

The LSST Semantics of the Term: Pipeline

from Glossary?

There are two senses to the word **pipeline** in use

- the WBS sense as a deliverable software element, and
- the Middleware sense as an architectural element in a framework.

I would argue we already have these identified and pretty clearly defined:

- Pipeline Execution Framework -- a middleware framework for connecting together application processing modules and executing them in an automated way. There are several such frameworks, including IRAF, Opus, etc.
- LSST Pipeline Execution Framework -- a particular (default) pipeline framework used for standard LSST processing. This is our middleware and is python and C++, plus off-the-shelf packages.
- (XYZ) Pipeline - a particular WBS element that is a configured instance of the LSST PEX and application processing modules to be delivered as part of our MREFC construction effort and used in LSST operations.

It is true that within the context of the LSST PEX, we have the architectural elements of Production, Pipeline, Stage, Slice and that Pipeline is thus overloaded. In this context Pipelines are software components/classes. We also in this context have:

- Black box Pipeline - a particular instance of a Pipeline that may or may not have been built on a PEX and was definitely not built on the LSST PEX,

that we are incorporating into our system for use in LSST processing.

All that is clear and semantically consistent, we just need to do a better job of explaining it.

Extracted from email from Jeff Kantor to Ray Plante on 25 Feb 2009 08:42:02 on the subject: [DC3] DC3a Pipeline/Stage boundaries