

# ARC Release Management in Action

Jon Kerr Nilsen  
ARC Release Manager

# Outline

- A bit on ARC releases and version numbers
- [https://wiki.nordugrid.org/wiki/Release\\_management#Release\\_categories](https://wiki.nordugrid.org/wiki/Release_management#Release_categories)
- Problems, progress, protips

# ARC releases

- A NorduGrid ARC release consists of a set of source packages
- A certain source package is identified by its version number, <major>.<minor>.<bugfix>
- E.g., NorduGrid ARC 4.1.0 or NorduGrid ARC Documents 1.3.4

# Release types

- major release
  - longer term planning
  - 3-6 months preparation
  - alpha, beta, rcx test releases
  - introduces new components, features
  - obsoletes components
  - may break backward compatibility
  - scheduled release
  - this release bumps the major number in the version number of the package causing the major release

# Release types

- minor release
  - planned bugfix release
  - max one month preparation
  - no new components
  - only minor new features and/or significant bugfixes
  - scheduled release
  - this release bumps the minor number in the version number of the package causing the minor release

# Release types

- bugfix release
  - planned bugfix release
  - max one month preparation
  - no new features, no new components
  - only a limited number of less significant fixes
  - scheduled release
  - this release bumps the bugfix number in the version number of the package causing the bugfix release
- emergency release (rarely used)
  - unplanned urgent release to fix a security issue or a critical bug
  - effectively the same as bugfix release, but may or may not require bump of bugfix number

# Release workflow

1. Announce release plan
2. If needed, create new branch in svn
3. Start merging commits from trunk to release branch
4. Start nightly builds on branch
5. Monitor blocking bugs and commits to trunk
6. When nothing is blocking, make release tag
7. Build binaries
8. If release candidate, ask test team and volunteers to test the candidate
9. If all works and most important bugs are fixed, tag final tag, announce it
10. Push to EPEL testing, Debian unstable
11. Wait for people screaming two weeks later because release automatically went to EPEL stable

# Since last year

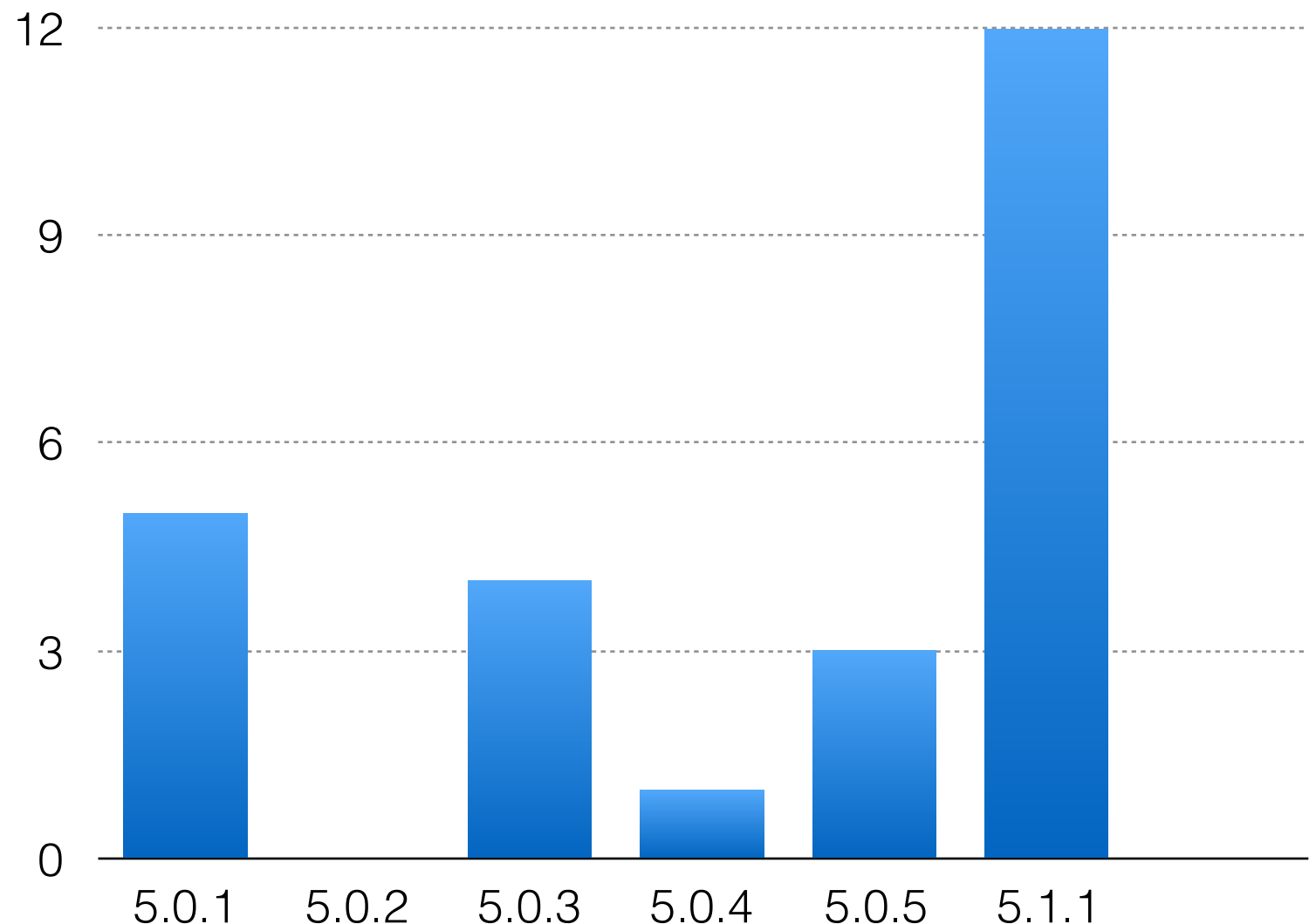
- We're still at release 15.03
- Update 7 was released on Thursday last week
  - (that would be ARC 5.1.1)
- 6 updates involving ARC
  - 5 bugfix releases, one minor release



