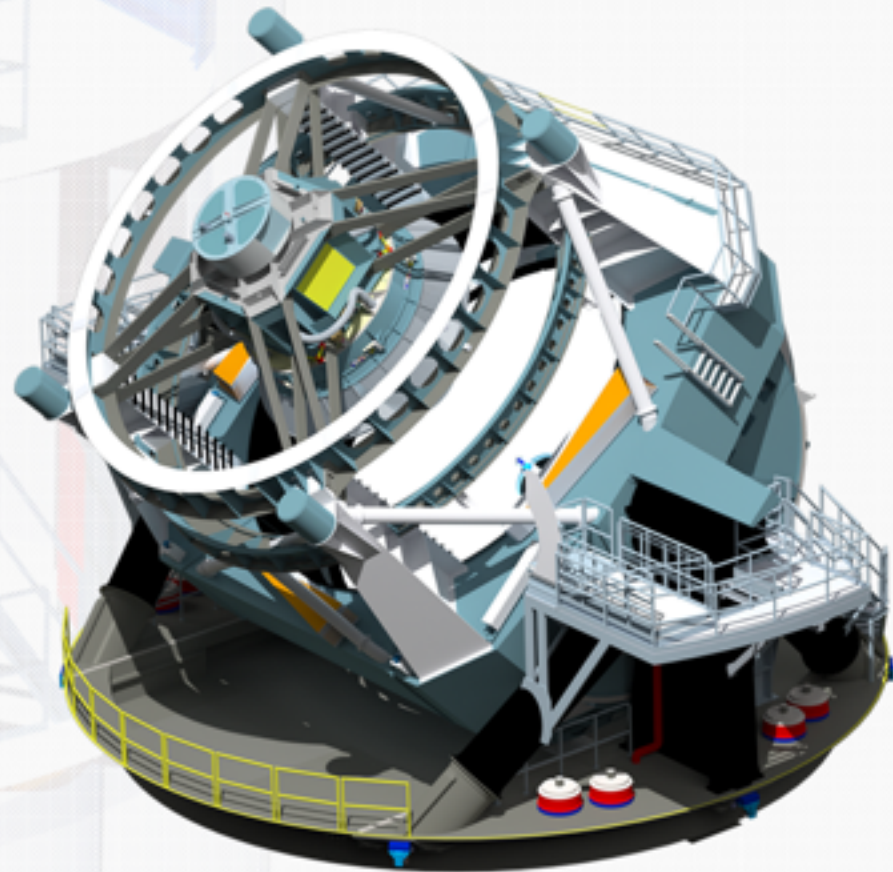


LSST Data Management Software Development

Tim Jenness

DM Deputy System Architect

September 12th 2016



What is the LSST?



- 8.4m (6.7m effective aperture) optical telescope with a 3.5-degree diameter field-of-view, a 3.2 billion pixel camera and 6 broad-band, optical filters
- A data facility (at NCSA) that will process, archive and distribute survey images, associated transient alerts, and calibrated catalogs, as well as calibration and other metadata.

We will deploy this system for a 10 year, time domain survey covering > 18,000 square degrees of the southern sky.

Products to be Delivered by the LSST Project



- A stream of ~10 million time-domain events per night, detected and transmitted to event distribution networks within 60 seconds of observation.
- A catalog of orbits for ~6 million bodies in the Solar System.
- A catalog of ~37 billion objects (20B galaxies, 17B stars), ~7 trillion single-epoch detections (“sources”), and ~30 trillion forced sources, produced annually, accessible through online databases.
- Deep co-added images.
- Services and computing resources at the Data Access Centers to enable user-specified custom processing and analysis.
- Software and APIs enabling development of analysis codes.

