 - Multi-stream development
 - All-inclusive releases
 - Fixed build tools (CMT, SVN)
- Result: perfect fit for 'traditional' athena
 - Reliable and fast builds
 - Web UI tailored for 'traditional' development schemes

New development trends

- Diverse build tools (RootCore, cmake)
- Multiple release purposes and types
- Various architectures (ARM 64bit, PowerPC)

Ways for the System to accommodate

- Adapt the existing in-house system
 - Increasing complexity, support effort
 - In-use web techniques are aging
- Make use of high quality third-party integration and build tools
 - Share expertise with other experiments through participation in the HEP Software Foundation

Benefits of third-party tools

- Expertise
 - Larger pool of committers
- Stability
 - All possible problems in various environments are encountered and documented
- Learning
 - Communication with top-notch developers helps to advance professional growth


→ cnt'd

Benefits of third-party tools (cnt'd)






- New functions
 - Often come automatically, as pleasant surprises
- Support
 - Varying level, sometimes can be purchased
- **Money savings**
 - Cost of development can be much higher than the cost of evaluation and deployment
 - Funding agencies may be interested in the support of common tool development (e.g. HEP Software Foundation projects)

Jenkins:

A third-party tool with all these benefits



 **Jenkins** **TEST INSTANCE AT BNL**







Jenkins >

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

Build Queue
No builds in the queue.

Build Executor Status

 master
1 Idle
2 Idle
 jenkins04.usatlas.bnl.gov (offline)

S	W	Name ↓	Last Success
		checkout_continuous	18 min - #3
		checkout_test	6 min 49 sec - #10
		helloworld	6 min 55 sec - #1350

- "Continuous integration tool"
 - Written in Java
 - Development started in 2004 (as "Hudson")
 - Comes with a web application
- Rich functionalities
 - Detailed job definitions, many synch. options
- Vast customization possibilities
 - Plugins allow integrating with build tools, databases, etc. (as in NICOS)
 - 100s plugins available in Jenkins dashboard

Jenkins master-slave pair at BNL: Easy and fast installation



Jenkins

New Item
People
Build History
Manage Jenkins
Credentials

Build Queue
No builds in the queue.

Build Executor Status

master	
1	Idle
2	Idle

jenkins04.usatlas.bnl.gov
(offline)

S	W	Name ↓	Last Success
		checkout_continuous	18 min - #3
		checkout_test	6 min 49 sec - #10
		helloworld	6 min 55 sec - #1350

- "Global" panel
 - Executors status
 - Jobs status and results (exit code based)
 - Immediate start buttons
- Job steps
 - Jobs can be grouped (under tabs)
 - 3 kind of job steps (checkout, build, post-build)
 - No synchronization between steps of different jobs
 - Plugin for step extension available

Jenkins master-slave pair at BNL: Project panel with build history and workspace links



Jenkins

TEST INSTANCE AT BNL

search

Jenkins ▶ checkout_continuous ▶

- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Delete Project](#)
- [Configure](#)
- [Subversion Polling Log](#)

Project checkout_continuous

Continuous Checkout Test



[Workspace](#)



[Recent Changes](#)

Permalinks

- [Last build \(#3\), 3 days 8 hr ago](#)
- [Last stable build \(#3\), 3 days 8 hr ago](#)
- [Last successful build \(#3\), 3 days 8 hr ago](#)



Build History

[trend](#) —

- | | |
|-----------|----------------------|
| #3 | Nov 5, 2015 1:28 PM |
| #2 | Nov 5, 2015 1:13 PM |
| #1 | Nov 5, 2015 11:36 AM |



[RSS for all](#)



[RSS for failures](#)

Jenkins master-slave pair at BNL: Flexible polling mechanisms



TEST INSTANCE AT BNL

Jenkins > checkout_continuous > Subversion Polling Log

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Subversion Polling Log](#)

Subversion Polling Log

Started on Nov 5, 2015 1:46:00 PM

Received SCM poll call on for checkout_continuous on Nov 5, 2015 1:46:05 PM

<https://svn.cern.ch/repos/atlasoff/AtlasTest/NightlyTestTools> is at revision 7

Done. Took 8.4 sec

No changes

Build History

[trend](#)

- [#3](#) Nov 5, 2015 1:28 PM
- [#2](#) Nov 5, 2015 1:13 PM
- [#1](#) Nov 5, 2015 11:36 AM

[RSS for all](#) [RSS for failures](#)

Jenkins master-slave pair at BNL:

