

Feedback from Release Strategy Discussion

Frederick Luehring
luehring@indiana.edu
Indiana University
October 26, 2007
SW Week Friday Plenary

Introduction

- On Tuesday there was a well-attended session designed to give the developers a chance to provide feedback the strategy used to build the offline software releases. We had four presentations:
 - Release Building Strategy - Status and Plans by D. Quarrie
 - Validation, Transforms, and Feedback on Offline Software Release Process by Manuel Gallas
 - Former Release Coordinator's Comments by S. Binet
 - on Releases and Software Project - impressions from an outsider by Adele Rimoldi
- There also was much discussion from the floor and the meeting tone was constructive. Please send additional comments to Atlas.Release@cern.ch.

David Quarrie's Talk

- David's main points were:
 - Several release types production, bug fix, validation, migration, development, nightly.
 - The builds are staged - lower level projects freeze first.
 - The use of validation and migration releases improves release process.
 - Tag Collector has been improved to work with the current strategy but more work is needed especially on tag bundles.
 - We are moving to making a numbered release by freezing nightly build now that the nightly builds the kit.
 - There will be a build each night so that patched code can be distributed daily to Point1.
 - We need to revisit when to build a release because the number of patches has become excessive.



Conclusions

- λ Major changes to release building procedures made during 2007
- λ ~20 lxbuild machines now dedicated to support several platforms and branches
- λ My perspective is that things have considerably improved as a result of new procedures
 - √ Albeit with some developer confusion
 - √ and we have a shopping list of improvements to the tag collector
 - λ Many of these are being developed and should be available soon
- λ Overall numbered release build procedure being re-evaluated to make more robust
- λ My baseline strategy is to continue the present strategy and refine procedures to make them more automatic and robust

Intro to Manuel Gallas' Talk

- Make sure that people understands AtlasPoint1 (point1) and AtlasProduction (pcache) as two independent technical mechanisms to patch (with limitations) a FAULTY release.
- AtlasPoint1 can and will be (if needed) installed daily in Point1. AtlasProduction will be created as unvalidated cache daily but will not be distributed. To distribute the pcache everyday would be too much and lead to confusion.

Manuel Gallas' Talk

- Manuel's main points were:
 - A “crash” program to validate the software has been started with release 13. Manuel coordinates this program.
 - 4 tools are used ATN, FCT, KV, and RTT.
 - There is weekly monitoring of the validation and bi-weekly of GRID production using SampleA.
 - The number of tests is large but still more tests are needed for complete coverage.
 - Patching of the release is available for both GRID production (offline) and Point1.
 - For 13.0.30 the number of patches is large: ~90 patches for the offline release and ~130 patches for Point1
 - Strict tag app. since 13.0.10 has improved release quality.
 - New method proposed to deal with the database release.
 - Just starting to produce SLC4 patches/transforms.
 - We will always be testing on more than one platform.

Manuel Gallas' Talk

- Development and release validation need stable nightlies: bug-fix and val nightlies have worked fine
- The model of bug-fix & val nightlies should work now for development nightlies (dev & dev-val)
- The migration nightlies help but were disruptive especially when they were kept open for a long period.
- We still need documentation: a Twiki explaining what the nightlies are meant for and who are their potential users
- Tag approval procedure was initially confusing but it is necessary.
- Problems:
 - Disruption from re-merging migration nightlies - need to do this ~1 week prior to release build
 - Many deadlines missed
 - More validation needed (but a good start has been made).
 - Too many tags for patching the release

Manuel's Final Point

“Overall it has been a good learning exercise that we have to criticize and improve for the next release. Mistakes we had made and solutions we may found have to be written down as a part of the release procedure.”

Sebastian Binet's Talk

- Sebastian's main points were:
 - Believes that current release tools are better than he had when he was the release coordinator.
 - The large number of nightlies was hard to deal with initially but in the end it was very useful.
 - Ideas for improving release procedure:
 - Develop deliverables for each project for a release before work starts on the release.
 - Require each deliverable to provide a test (ATN or RTT) that the code for the deliverable must pass before it can be added to the release.
 - Have a one to two week freeze before delivering at release or allow for a stabilization period during building N.O.X releases.
 - Don't let tag bundles introduce minor improvements when fixing a bug.