Nightly
(Level 1)

Annual
DRS
(Level 2)

Added Value
(Level 3)



- Data Management (DM) team tasked with delivering the data management system:
 - Transport of data from summit to archive facility.
 - Data archiving system.
 - Alert production system.
 - Annual data releases.
 - Data Access Centers.
 - Data processing and analysis software.
- Distributed team of ~ 80 developers with 6 main sites:
 - Tucson, AZ
 - NCSA
 - University of Washington
 - Princeton University
 - SLAC
 - IPAC

Work Packages

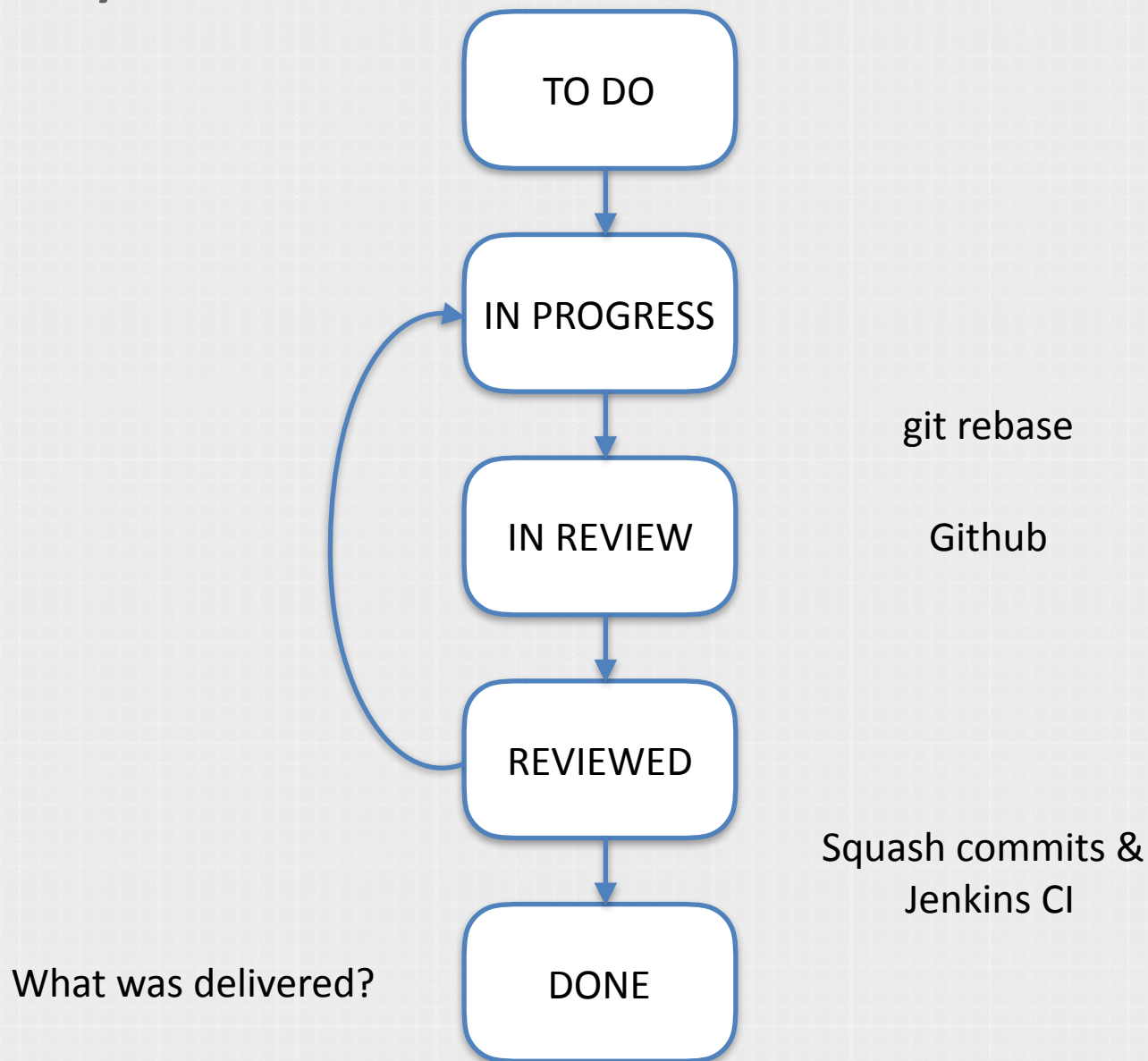


- Give sites self-contained work packages: database, process middleware, data access, science pipelines, developer support tools.
- Science pipelines split across two main sites but with distinct WBS deliverables.
 - They are also highly dependent on process middleware and data access abstractions.
- Agile development practices within 6 month cycles but using Earned Value Management to track overall progress (see [Kantor et al 2016, Proc. SPIE](#)).
- All teams use the same development methodology and share coding standards.
 - Support Python 3 (2.7 optional), and C++11.