First test results

Installation	✓ Easy and Fast (10 min, requires port 8080 opening)
First job setup (shell script)	✓ Easy and Fast (5 min)
Setup of job with SVN checkout and polling	✓ Easy and Fast (15 min)
Adding a master-slave connection (ssh method)	✗ Difficult (firewall barriers, > 1 hour, abandoned)
Scheduling dependent jobs	✓ Easy and Fast (1 min)
Stability	✓ Excellent: no problems in 2 weeks
Updates	✓ Automatic
Learning curve	✓ Fast: intuitive web design, all features are documented in pop-up hints

Jenkins and NICOS comparison

	NICOS	JENKINS
Modular design	YES	YES
Needed module kinds?	YES	NO
Fixed time schedule	YES	YES
On-demand	YES	YES
Continuous integration	NO	YES
Multiple OS support	LIMITED	YES
Detailed result analysis	YES	NO (but possible via plugins)
Integrated testing	YES	NO (but in synchronized jobs)
Dynamic Web UI	YES	YES
Personalized emails	YES	NO (but sends job 'net result')
Database fortified	YES	NO
Home-made	YES	NO
Used by other teams	NOT POSSIBLE	ALICE, CMS, ROOT, LCG ...
Plugin library	NO	YES
Support effort (CSOPS help including)	~0.7 - 1.0 FTE	~0.2 FTE (without plugins)

