SAS-SO CCB

Introduction

The software comprising the pipeline and what it runs is under configuration control. After baselining the code, all modifications to the running code will need to be approved by the Change Control Board. The purpose of configuration control is to understand the code content that affects processing results, and to maximize system uptime by vetting changes that affect reliability.

Quick Links

SAS-SO CCB Actions	SAS-SO CCB email	Mission CAR Request

Process

Proposed changes will be submitted to the CCB with an explanation of the need, changes and consequences. These can be recorded in blogs (aka "news item", labeled sassoccb) in any space, if desired. Otherwise the content can appear entirely in jira itself. Approval will be recorded here also. Consideration and approval may be done via email, evo meeting or phone depending on the urgency of the request. We will start off by requiring unanimous approval from the board members for the action to be approved.

Here is the recipe:

- requestor creates a Jira item in the SSC project, leaving the state as "In Preparation". If you have created a news item to track the content of the request, fill the url into the CCB news item url in the Jira issue.
- once it is ready to go, the requestor selects Submit to SAS-SO CCB from the workflow. This puts it in the hands of the CCB.
- the CCB determines the "severity" of the request (whether to send to the LAT &/or and Mission CCBs) and changes the priority on the issue.
- Then the issue is routed to the approriate next CCB or approved (or rejected)
    - if a minor issue, and approved then the LAT CCB is notified of the change
- if approved, the issue is assigned back to the requestor, who sets the issue to "Implemented", and then closes the issue or the CCB folks do.

The request should cover the need for the change; the fix; how it was tested; and how it can be backed out.

The software under control is the pipeline itself (database, pipeline code and user interfaces) and all scripts and executables run by the pipeline for Halfpipe and Level 1 processing and ASP. This includes Science Ops and SAS code.

Once we have approved the request internally, we will forward it to the Mission Office, where a 24 hour turnaround is promised.

GlastRelease package

A tagged release of GlastRelease will form the core of reconstruction code run in the pipeline. Changes to this release would be performed on a branch in cvs. Inclusion of the tagged packages will be proposed to the gatekeeper (Heather Kelly) who will prepare new releases. These are based on tested GlastRelease releases as much as possible. System tests will be run. There should be an Science Ops meeting presentation demonstrating the evidence - system tests plus jira issues. Indviduals should represent their changes at these meetings (**note: this did not work in I&T!**)

ScienceTools package

A tagged release of ScienceTools will form the core of Level 2 ASP code run in the pipeline. Changes to this release would be performed on a branch in cvs. Inclusion of the tagged packages will be proposed to the gatekeeper (Jim Chiang) who will prepare new releases. These are based on tested ScienceTools releases as much as possible.

Other elements

The Science Ops and ASP scripts and pipeline front and back ends are more procedural pieces, not affecting the results; only the operability of the system.

We are attempting a slightly different QA approach here - document all changes with jira issues, and have an expert client verify the changes work on dev servers. Requests with such qualifications can receive expedited approval.

Automated Science Processing (ASP)

The software and infrastructure that specifically support ASP consist of

- Oracle database tables. These contain the configuration information (i.e., thresholds, response functions to use, information from external catalogs) and science results that will be made available to the public and to the collaboration.
- ScienceTools build. This is a specific release version of the ScienceTools package against which the compiled parts of the ASP code are built and which provides the exectuables that are driven in the ASP scripts.
- ASP package. This is a collection of subpackages, comprising python scripts, compiled C++ code, and xml pipeline task definitions, that are the core of the ASP functionality. Everything is under CVS control, and are tracked using the same tagging conventions and organization as other elements of the ground software system (e.g., ScienceTools, GlastRelease, etc.). The specific versions of the various sub-packages are defined in the ASP checkout package, which in turn is tagged and defines a release version.
- ASP Data Viewer web application. This is the interface that presents the analysis results to the collaboration and the public.
- External software pacakges. Most external packages are available through the ScienceTools external libraries. The following are ASP-specific:
    - CMT v18r...
    - cx_Oracle
    - pylab

Testing changes:

- All changes will be tested using the DEV versions of the pipeline and database tables.

Board Members

The board is composed of Richard Dubois (chair), Eduardo do Couto e Silva , Steve Ritz, Rob Cameron, Analysis Coordinator (Nicola Omodei), C&A Lead (as needed; Luca Latronico and Philippe Bruel). email traffic for the board can be viewed here. Nicola Omodei will act as CCB chair in Richard's absence.

CCB Actions List

CCB Actions