JWST Science Data End-to-End Validation Framework

R. Diaz, A. Canipe , M. Cracraft, S. Crawford, M. Fix, M. Garcia Marin , M. Jordan, C. Kaleida, M. Kyprianou
Space Telescope Science Institute

At STScI we have successfully carried out a series of end-to-end tests of the Science & Operations Center (S&OC) campaigns to validate the functionality and changes to the ground and flight operation Subsystems. We have developed a process that includes scientific validation of calibration products via science simulations and a well- integrated testing infrastructure within the Science Instruments calibration pipeline. We describe them in this poster.

1

JWST observations consist of dithered images or spectra and in some cases, additional observations combined into a single product called "association." Multiple observations are necessary to mitigate bad detector pixels, improve quality of the image or spectra, fill gaps in the detectors for full-sky coverage, and improve image resolution. Correctly constructing these associations requires coordination among the different Subsystems and effective methods to share information. The ultimate goal is to produce the expected science products.

What do we mean by End-to-End testing?

End-to-end testing exercises the full planning, execution, calibration, and archival cycle of a JWST proposal. It validates proper integration of the individual components and information propagation through the whole S&OC subsystem. The Data Management Subsystem (DMS) takes care of constructing associations, generating science products, and archiving them. The diagram below provides the outline of the process with emphasis on DMS.

2

The process starts with **Designing observations** covering a minimum set of modes and parameters that will verify the correct propagation of the information across the subsystems and produce a fully calibrated product that is archived and retrievable by the science community. These observations are put together using the **Astronomers Proposal Tool (APT)** and submitted to mission **Program Coordinators (PCs)** for ingesting in the **Proposal Planning Subsystem (PPS).**

3

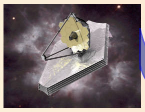
The **Observation Plan Generation Subsystem (OPGS)** schedules the observations within the allowed time and using the requested prioritized target selection.

The **Operations Script Subsystem (OSS)** Team, reviews visit plans & provides simulations for Target Acquisition observations.

4

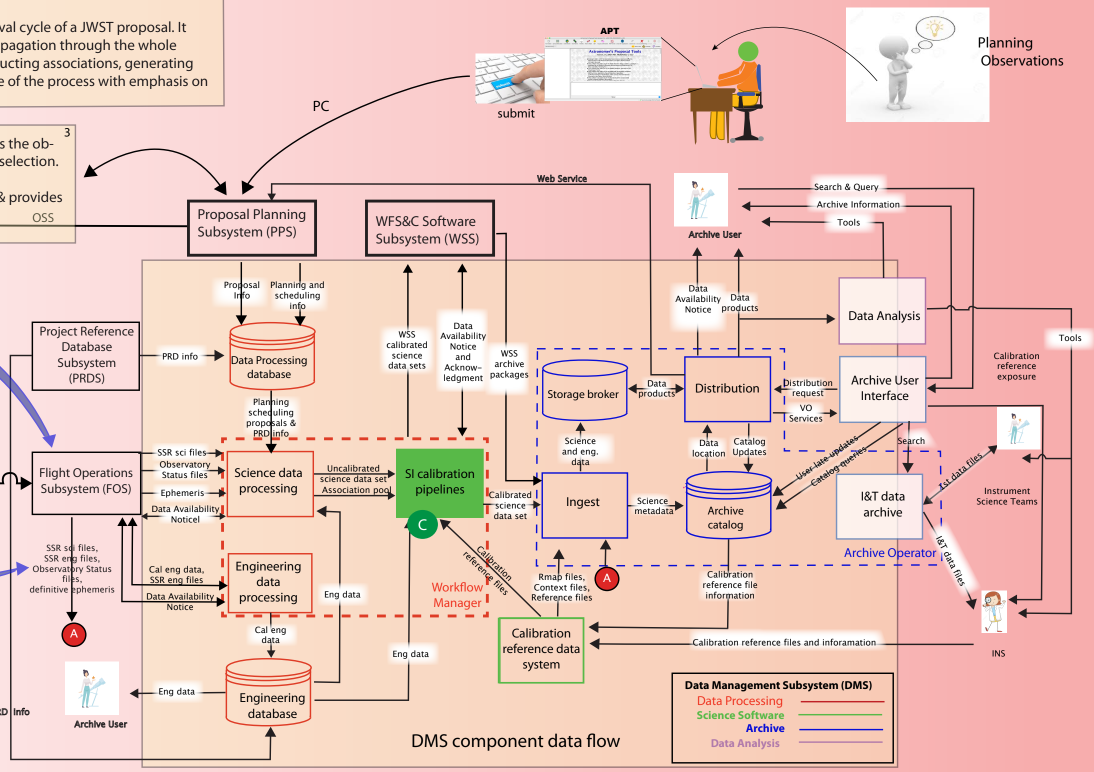
Flight Operations Subsystem (FOS) sets up simulation plans for different guiding scenarios: Deep Space Network (DSN) discrete contact schedule; plan for Solid State Recorder (SSR) playback and contact schedule upload; ephemeris; Observation Plans(OP) upload; clock correlation; dump on-board logs; station keeping visits; and management moment unloads.