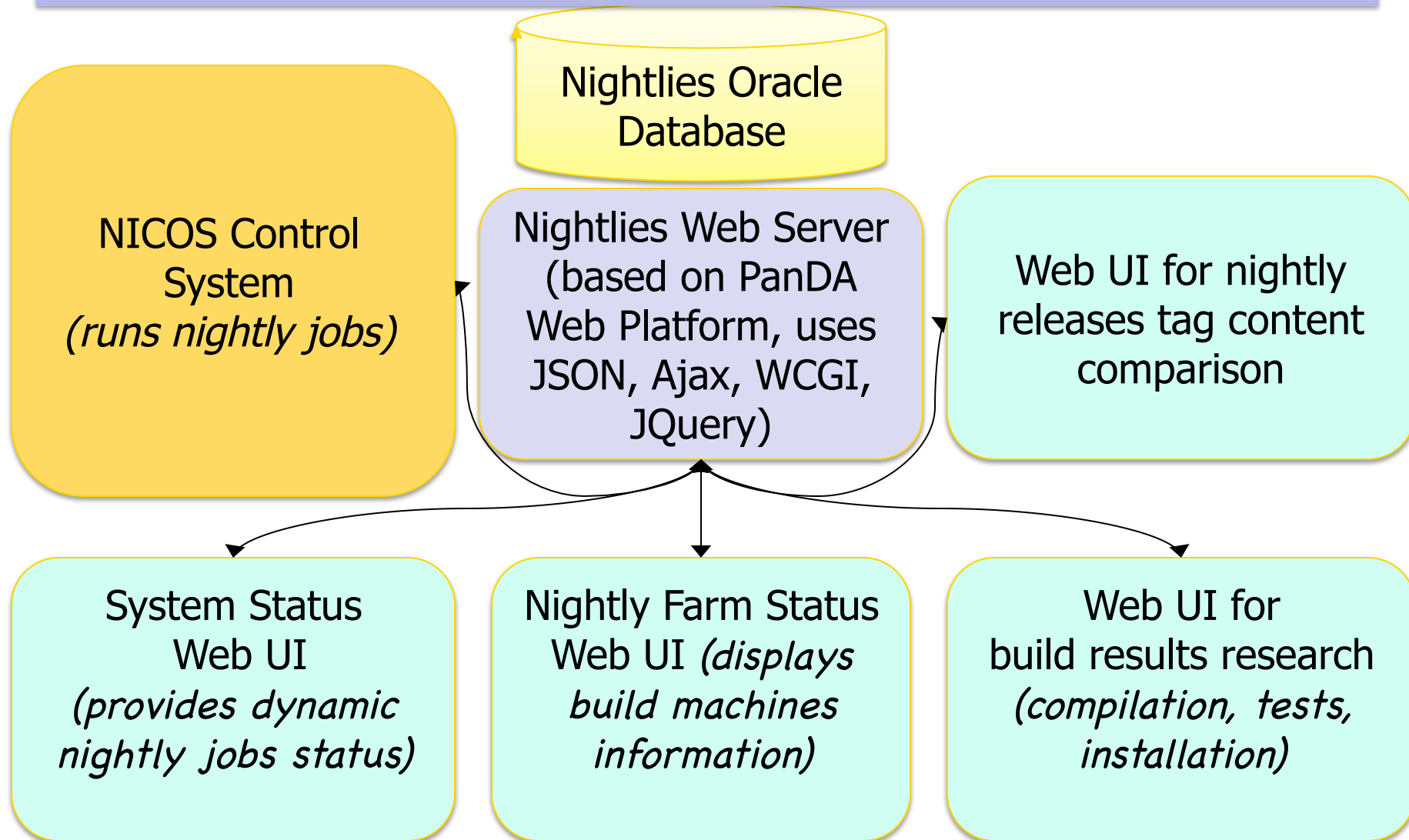
Reasons to switch to Jenkins

- Continuous integration
- Supports distributed builds
 - Remote nodes
 - Various architectures
 - ssh or Java Web Start (JNLP protocol) methods to start slave agents
- De-facto common tool
 - HEP Software Foundation provides an opportunity to share experience and develop common plugins
 - Easier to get a CERN IT support
 - Essential (puppet, firewall setup for bi-directional master-slave communications)
- Eventual **support effort reduction**

Changeover to Jenkins

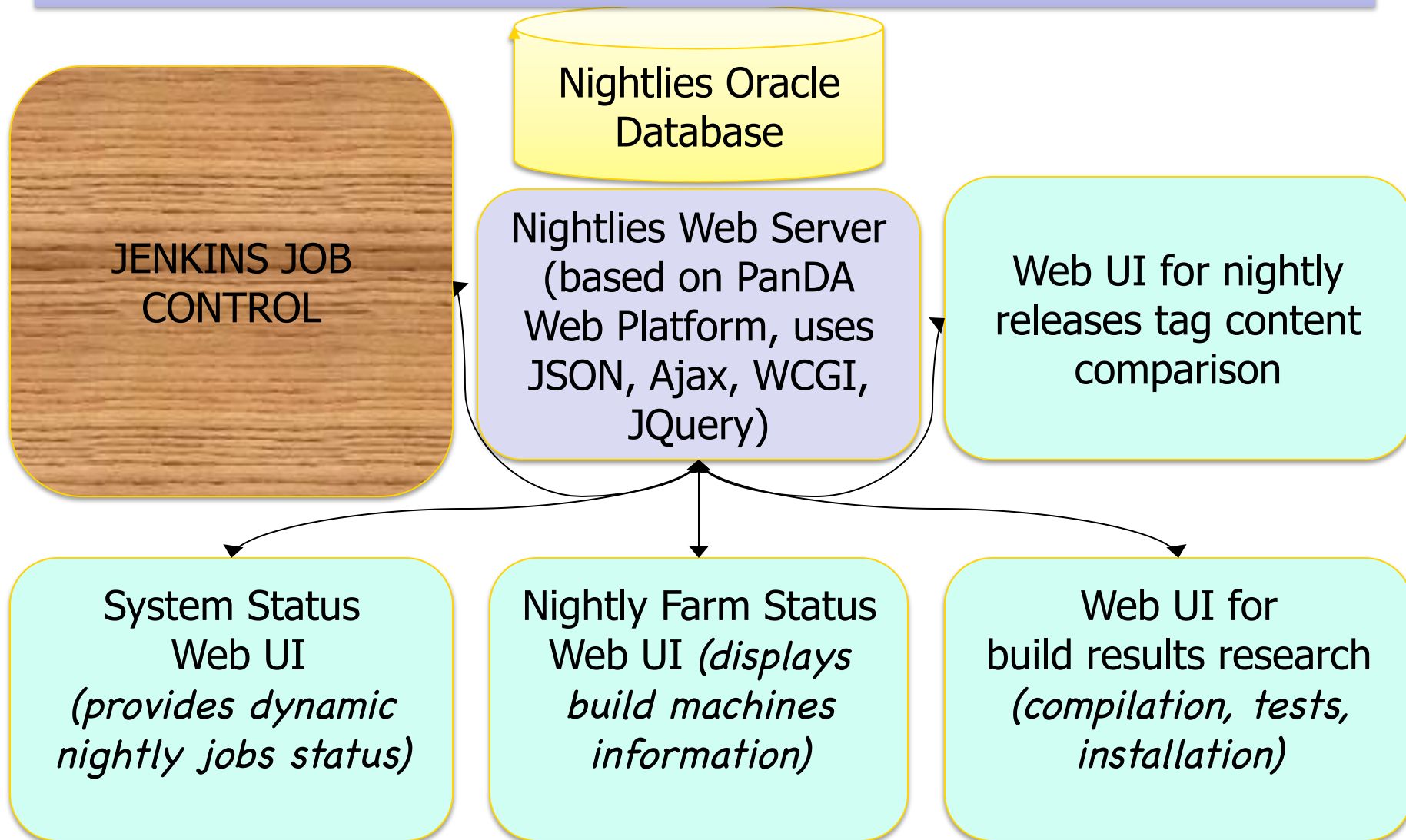
- Small, incremental changes, iterations (an agile development principle)
- Start with replacing of cron with Jenkins scheduling
- Should be in the list of high priority tasks (<https://twiki.cern.ch/twiki/bin/view/Atlas/AuthorShipCommittee>)
- Detailed plan is essential for a success
- Project completion by Run 3 start is desirable

