

Lessons from ALMA Software Development

Jeff Kern
NRAO/ALMA

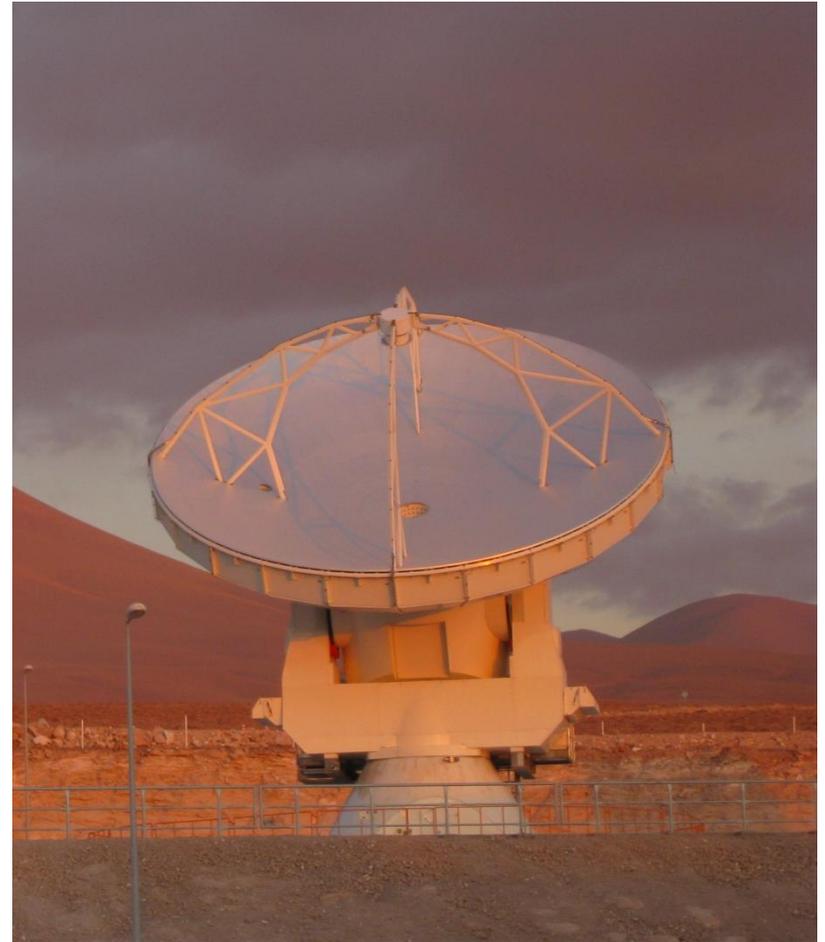
Why is Online Telescope Software Not a Solved Problem?

- ▶ Scaling and Complexity
 - SKA is not just a bigger version of existing systems
- ▶ Higher Expectations
- ▶ End to End Systems
- ▶ Archive Services
- ▶ VO Applications



ALMA

- ▶ Largest Ground Based Radio Astronomy Project
- ▶ International Collaboration
 - NAOJ / ESO / NRAO
- ▶ Remote Deployment
- ▶ Distributed Development



Configuration Management

“Oh the code works fine, that’s just a configuration issue.”

- ▶ In ALMA I would estimate at least a third of our issues have been just a configuration issue.
- ▶ Configuration is a decentralized problem
 - Depends on both development and deployment
- ▶ Time Dependent
 - Not constrained to a “Release Cycle”
- ▶ Crosses many project boundaries
 - Software, Hardware, Operations

What's so hard... it's just data.

- ▶ Computational tasks are usually easy.
 - Interesting, lots of research on these types of problems.
 - Centralized, one developer can usually solve the problem.
 - Scaling can be a problem, but can design for it.
- ▶ Information flow causes problems.
 - Coordinating the information to be stored and agreeing on format.
 - Synchronization across a large distributed system.

Scope of the Issue

- ▶ In ALMA Configuration Depends on:
 - Hardware Version
 - Software Version
 - Firmware Version
 - Deployment Scenario
 - Calibration Data
- ▶ In ALMA Configuration is needed by:
 - Hardware Engineer
 - Scheduling System
 - Software System
 - Observing Preparation Tool
 - Maintenance System
 - Operations Staff

Mitigation

- ▶ This is a project task, not just a computing team issue .
- ▶ Comprehensive Requirements Capture.
 - Strong, well staffed System Engineering team.
 - Design the configuration management system with all stakeholders from start of project.
 - Do a few key use cases in detail.
- ▶ Interface Control Documents
 - Need responsive change control system to be effective.
 - Should include “Theory of Operations”

Distributed Development

▶ ALMA's Approach

- Concentrations of people in the larger teams with a few people at other locations to cross-pollinate.
- Smaller teams are fairly isolated.
- Some people split amongst teams.

▶ Advantages

- Intra-team communication good.
- Collocated hardware and software expertise.

▶ Disadvantages

- Us vs. Them mentality very easy to fall into.
- Integration and release is very difficult.
- Split effort is less effective.

Mitigation: Matrix Management

- ▶ The advantages of collocating a team are too great to ignore.
- ▶ Break down the “Us vs. Them” mentality by forming cross team groups to deliver specific functionality.
 - In ALMA jargon: **Function Based Teams**.
 - 3–6 Month spans (ALMA targets 2 months)
 - At least one, and sometimes two face to face meetings.
- ▶ Could be expanded beyond the Computing team (i.e. a team to deliver a hardware component).

Mitigation: Integration Team

- ▶ Integration is the careful assembly of the software components into a working whole.
 - More to this than a simple build and test process.
- ▶ Integration team should know:
 - What the software is supposed to do.
 - How it is supposed to work.
 - The use cases of the clients in detail.
 - How to determine why it's not working when it fails.

I advocate transitioning the architecture team to an integration role as the system matures.

Commissioning Comes Before Operations

- ▶ Cannot commission a system from Scheduling Blocks.
 - Low level hardware control
 - Very different exception handling
 - Different use cases.
- ▶ Most Difficult Problems Come First
 - Fastest data acquisition
 - Debugging hardware and software simultaneously, need lots of monitoring and logging.
 - Edge cases of the system, missing hardware *etc.*

Mitigation: **Invest in time travel?**

Summary

- ▶ Configuration Management
 - Project level issue
 - Include this from the beginning
 - Careful requirements capture with all stakeholders.
- ▶ Distributed Development
 - Integration Team
 - Matrix Management
- ▶ Commissioning Comes Before Operations
 - Plan accordingly