# ARC5 releases in numbers

- Time from release decision to release date varies a bit
- from <1 day (bugfix release to fix bug from previous bugfix release)
- to 3 months (minor release from 5.0.5 to 5.1.1)
  - Minor release will always take more time than bugfix release since it involves new features
  - Having release manager on parental leave probably doesn't speed things up
- Release 5.1.1 was tagged even before 5.1.0 was released (thanks, Andrej :)

No. of core meetings per ARC release (KPI?)



# Problems

- New releases can drag out for months
  - most bug fixing and finding starts when new release preparation phase is announced
  - usually minor releases are triggered by decisions to introduce new features – these are not always implemented at decision time
- testing takes time
  - forgetting testing speeds up release procedure a lot
- Release procedure and bug fixing depends too much on each other
  - bugs found after release decision often blocks release!

# Improved release procedure

(slide from back in the good old days of 2014)

- One major release per year
  - containing all major new features and backwards-incompatible changes
  - containing NO bugfixes
- Rapid bugfix releases - every week where there has been one or more bugs fixed
  - Bugfix releases are backwards compatible - skipping a few bugfixes or updating for every new release should not be problematic
  - Requires very easy and clear system updates
  - Makes it much easier to test (and if bug is hard to test, it should say so in release notes)
- Every commit to repo should be reviewed by someone before allowed to port to branch
  - already done (sort of) by release manager for bugfix releases, but not for major releases
  - what a commit comment should contain should be written down and followed (why, reference to bug, new feature -> link to jira, how to test...)

# Improved release procedure

(slide from back in the good old days of 2014)

- One major release per year **More like every 1.5 year**
  - containing all major new features and backwards-incompatible changes
  - containing NO bugfixes **Uhm, that won't work**
- Rapid bugfix releases - every week where there has been one or more bugs fixed
  - Bugfix releases are backwards compatible - skipping a few bugfixes or updating for every new release should not be a problem
  - Release notes and clear system updates
  - Makes it much easier to test (and if bug is hard to test, it should say so in release notes)
- Every commit to repo should be reviewed by someone before allowed to port to branch
  - already done (sort of) by release manager for releases **Still done by release manager**
  - what a commit comment should contain should be written down and followed (why, reference to bug, new feature)  
**Almost all commit comments are understandable, almost always clear if it's not a simple bugfix**

# Final solution


The dream

- Press a button on a web page, wait for binaries
- When binaries ready, press "automatic testing" button
- If release candidate, announce ready for early testers
- If final release, press "publish to repos" button
- Any failures should be mailed to release team mailing list

# Release Manager Wanted

- Current Release Manager got hired as group leader for the HPC group at UiO
- Means there will be a very nice position open as ARC Release Manager and developer for NeIC/NT1
- If you
  - have experience in ARC development (or similar)
  - are not too scared of release management from this talk
  - want to move to a very nice multi-national working environment in beautiful surroundings in Oslo, Norway
  - **then please contact me after this talk :)**



A dolphin is captured mid-leap, its body arched gracefully above the ocean's surface. The dolphin's skin is a dark, sleek grey, contrasting with the vibrant orange and yellow hues of the sunset. Water droplets are frozen in mid-air around the dolphin's tail, creating a dynamic sense of movement. The background features a bright sun low on the horizon, casting a long, shimmering reflection across the water. The sky is filled with soft, wispy clouds, and the overall atmosphere is one of tranquility and natural beauty.

So Long, and Thanks  
for All the Fish!