



# Testing Complex Scientific Software: Key Challenges and Effective Strategies

Sandra Castro (ESO)

Akeem Wells (NRAO)

and the [CASA](#) Team





# Outline

- What is CASA?
- CASA Early Challenges
- Effective Strategies for Testing
- CASA New Challenges
- Lessons Learned



CASA is the official data processing software for ALMA, VLA, Nobeyama 45m and frequently used for other radio telescopes (EVN, GMRT, etc.).

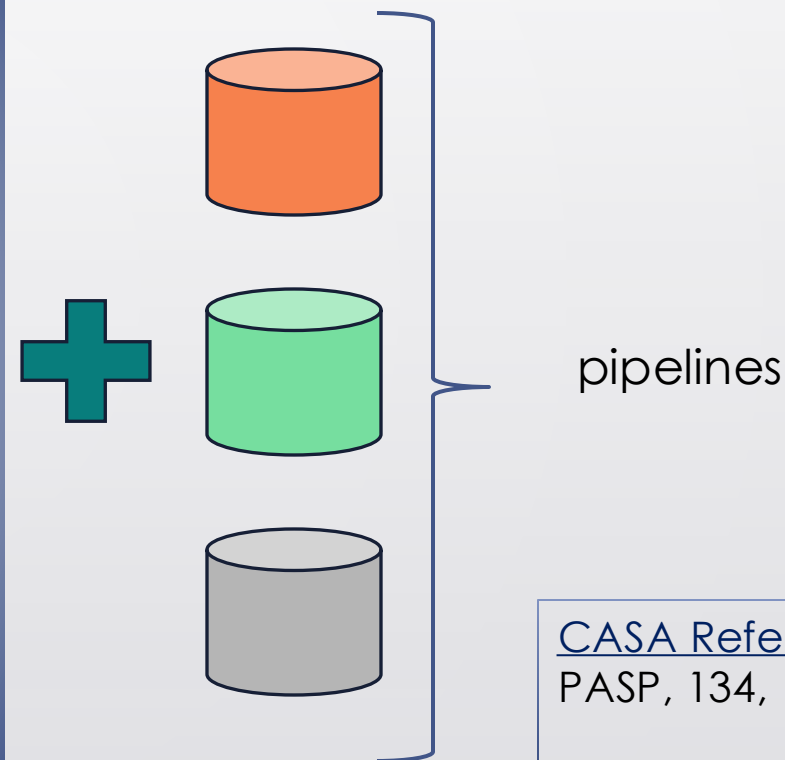
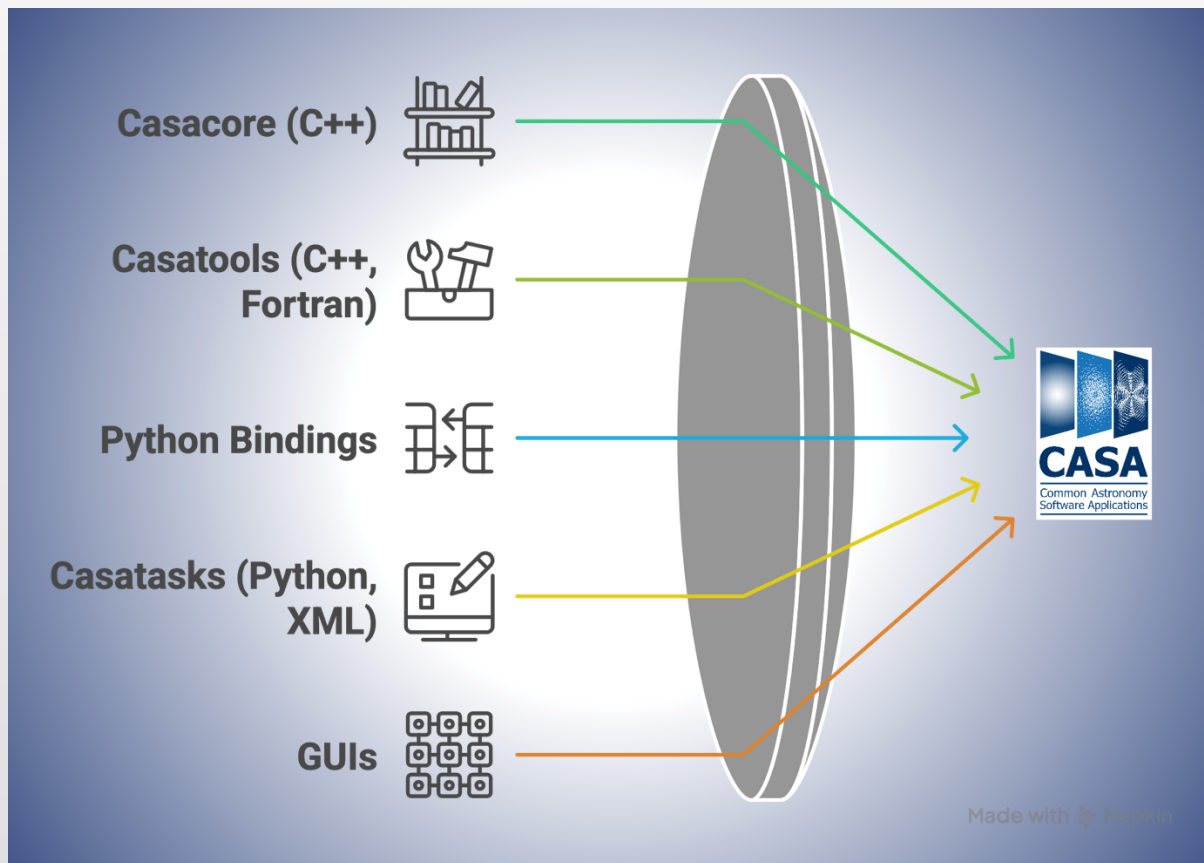