- Work for a 6 month cycle is categorized into epics associated with a defined deliverable.
- Epics are associated with particular teams and a WBS.
- Epics are made up of stories consisting of small self-contained tasks that can be scheduled in a sprint.
- Sprints last 3 to 4 weeks.
- Stories and epics are tracked using JIRA.
- Each epic in JIRA describes the overview of the work and what the deliverables are.
- Each story in JIRA contains a description of the task and deliverable.
- More details in DMTN-020: <https://dmtn-020.lsst.io>

Life of a Story



Data Management / DM-6179
Support Python 3 migration

[Edit](#)
[Comment](#)
[Assign](#)
[More](#)
[To Do](#)
[In Progress](#)
[Workflow](#)
[Email](#)
[Export](#)

Details

Type: **Epic** Status: **DONE** (View Workflow)

Priority: **Major** Resolution: **Done**

Component/s: **Developer Infrastructure** Fix Version/s: **None**

Labels: **None**

Epic Name: **Python3-F16-Arch**

Story Points: **40**

WBS: **02C.02.02**

Team: **Architecture**

Cycle: **Fall 2016**

Description

Support the migration of the DM code to Python 3. This includes writing transition documentation, integration of a new scons, migrating a handful of low-level packages and liaising with the teams on their packages.

The final outcome of this epic is that everything would be in place for the migration at the August All Hands meeting.

Stories in Epic (DM Custom)

| Key | Summary | Story Points | Assignee | Status |
|---------|--|--------------|-------------|--------|
| DM-7163 | pex_config should not mandate print function | 0.1 | Tim Jenness | DONE |
| DM-7337 | ChoiceField requires unicode strings | 0.2 | Tim Jenness | DONE |
| DM-1972 | upgrade SWIG to 3.0.8 or later | 2 | Tim Jenness | DONE |
| DM-6312 | Update Scons to v3.0 | 4 | Tim Jenness | DONE |
| DM-6313 | Create miniconda3 EUPS package | 1 | Tim Jenness | DONE |
| DM-6314 | Port lsstsw to Python 3 | 5 | Tim Jenness | DONE |
| DM-6315 | Write Python 3 porting guide (SQR-014) | 10 | Tim Jenness | DONE |
| DM-6316 | Update newinstall.sh to support Python 3 | 1 | Tim Jenness | DONE |
| DM-6317 | Update developer guide to include Python 3 | 2 | Tim Jenness | DONE |

People

Assignee: **Tim Jenness**

Reporter: **Tim Jenness**

Watchers: **Pim Schellart, Tim Jenness**

Votes: **0**

Watchers: **2** [Stop watching this issue](#)

Dates

Created: **23/May/16 7:35 AM**

Updated: **2 days ago**

Resolved: **4 days ago**

Epic Panel

Completed Issues **27 / 27**

Story Points **49.8 / 49.8**

Development

[Create branch](#)

Agile

[View on Board](#)

HipChat discussions

Dedicated room: [Create a room](#) [Choose a room](#)

Other rooms: [Issue mentioned in 5 rooms](#)

Data Management / DM-7026 Port daf_base to Python 3

Buttons: Edit, Comment, Assign, More, To Do, In Progress, Workflow, Email, Export

Details

| | | | |
|---------------|------------------|----------------|-----------------------------|
| Type: | Story | Status: | DONE (View Workflow) |
| Priority: | Major | Resolution: | Done |
| Component/s: | daf_base | Fix Version/s: | None |
| Labels: | None | | |
| Story Points: | 0.5 | | |
| Epic Link: | Python3-F16-Arch | | |
| Team: | Architecture | | |

People

| | |
|------------|--------------------------------------|
| Assignee: | Tim Jenness |
| Reporter: | Tim Jenness |
| Reviewers: | Paul Price |
| Watchers: | J Matt Peterson, Paul Price, ... (1) |
| Votes: | 0 |
| Watchers: | Stop watching this issue |

Dates

| | |
|-----------|-------------------|
| Created: | 26/Jul/16 1:22 PM |
| Updated: | 2 days ago |
| Resolved: | 05/Aug/16 1:17 PM |

Description

Changes necessary to get daf_base to work with Python 3.