Adele Rimoldi's Talk

- Adele's main points were:
 - It takes a long, long time for the release to converge.
 - View of Release 13:
 - The val nightlies were useful
 - Worries that new tags are not consistently being put into 13.1.0 also
 - Changes to externals delayed release
 - Worries about right timing in doing migration to SLC4/SLC5
 - The complicated tag approval process causes delay
 - The developers don't know how to use the tag collector for managing tag approval (too complicated?).
 - The goals of the software project should be:
 - An affordable, clean design
 - Reproducible results throughout release (not true)
 - Functional Releases (new code shouldn't break the release)
 - Validation of the geometry and physics
 - Maintainable code

Release 13

- **June 26: 13.0.10 was out**
“Kit patched in place several times
The kit had some bad links to AFS in its externals
External packages were delayed and even be unavailable for 13.0.20”
 - Many improvements since then, after 4 Months..
- October 12** - “jobs started running 24 h ago. 214 jobs BERT and 500 EMV runned in Toronto. 75 BERT failed. No bug report yet, reason for failures not yet known”, IH



Release 14

- Is this release the one where everything is collected and will be functioning all together?
- If not is it perhaps release 15?
 - “we can’t reconstruct with rel xx if we use simulated data with rel yy”
 - “we should use the cache xx.yy.zz for production, ready by tomorrow and hear the reports from validation groups at the next weekly meeting”
 - “...”
 - Never heard this?

Adele Rimoldi's Talk

- How the subprojects convergence could be reached at the end of software project itself?
- Rejecting patches is critical to the success of the release but we rarely reject patches.
 - Late requests are always accepted
 - Code in proposed patches is insufficiently tested
 - The patching process does allow for integration of the whole release
 - The patched code is not validated enough as a complete release
- Too many branches are open at one time.
 - The reason for the various branches is not sufficiently explained
 - The number of branches confuses users
- Many users want to use stable software and don't care if they don't have the newest, greatest software with the latest, greatest features
- The problem is: release something useful
 - The validation tests good at catch some types of problems but miss large problems that come from completely new sources

In summary

- All the ingredients for a good software project are in place, all the steps required are set
- **What is worrying me is**
 - how to converge each project at the project end (is Atlas Simulation a separate project for the experiment life?), and how to make it, even if dependencies are making a big blog
 - Release validation: a good exercise to walk through all the rtt and nightly tests, but still if afs or external problems (errors) are mixed with the “true” errors it is impossible to promptly follow the project evolution and its cure
 - No savannah bug report chain active from the sites where production is done. If reports arrive they are always incomplete, logs aren't available, true errors are normally not spotted
 - the so called physics validation: it shouldn't be a rough proof of functionality done three days after the release as it seems to be now, based on a branch or subbranch. This is a sort of release validation without any strong and clean connection to physics
 - The complete set of tests that should be done at release time **before** kit distribution
 - Agreement on the goals at page 5

Discussion from Floor

- Comments from Simon George, Thomas Kittelman Walter Lampl, David Rousseau and others:
 - People always do development when they are supposed to be patching.
 - When tag in release in a particular branch is failing, a tag to fix the problem should be accepted without approval process
 - Any domain can focus on one branch only. It is very hard to work on more than one branch.
 - The migration builds cause problems but it is OK to have val and dev builds,
 - Many crashes at Point1 are caused by truly corrupted data.
 - We stage the release by projects only for short periods.
 - Developers should do a standard test when submitting tags.

Discussion from Floor

- Need better checking of the tags submitted for approval.
- Hard to tell if last minute patching is useful or causes more disruption than problems that it solves.
- Release by the calendar (i.e. by having a target date) is not working.
- The Trigger and Reconstruction project coordinators want to be more closely included in the final decision to make a release.

Summary

- The release procedure is working and does produce releases that are usable.
 - Still manually intensive -simplifying/automating (e.g. nightly snapshot).
- The release coordinators and builders generally have tools and a procedure that works.
 - Improved tag collector support underway
- The increase in validation and testing is really helpful.
 - The RTT/FCT are so busy that the number of servers is being doubled.
- There are a number of areas to work on:
 - Some people are still confused by the large number of builds. We are putting together a Twiki page to explain this.
 - More validation and testing needs to be done.
 - Some project coordinators need to take a more active role.
 - The integration of changes in the migration builds still cause disruption when a large number of tags are put in just before the release. In the future this integration will be done at 7 days before the release.
 - There are too many late or last minute tags!!!!

Summary

- Final thoughts:
 - It is clear to me that there is an (expected) tension between delivering the SW on time and delivering working SW.
 - We must focus finding a system to produce the best possible releases in a timely manner.
 - And evaluate and justify the need for special-purpose releases.