For more detailed information on CASA, see the talk by **Bjorn Emonts** today at 14:00

*CASA is being developed by an international consortium of scientists and software engineers based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for VLBI European Research Infrastructure Consortium (JIVE-ERIC), under the guidance of NRAO.*

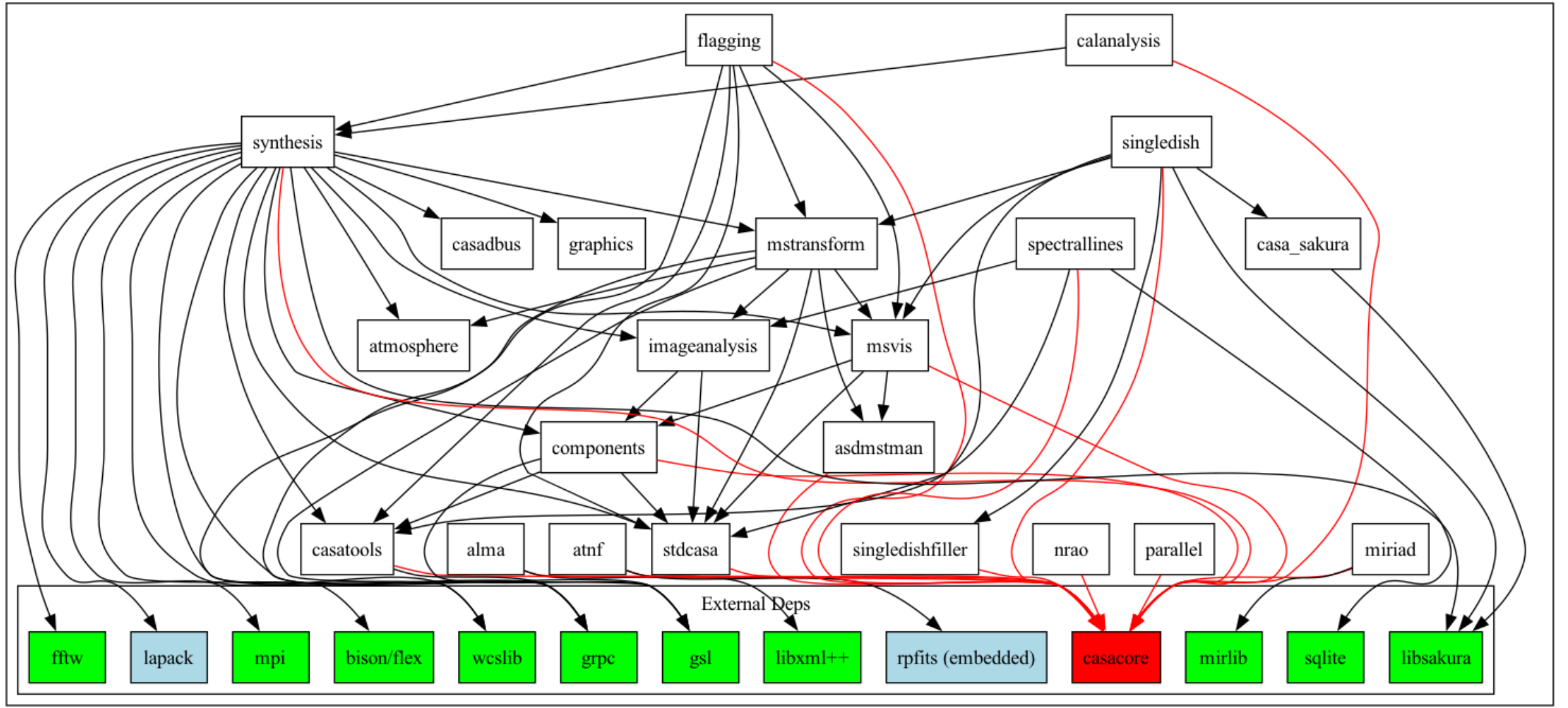


# CASA and Pipeline Packages



[CASA Reference](#)  
PASP, 134, 114501

DOI: 10.1088/1538-3873/ac9642





# CASA Early Challenges



# 1 – Heavy Annual Release



- ALMA Pipeline Cycle expected an annual release of CASA, used for the operations at the ALMA Observatory.
- Each annual CASA release contained mixed stakeholder features, infrastructure changes, maintenance bugfixes, occasional dependency updates.
- Difficult to debug if issues occurred after the release.
- Difficult to revert version to previous year.
- **STRESS!!!**



## 2 – Lack of Trust from Stakeholders



- Validation tests by stakeholders would find trivial bugs in the code
- Unclear information on what code was tested by CASA
- Fear of any changes introduced to the code
- Doubts about the quality of the CASA code