Issue Links

| | | |
|---------------|---|---------|
| blocks | DM-7416 Port dax_imgserv to Python 3 | ↑ TO DO |
| is triggering | DM-7141 PropertyList Int type should be defined as 32-bit | ↑ TO DO |
| mentioned in | Python 3 Porting progress | |

Activity

Activity tabs: All, Comments, Work Log, History, Activity, Emails, CI Builds

- Tim Jenness added a comment - 04/Aug/16 9:07 AM

Please would you review?

- Paul Price added a comment - 04/Aug/16 10:18 AM

Nothing major. Whitespace and style commits can probably all be squashed, and the SWIG `list` commits should be squashed.

- Tim Jenness added a comment - 04/Aug/16 10:54 PM

Paul Price Can you take a quick look at the `baseLib.1` changes again please? I reworked them a bit and fixed a bad bug (was not checking for negative numbers, was not handling bool and was treating int as 64bit!). Now passes Jenkins.

- Paul Price added a comment - 05/Aug/16 8:40 AM

Had another look, and made comments on the GitHub PR.

- Tim Jenness added a comment - 05/Aug/16 1:17 PM

Thanks for the great review. All changes implemented and extra tests added. Can build `lsst_apps` ok now.

Development

| | |
|----------------|---------------------------|
| 1 branch | |
| 1 commit | Latest 05/Aug/16 1:16 PM |
| 1 pull request | Updated 05/Aug/16 1:16 PM |

Create branch

Agile

View on Board

HipChat discussions

Dedicated room: [Create a room](#) [Choose a room](#)

Other rooms: [Issue mentioned in 2 rooms](#)

git Process



- All code is open and on Github.
- All work is on a ticket branch.
- Non-merge commits on master only for critical fixes or trivial documentation patches (spelling mistakes).
- Branches are always rebased before merging.
 - CI can test the state of the system that will exist after the merge to master.
 - History is much easier to read and git bisect is more straightforward.
- Commits can be split, squashed or re-ordered before merging.
- Commits should be a self-contained unit of functionality or cleanup.
 - whitespace changes always separate commits.

Clean history



| Subject | Author | Date |
|---|---------------|-------------------------|
| tickets/DM-6168 origin/tickets/DM-6168 w.2016.32 Merge branch 'tickets/DM-1972' | Tim Jenness | August 2, 2016 at 16:08 |
| tickets/DM-1972 origin/tickets/DM-1972 lsst.afw.table.PIXEL is CoordinateType_PIXEL with new SWIG | Tim Jenness | August 1, 2016 at 19:07 |
| Merge branch 'tickets/DM-2280' | Russell Owen | July 19, 2016 at 10:42 |
| tickets/DM-2280 origin/tickets/DM-2280 Remove pixelScale arg | Russell Owen | July 19, 2016 at 08:36 |
| Modernize testMakePixelToTanPixel | Russell Owen | July 11, 2016 at 10:33 |
| Run autopep8 on pix to tan pix code | Russell Owen | July 11, 2016 at 10:31 |
| Correct the definition of TAN_PIXELS | Russell Owen | July 9, 2016 at 09:27 |
| w.2016.28 Merge branch 'tickets/DM-6641' | Jim Bosch | June 24, 2016 at 07:54 |
| tickets/DM-6641 origin/tickets/DM-6641 Add exact versions of image test utilities. | Jim Bosch | June 23, 2016 at 06:16 |
| Don't use PyFITS internal modules. | Jim Bosch | June 23, 2016 at 06:14 |
| Merge branch 'tickets/DM-6566' | Russell Owen | June 22, 2016 at 13:54 |
| tickets/DM-6566 origin/tickets/DM-6566 Add updateRefCentroids and updateSourceCoords | Russell Owen | June 21, 2016 at 17:15 |
| Merge branch 'tickets/DM-5293' | John Swinbank | June 13, 2016 at 05:07 |
| tickets/DM-5293 origin/tickets/DM-5293 Fix test to cover both Wcs and TanWcs. | John Swinbank | June 12, 2016 at 18:33 |
| Merge pull request #78 from lsst/tickets/DM-5822 | Pim Schellart | June 12, 2016 at 08:07 |
| tickets/DM-5822 origin/tickets/DM-5822 Fix for optimization dependent fail of convolve.py | Pim Schellart | June 10, 2016 at 16:30 |
| Merge branch 'tickets/DM-5462' | Russell Owen | June 7, 2016 at 11:19 |
| tickets/DM-5462 origin/tickets/DM-5462 Minor cleanups to image py files | Russell Owen | June 7, 2016 at 11:18 |
| Add makeRampImage | Russell Owen | June 2, 2016 at 10:44 |
| Add lsst.afw.geom.testUtils.BoxGrid | Russell Owen | June 2, 2016 at 10:10 |
| Add NullLinearityType to cameraGeom | Russell Owen | May 26, 2016 at 09:02 |
| Merge pull request #77 from lsst/tickets/DM-6090 | Pim Schellart | June 2, 2016 at 04:42 |
| tickets/DM-6090 origin/tickets/DM-6090 Replace boost::lexical_cast with std equivalent | Pim Schellart | May 24, 2016 at 12:21 |
| Merge pull request #76 from lsst/tickets/DM-6089 | Pim Schellart | May 31, 2016 at 13:47 |
| tickets/DM-6089 origin/tickets/DM-6089 Replace boost fixed width integer types with std equivalents | Pim Schellart | May 13, 2016 at 14:19 |
| Merge pull request #75 from lsst/tickets/DM-6325 | Pim Schellart | May 31, 2016 at 11:29 |
| tickets/DM-6325 origin/tickets/DM-6325 Replace BOOST_STATIC_ASSERT with static_assert | Pim Schellart | May 27, 2016 at 15:52 |
| Merge pull request #74 from lsst/tickets/DM-6094 | Pim Schellart | May 31, 2016 at 08:19 |
| tickets/DM-6094 origin/tickets/DM-6094 Use type_traits from std instead of boost | Pim Schellart | May 27, 2016 at 12:43 |
| Merge pull request #70 from lsst/tickets/DM-6092 | Pim Schellart | May 28, 2016 at 15:45 |
| tickets/DM-6092 origin/tickets/DM-6092 Remove use of boost::noncopyable | Pim Schellart | May 25, 2016 at 15:47 |
| Merge pull request #73 from lsst/tickets/DM-6095 | Pim Schellart | May 28, 2016 at 14:03 |
| tickets/DM-6095 origin/tickets/DM-6095 Use forwarding references instead of boost::ref | Pim Schellart | May 26, 2016 at 10:05 |
| Merge branch 'tickets/DM-6166' | Russell Owen | May 27, 2016 at 19:26 |



