ALMA Release Management: A Practical Approach

Ruben Soto[a], Tzu-Chiang Shen[a], Norman Sáez[a], Jorge Ibsen[b]
[a]Atacama Large Millimeter/submillimeter Array, Av. Alonso de Córdova 3107, Santiago, Chile, [b]European Southern Observatory, Av. Alonso de Córdova 3107, Santiago, Chile.

Abstract: The ALMA software is a large collection of modules, which implements the functionality needed for the observatory day-to-day operations. The main software components are: array/antenna control/correlator, telescope calibration, submission/processing of science proposals and data archiving. The implementation of new features and improvements for every software subsystem must be in coordination with developers schedule, observatory milestones and testing resources available to verify new software. This paper describes the software delivery process adopted by ALMA since the construction phase and its evolution until these days. It also describes the acceptance process defined by the observatory in order to validate the software used for science operations. Main roles of the software delivery and acceptance processes are mentioned on this paper including their responsibility at the delivery of the different software releases. Finally, some ideas are presented about how the process should change in the near future by considering the operational reality of ALMA Observatory.

The ALMA Observatory started its commissioning phase at the Chilean site during the end of 2009. It considered some antennas installed at the high site (5000m) and the deployment of the first quadrant for the baseline correlator. Additionally, the assembly, integration and verification activities (AIV) related to the new array elements delivered by manufacture vendors [1], continued more intensely at the Operation Support Facilities (3000m). Commissioning process by using direct observing systems (control and correlator software, front-end archive, etc.) was required at the Observatory and, at the other hand, the preparation, integration and testing of pre and post observing software. In terms of software releases, a cycle of 6 months for the delivery of new features was established, which included capabilities for the observing systems and the proposal handling process as well. These cycles considered several phases before declare the software as accepted for AIV activities or Early Science observations as described at figure above. Software was commissioned by using prototypes antennas and there was time available with the operational hardware for testing purposes. However, this approach was deficient when commissioning and AIV activities started at the operational site. There were less access to the hardware for testing and more pressure for having new software capabilities working in order to continue progressing into array commissioning.

The model currently adopted differs from the previous one in terms of periodicity of the incremental releases delivered for science commissioning. We moved from 6- month period to bi-monthly schedule, which consider testing and integration as part of the cycle. This model also included the definition of different phases with formal handover between each one. Three phases were established:

- **Phase A - Developer Integration & Testing**: Developer tests
- **Phase B - Verification**: Software integration and verification
- **Phase C - Validation:** Science Validation

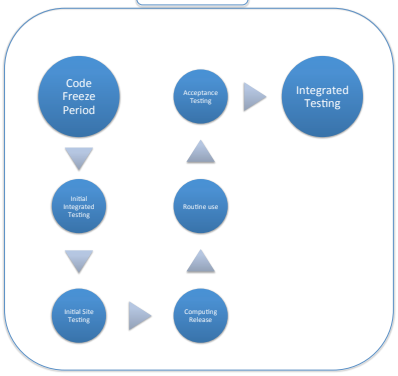
A calendar with dates for every phase was prepared and circulated to developers, computing and science testers [5]. Also independent phases were parallelized (as showed in figure above) which optimize the available resources. Formal responsibilities were defined at the computing and science teams related to the planning and delivery of the software. Thus, the ***release*** and ***acceptance*** manager roles were introduced. The steps to accepts a release are:

- **Test Report Review (TRR) :** A meeting to discuss validation/verification of previous phases
- **Acceptance Testing Period :**Final test before acceptance (software verification)
- **Acceptance Review :** Meeting to accepts/rejects the final release
- **Software Deployment in Production Environment**: Final deployment in production

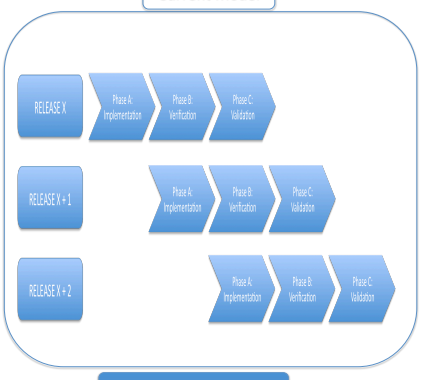
Software Change Control Board (SCCB) is a committee compound of various project stakeholders .The SCCB will meet once a week to discuss and decide on outstanding software requests.