## 3 – Unclear Test Coverage of Code

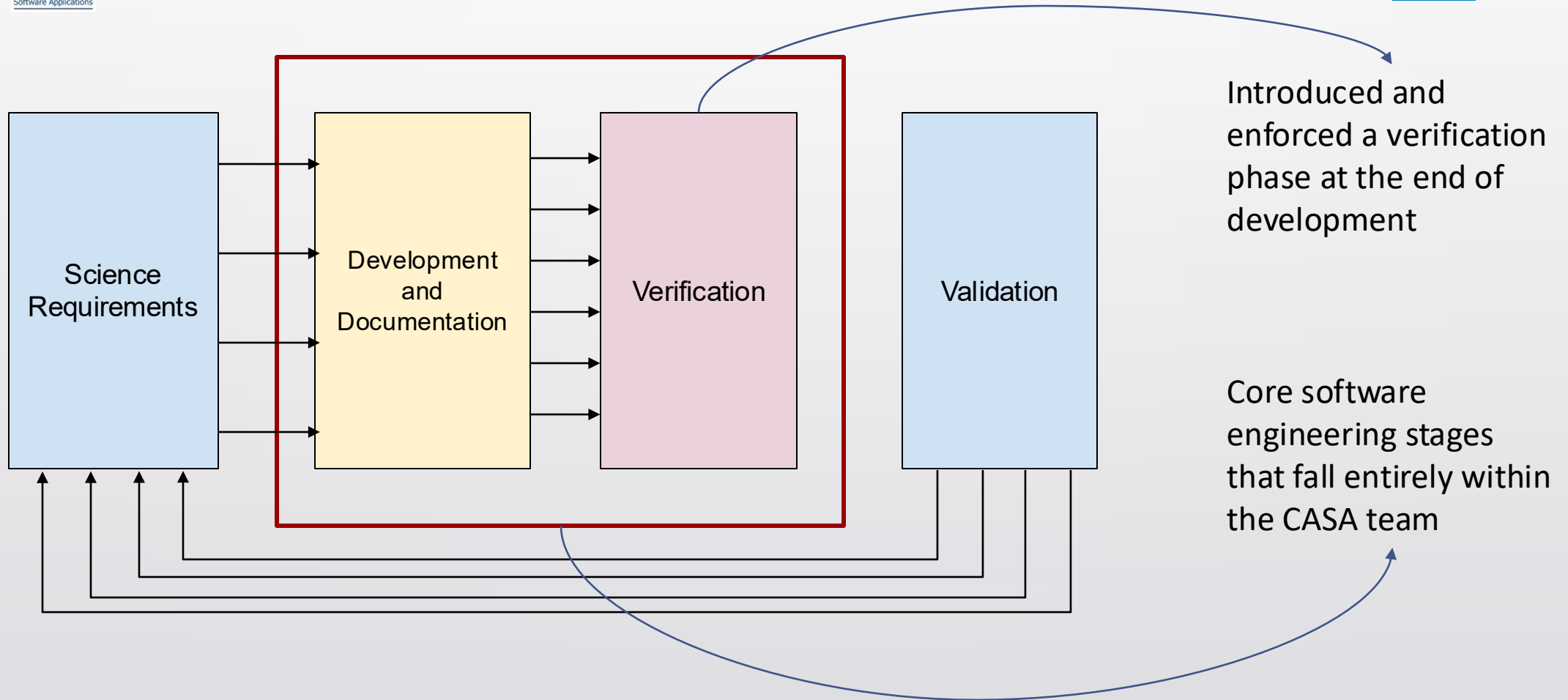


- Automated tests existed but...
- Lack of collective information about automated tests in CI
  - what is being tested?
  - when is it tested?
  - how is it tested?
- Main operational mode lacked test coverage
- PR tests needed review



Effective strategies for  
testing that worked for  
CASA