- Unit testing is done using the Python `unittest` package.
- Migrating to the `pytest` test runner
- Key advantages of `pytest`:
 - JUnit XML output that can be ingested by Jenkins.
 - Ability to run all tests at once (this makes global variables obvious).
 - Can filter such that only matching tests are executed.
 - Can easily run coverage analysis of codebase.
- Developing large scale integration tests for the science pipelines using real data.
- Track performance metrics of science pipeline over time.

Package and version management



- The DM software consists of almost 100 separate packages with complex dependencies.
- Updating one package triggers rebuilds of all packages that depend upon it.
- A single build should be reproducible based on the state of each package used for the build.
- When debugging we would like to be able to switch in libraries from earlier builds.
- We use the EUPS tool to manage the dependencies and allow version management.
 - EUPS uses environment variables to adjust search paths.
 - Each build of a package is installed into a distinct directory tree.
 - <https://github.com/RobertLuptonTheGood/eups>

Continuous Integration



- Use Jenkins for CI
- master should always be releasable.
- Regular jobs ensure that master is always buildable and passes unit tests and integration tests.
- Matrix includes CentOS 6 and 7 and OS X (2 versions), along with Python 2 and Python 3.
- Developers can submit jobs at any time, specifying their ticket branches.
 - Multiple packages can have the same ticket branches.
 - Multiple branches can be specified.
- Some standalone packages (those without EUPS dependencies) backed by Travis triggered directly by Github pull requests.