The proposals are run against the JWST simulators: **Observatory Test Bed (OTB)** or **Engineering Model Test Bed (EMTB),** by the Mission Operation team using



Generated Flight Dynamics, S&OC products, maneuver plans, updated ephemeris files, and engineering data are gathered and submitted to the Data Management Subsystem (DMS) for processing.

A DMS uses all the generated data to construct raw data and associations that are fed to the JWST Science Calibration Pipelines. Generated products are archived in the MAST subsystem and made available to users.



5

SI Calibration Pipelines development and testing

Validating that the calibration pipelines are generating the best quality science products is one of our top priorities. We have designed the testing infrastructure to support verification and validation and have implemented many of the tests.

The diagram to the right shows three different types of tests: unit test, regression tests, and validation tests integrated into the calibration pipeline development process. These run automatically at different stages of the development process to allow for fast development-testing-validation turnaround time and a higher level of confidence in the quality of our calibration products.

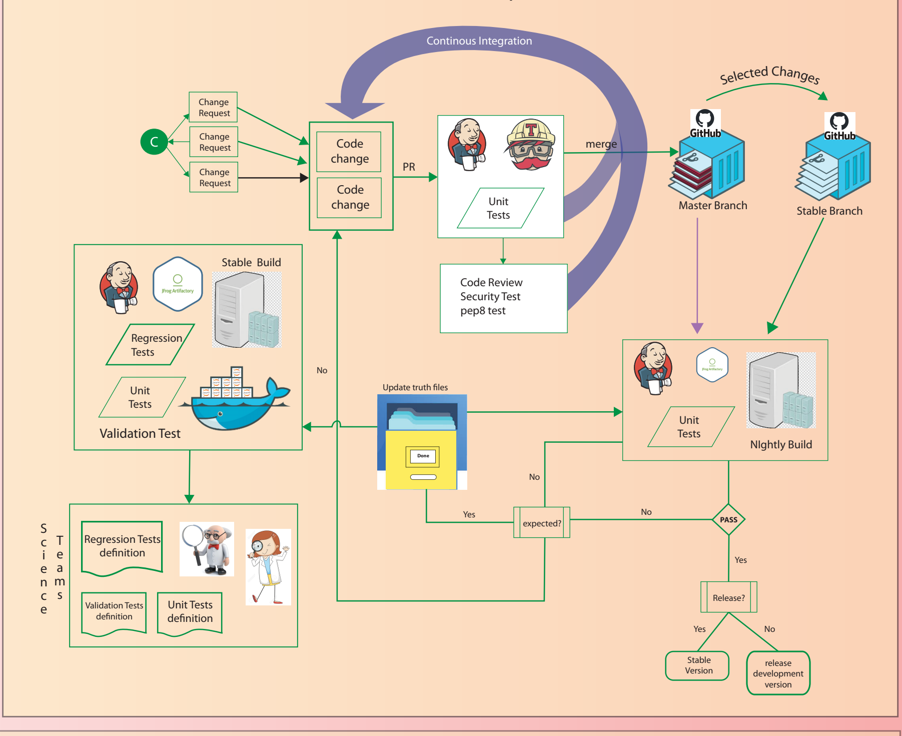
After finishing code development, a full set of unit tests for all the calibration steps is run with every pull request and within the continuous integration process. The process repeats until all tests pass. The code is merged into the master repository while a selected set of changes is also merged within the repository that contains a stable version of the calibration pipeline. Both of these versions run through a nightly build process automated using Jenkins and a suite of data stored in the Artifactory repository manager.

The failing test can be due to changes to the calibration code which might be expected or unexpected. In the first case, the truth files are updated with the newly generated data, and the test is rerun. In the second case, the nightly test is repeated, and if this fail again, a bug fix is filed.

Once a stable build is generated, the full set of unit tests, regression tests, and validation tests are run automatically using Jenkins. Validation tests are stored in docker containers and run via the jupyterhub science platform. In here, the tests, stored in Jupyter notebooks, are crafted by the instrument teams and will be manually reviewed. Data used for these tests is also stored in Artifactory.

Instrument scientists review the results of these tests and sign-off on the release of each of the operational builds of the Calibration pipeline.

Science Calibration Pipelines



6

Science Simulations

In support of end-to-end testing and since the OTB/EMTB produce test pattern detector data which is not adequate for validating the calibration pipeline, instrument teams are generating a set of simulations to match the proposals used for end-to-end testing. These simulations include dithers, background observations, and any other relevant observations that are needed for any of the generated associated products. These could include background observations and coronagraphic observations of the target and reference star at different orientations. Two simulators are currently used to create these simulations.:

The **Multi-Instrument Ramp Generator (Mirage,** https://mirage-data-simulator.readthedocs.io/en/latest/) is a Python software package creates simulated data for a significant subset of the science observing modes of NIRCam and NIRISS, including Imaging and Wide Field Slitless Spectroscopy (WFSS) data for both NIRCam and NIRISS. Mirage also supports imaging with the JWST Fine Guidance Sensor (FGS) used for engineering testing purposes. It generates uncalibrated JWST exposures and these can be reduced by the JWST Pipeline.

The **Miri Simulator (MIRISim,** http://miri.ster.kuleuven.be/bin/view/Public/MIRISim_Public) is a Python software package intended to allow MIRI users to gain experience with the data produced by the instrument. MIRIsim generates data files which emulate uncalibrated JWST exposures and can be reduced by the JWST Pipeline.