# 1- Redefined the Development Process

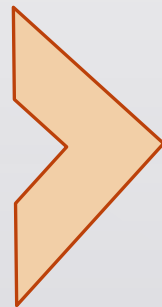


## 2- Added Verification Phase

### Infrastructure Development

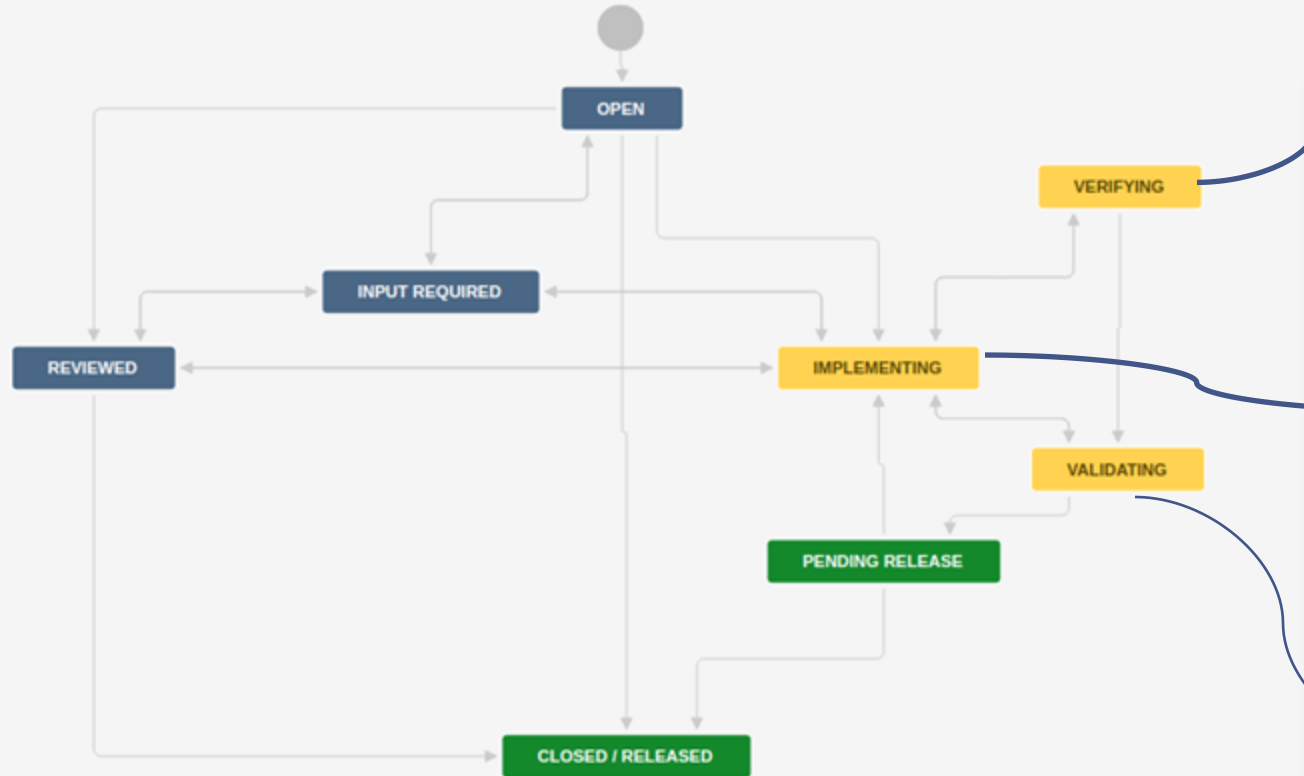
- Added Status=Verifying in JIRA
- Provided test utility package for development
- Optimized automated tests for branch development (JIRA mapping)
- Developed CI configuration for JIRA