Components of the ATLAS Nightly System: **NOW**



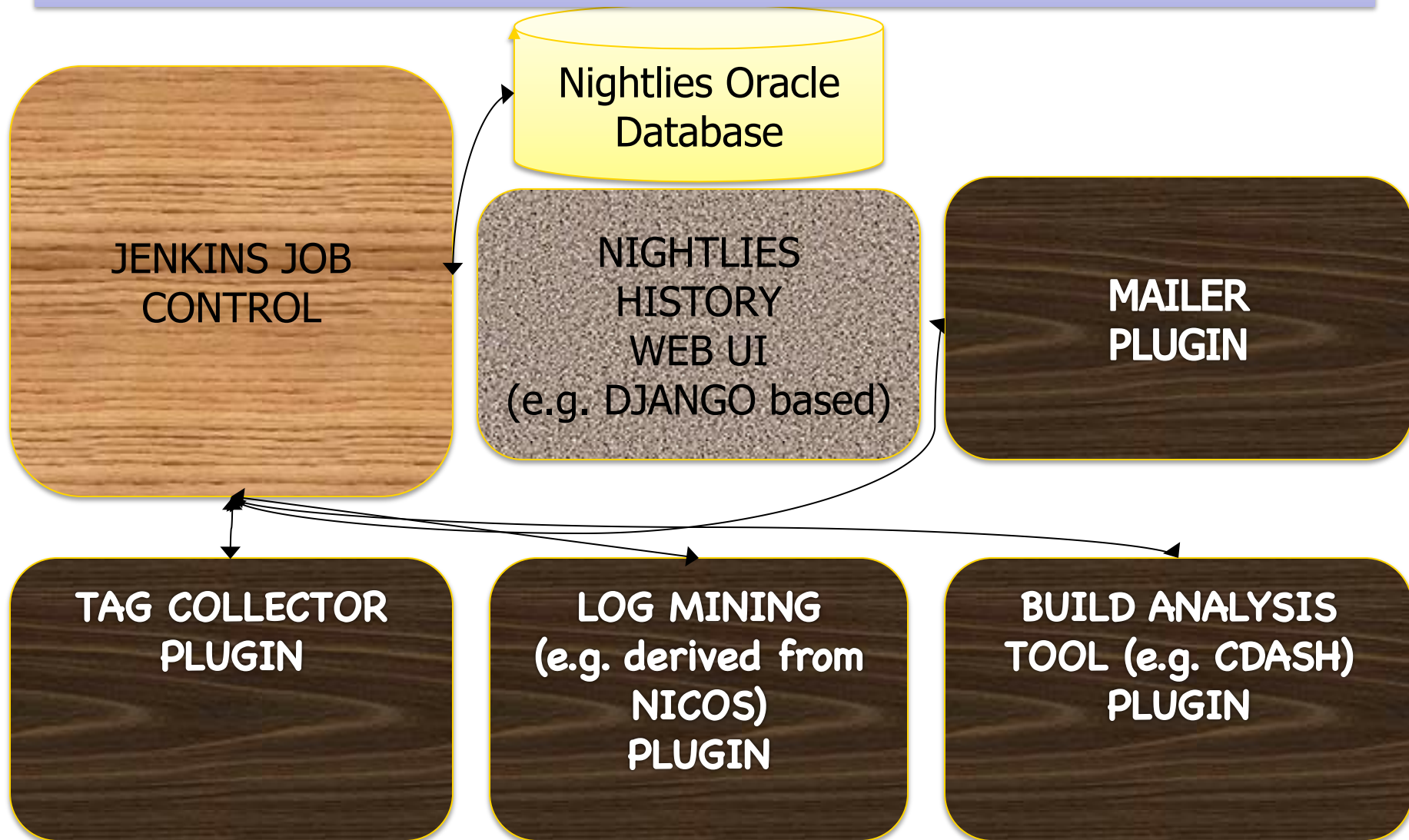
Components of the ATLAS Nightly System:

TRANSITION PHASE



Components of the ATLAS Nightly System:

TRANSITION COMPLETE



Conclusions

- ATLAS Nightly System works well in the current collaborative development paradigm
- Jenkins tool incorporation in the System promises multiple advantages
 - Continuous integration support
 - Multi-architecture builds
 - Rich collection of plugins
 - Expertise sharing opportunities through HEP Software Foundation participation