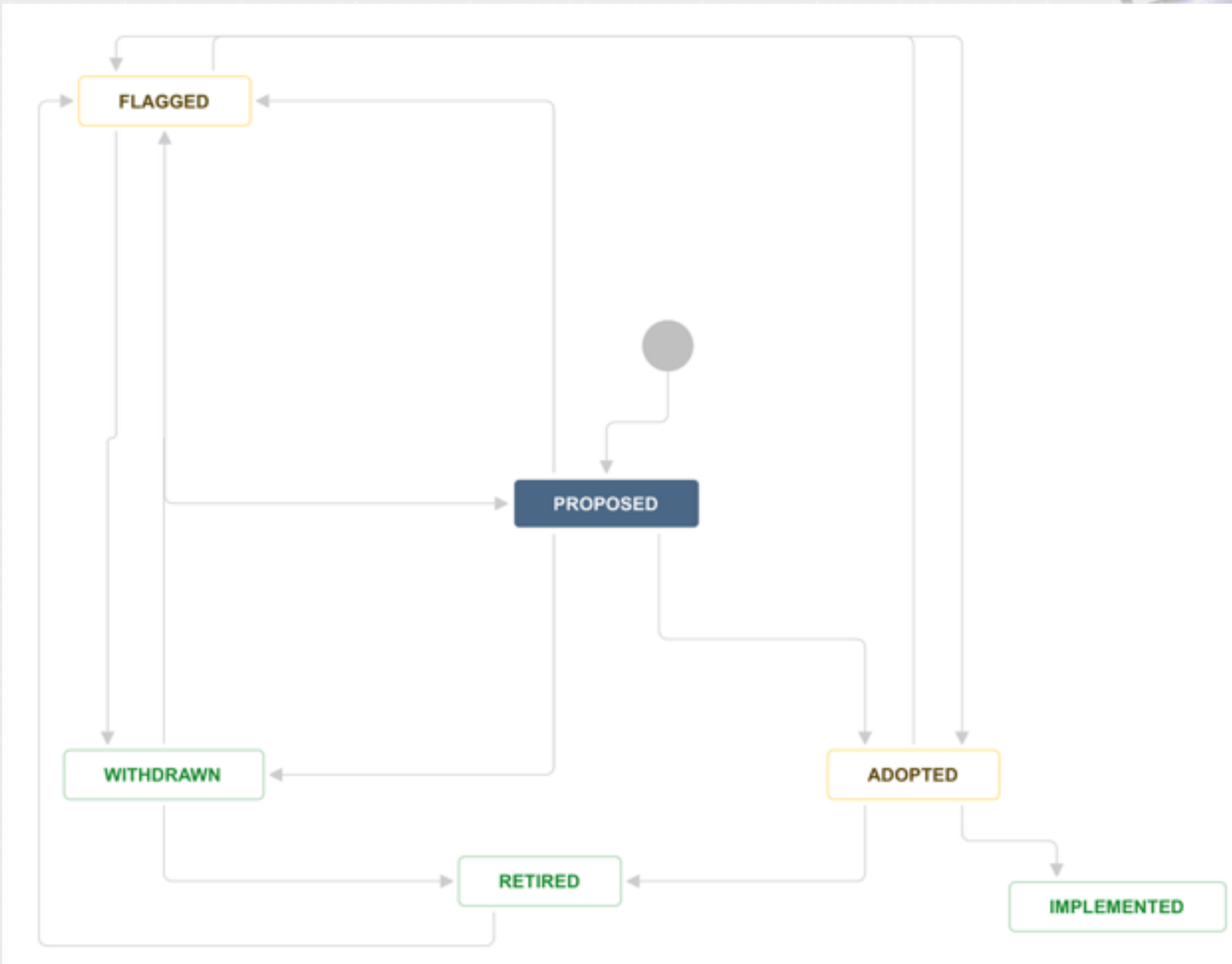


You are empowered to make decisions on any DM-internal matter—such as technical/algorithm issues, process improvements, and tool choices—when:

- 1. you are willing and able to do the work to implement the decision yourself or with people who agree with you,*
- 2. you (collectively) are willing and able to fix any problems if it goes wrong, and*
- 3. you believe that all affected parties (including your immediate manager) would not seriously object to your decision and implementation.*

We use an RFC (Request for Comments) process to publicize and discuss changes. The purpose of an RFC is to inform others about the existence and content of the proposed decision and implementation in order to allow them to evaluate its impact, comment on it, refine it if necessary, and agree (implicitly or explicitly) or object (explicitly) to its execution.

RFCs are a JIRA workflow



Coding Standards



- LSST DM Developer guide: <https://developer.lsst.io>
- Python coding standard: https://developer.lsst.io/coding/python_style_guide.html
 - Written in terms of PEP8.
- C++ coding standard: https://developer.lsst.io/coding/cpp_style_guide.html