**Improved  
communication**



- Weekly group meetings
- Strong CASA Leads relation
- Presentations to team and stakeholders

### 3- Boosted CI Tests in JIRA



automatically run all tasks and tools tests for full verification. On demand, special tests.

map JIRA components to tests, executed automatically for each push.

on-demand manual tests by validators/stakeholders



# 3- Dev Packages available in JIRA



- Avoid the “it works for me” from development team
- Bamboo automatically creates packages of dev branches and publishes to JIRA.
- Single source of truth in Bamboo.
- Packages are also used by stakeholders during validation testing.

The screenshot shows a JIRA issue page with the following sections:

- Issue Links:** A dropdown menu is open, showing "is blocked by" (CAS-13853 Add new functionality to casatestutils for creating png from CASA images, PENDING REL...) and "relates to" (CAS-14680 Modify ALMA stakeholder tests to use casatestutils' mom8\_creator, VERIFYING).
- Bamboo Branches:** A list of branches with status icons: CASASPECS > CI Build > CAS-14681 (green check), CASASPECS > Dev Build Casalith > CAS-14681 (green check), CASASPECS > Dev Package Manager > CAS-14681 (green check), CASASPECS > Dev Test 3.12 > CAS-14681 (green check), CASASPECS > Dev Verification Casalith Py3.12 > CAS-14681 (red exclamation mark), and CASASPECS > Upload Dev packages to GDrive > CAS-14681 (grey circle).
- Activity:** A tabbed interface with "Comments" selected. It shows two comments from "CASA adm account" added 2 days ago. The first comment includes a link to a BAMBOO artifact: <https://open-bamboo.nrao.edu/browse/CASASPECS-DEVTESTPY31244-1/artifact/shared/MACOS15-DMG/casa-6.7.3.2a14681.dev1-15.0-arm64-py312.dmg>. The second comment includes a link to a RHEL artifact: <https://open-bamboo.nrao.edu/browse/CASASPECS-DEVTESTPY31244-1/artifact/shared/ML228-Python-3.12-Tar-distribution/casa-CAS-14681-1-py3.12.el8.tar.xz>.



## 4- Increased Test Coverage



Increased coverage of public functionality (**tasks** and tools)

Added more test cases to main tasks (prioritized test development)

Added serial and MPI tests for parallelized tasks (mainly imaging task → tclean)

Added benchmark performance tests for main tasks ([Airspeed Velocity](#) – **ASV**)