An agile approach for the software delivery process should be adopted by the observatory. Basically, this approach is based in the **existence of a stable branch, which is patched for verification, and validation of new capabilities.** Developers should commit functionality in separate branches and verification team should patch stable branch for verification purposes. **If verification passed, Science testers should validate same functionality.** After successfully validation, **the patch can be integrated at a stable branch** and considered ready to be used for observatory's activities. This model differs of the previous one, since integration is controlled by verification team instead of developers. Stability should be also granted since less functionality is included per iteration. Features, which do not pass verification or validation phases are rejected and scheduled for another iteration. Observatory's technical times are also optimized since only features, which have passed simulation tests, are considered to be verified with operational hardware. The figure above illustrates the core of the new process where the science branch (accepted) is created after the verification and validation phases have been completed successfully and also an unsuccessful case.

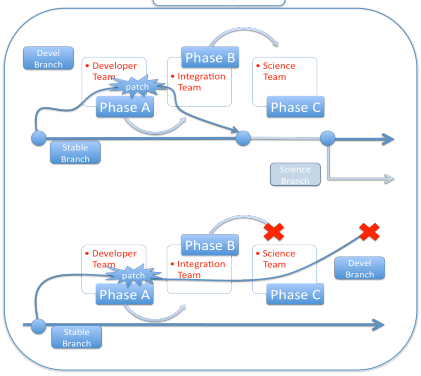
Old Model



Current Model



New Proposal



Conclusion

This paper presented the evolution of the release management process in agreement with the life cycle of ALMA Observatory. There was a transition from a traditional and static development model, suitable for early construction phases, toward a dynamical one, which considered commissioning restrictions. This new model takes into account the delivery of lite releases in terms of features but more stable as a whole. This also increased the frequency of the development cycles according to the observatory's milestones and decreased the integration/testing time required before the science commissioning phases. Formal phases were introduced as part of the process and responsible for every stage were properly identified and designated. This facilitated the process control, allowing a deterministic schedule for the entire cycle. There was also more emphasis for controlling changes over commissioned releases used for official science activities. The creation of a control board for approving/rejecting changes, evidenced the importance of maintain operational software stable as much as possible. The results showed at the end demonstrated this was the correct path since ALMA commissioning phase has been successfully performed from the software point of view. However, there is still another important milestone to be completed by the Observatory in the coming years: the full operations model that will demand a new adaptation of the software delivery process in order to fulfill the operational requirements. Thus, an agile approach was proposed that considers the robustness and stability of the system as a mandatory goal over the introduction of new capabilities. It is expected that several improvements at the system simulation and continuous integration environment must be developed as part of the implantation of the model. The experience reveals that the implementation of a new model is not a straightforward process. It will require several technical improvements but, more important and difficult, is the adaptation of human capital (developers, testers, valuators) to new paradigm.

REFERENCES
[1] B. E. Glendenning, and G. Raffi., "The ALMA computing project: initial commissioning", Proc SPIE, Vol. 7019, 701902 (2008).
[2] B. E. Glendenning , J. Ibsen , G. Kosugi , G. Raffi, "ALMA software management and deployment", Proceedings of SPIE Vol. 7740, 77401L (2010).
[3] B. Lopez, R. Araya, N. Barriga, et al., "Software regression testing: practical experience at the ALMA test facility", Proceedings of SPIE Vol. 7019, 70192X (2008).
[4] T.C. Shen, J.P.A. Ibsen, R.A. Olguin, R. Soto, "Status of ALMA Software", Proceedings of ICALEPCS 2011, Grenoble, France.
[5] V. Gonzalez, M. Mora, R. Araya, et al., "First year of ALMA site software deployment: where everything together," Proceedings of SPIE Vol. 7737, 77371Z (2010)
[6] JIRA Atlassian planning and tracking tool, https://jira.atlassian.com