Added stakeholder tests

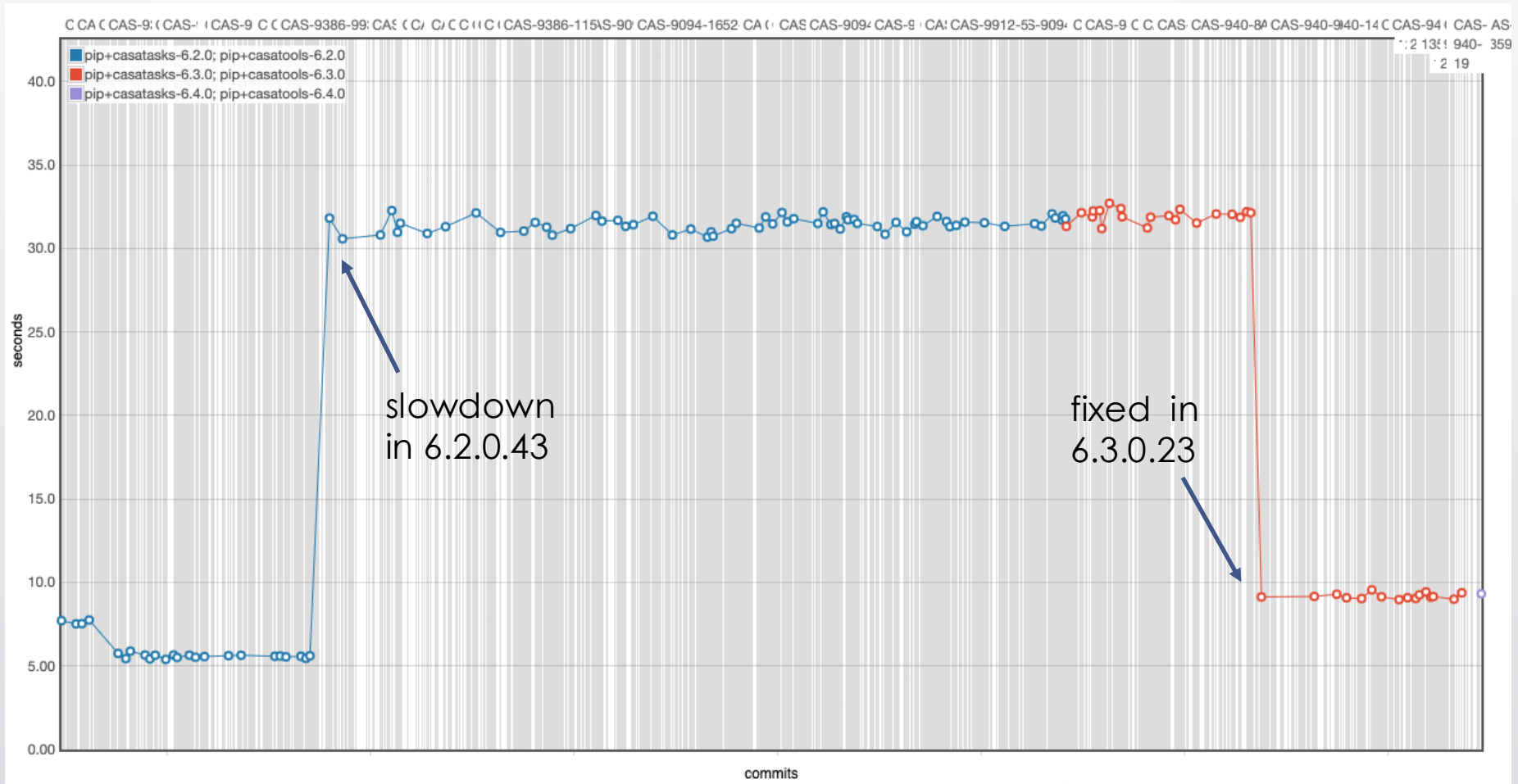


# 4- Increased Test Coverage – ASV Benchmark Tests



In 6.2 cycle, pipeline reported that flagdata was 4x slower than in 6.1.1

Performance tests showed issue ... and fix afterwards.



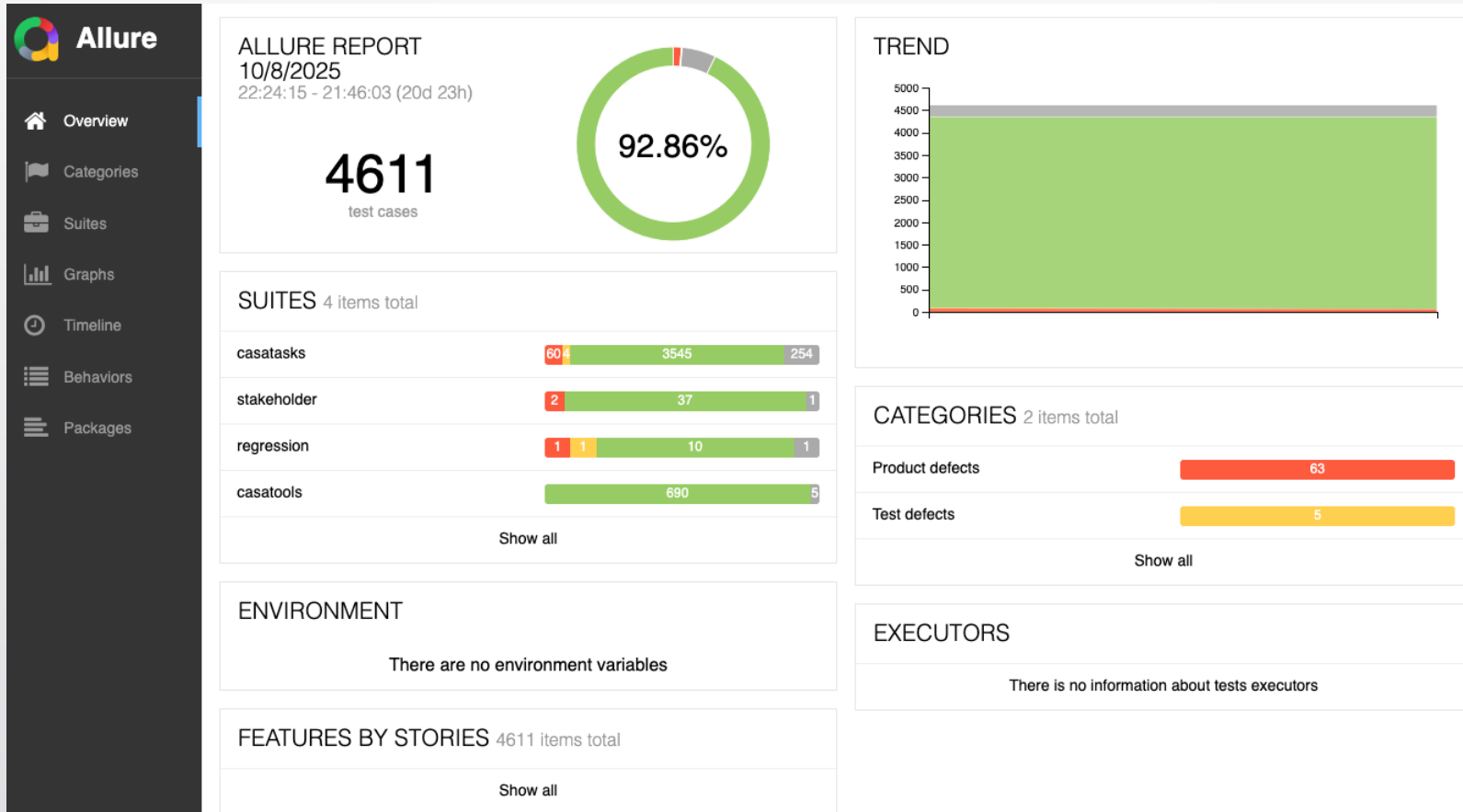


## 5- Improved Trust from Stakeholders



- Created the Stakeholder Tests
- The most important use-cases defined by the stakeholder who provided data and scripts.
- Verification team: discussed requirements, references for comparisons, sub-selection of data to lower runtime.
- Tests were validated by stakeholders, added to CI, used for feature and main branch verification.
- Side-effects of code changes for one stakeholder can be verified for other stakeholders. Common code is particularly prone to this.

# CASA Tests Allure Reports



ALLURE REPORT  
10/8/2025  
22:24:15 - 21:46:03 (20d 23h)

4611  
test cases



SUITES 4 items total



Show all

ENVIRONMENT

There are no environment variables

TREND



CATEGORIES 2 items total



Show all

EXECUTORS

There is no information about tests executors



## 6- More frequent releases



- CASA moved to quarterly releases.
- A few CASA releases for general users during the year and CASA+Pipeline annually for pipelines (e.g. ALMA operations).
- Avoided flood of features and bugfixes on the same single annual release
- Pre-release packages available for download anytime.
- Reduced “release stress”. 😊



# CASA New Challenges



# Too many tests, too long to run



## Automated Tests Run Time in Bamboo: main branch

~4600+ test cases executed multiple times for each build

- concurrent plans with concurrent jobs on Dockers if possible
- limited available host runners for stakeholder tests
- Matrix of multiple OSs and multiple Python versions
- Most optimized for single Python:
  - $\text{Max}(\text{job}(\text{Plan1}, \text{Plan2}, \text{PlanN})) + \text{Max}(\text{job}(\text{Stk1}, \text{Stk2}, \text{StkN})) \sim 32 \text{ hours}$



# Task tclean: a continuous challenge for testing



## tclean

```
tclean(vis, selectdata=True, field="", spw="", timerange="", uvrange="", antenna="", scan="", observation="", intent="", datacolumn='corrected', imagename="", imsize=[100], cell=""1arcsec", phasecenter="", stokes='I', projection='SIN', startmodel="", specmode='mfs', reffreq="", nchan=-1, start="", width="", outframe='LSRK', veltype='radio', restfreq="", interpolation='linear', perchanweightdensity=True, gridded='standard', facets=1, psfphasecenter="", wprojplanes=1, vptable="", mosweight=True, aterm=True, psterm=False, wbawp=True, conjbeams=False, cfcache="", usepointing=False, computepastep=360.0, rotatepastep=360.0, pointingoffsetsigdev="", pblimit=0.2, normtype='flatnoise', deconvolver='hogbom', scales="", nterms=2, smallscalebias=0.0, fusedthreshold=0.0, largestscale=-1, restoration=True, restoringbeam="", pbcor=False, outlierfile="", weighting='natural', robust=0.5, noise='1.0Jy', npixels=0, uvtaper=[]], niter=0, gain=0.1, threshold=0.0, nsigma=0.0, cycleniter=-1, cyclefactor=1.0, minpsffraction=0.05, maxpsffraction=0.8, interactive=False, nmajor=-1, fullsummary=False, usemask='user', mask="", pbmask=0.0, sidelobethreshold=3.0, noisethreshold=5.0, lownoisethreshold=1.5, negativethreshold=0.0, smoothfactor=1.0, minbeamfrac=0.3, cutthreshold=0.01, growiterations=75, dogrowprune=True, minpercentchange=-1.0, verbose=False, fastnoise=True, restart=True, savemodel='none', calcres=True, calcpsf=True, psfcutoff=0.35, parallel=False) \[source\]
```

- 94 parameters
- needs **use-case** testing instead
- stakeholder tests appropriate for this task



# Build System and Coverage Reports



- In the current state, the coverage reports are not reliable
- CASA's cmake build system includes files that are not used, and those lines are considered when creating the reports
- Need to refactor the build system to get more accurate coverage numbers

## for the Flagging module

Coverage including all files: Lines 36%, Functions: 29%

Coverage including only used files: Lines 66%, Functions: 70%



# Lessons Learned



## Random Tips...



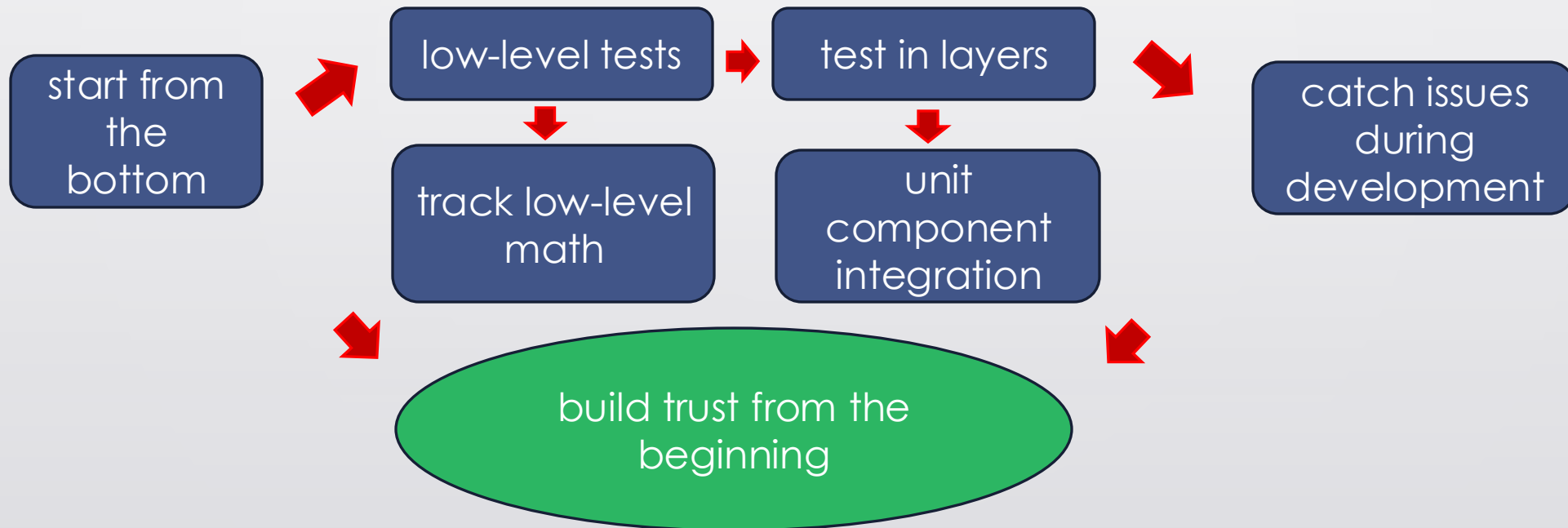
- Not all tests are useful!
- Check redundancy of tests; review tests; cleanup tests
- Use coverage reports to guide test development (to add **and remove**)
- Provide template for test development; it helps maintenance of tests
- Keep runtime down\*
- Multiplicity of OSs and other dependencies only when/where needed
- **laC** to help adding/migrating test plans

# Starting from Scratch – RADPS Test Strategy



## Radio Astronomy Data Processing System (RADPS)

development led by NRAO, in collaboration with ESO and NAOJ.





Thank you!

[scastro@eso.org](mailto:scastro@eso.org)

[CASA readthedocs](#)

CASA ASV Benchmark Tests – [casabench](#)