PROPOSAL for reimplementation of the DESDM workflow

This is a description of the tasks needed to produce a new DESDM workflow that will fix the known deficiencies of the current system. This is not a prioritized list. Due to the intertwined nature of the current workflow it is very difficult to fix one part in isolation. There are two largely separable components. The first is the workflow layer and file structure. The second is the wrapper layer. In DESDM we are using publically available science codes to do many of the processing steps. This means that in order to incorporate them into a pipeline they require a wrapper layer that handles providing the input data, program configuration and collection of provenance information. This is a similar process to integration of one of the HEP Monte Carlo simulations packages like HERWIG into an HEP experiments software framework.

We will initially focus on the pipelines that are needed to accomplish the initial science verification and start of DES operations. This is the calibration pipeline (Precal), the nightly processing (Firstcut) and the Supernova (SN). We will then work through the rest. The breakdown of effort would be as follows:

- New Wrapper layer – Fermilab
- Workflow and file structure – Primarily NCSA but with input on design by Fermilab scientists and computing professionals.

Task List:

- New wrappers:
    - Takes specific format wrapper configuration files that describe inputs, configuration, auxiliary files, and outputs.
    - Outputs data and software provenance as well as metadata about the output.
    - Specific format and rules need to be polished.
    - Effort Estimates:
        - Software designers:
            - Initial design:
                - Fermi: 2 FTE for 1.5 months
                - NCSA: 0.3 FTE for 1.5 months
            - Throughout implementation: Fermi 0.1 FTE
        - Science oversight:
            - Fermi: 0.5 FTE throughout
        - Wrapper writers:
            - Fermi: ~70 wrappers at 1 week per wrapper = 17.5 FTE months

- Framework changes:
    - Implement new unique filenames and new processing directory structure.
        - Codes shall not overwrite inputs
        - Rerunning a pipeline does not overwrite previous processing.
    - Add flag per file in operations DB to indicate file lifecycle. Consider adding same flag to objects tables.
    - New data and software provenance tracking:
        - Need new DB tables
        - New software to ingest into tables
    - File IDs should no longer be created automatically by the DB. They should be constructed from unique filename. Object IDs should also constructed: catalog id + object number (object number is unique number within catalog). Algorithm to be standardized. DB will enforce uniqueness.
        - Natural keys are more easily utilized by the system than surrogate keys.
    - Target node workflow:
        - Orch will make a workflow plan for the job before submitting the job.
            - Filenames will be based upon patterns if depend upon previous tasks in the job.
            - If *any* wrapper exits with non-zero exit code, timeouts, or dies unexpectedly, workflow halts. This means that other CCD's or bands will not continue even if science workflow would allow.
            - Job brings back outputs from wrappers (provenance, metadata, science files, etc). Intermediates/tmp files are brought back on failure or operator flag.
            - Primary method of conveying metadata between wrappers within a single job is by fits header keywords. If need is discovered during wrapper writing, may need a way to pass information between wrappers within a single job in cases where need special configuration value. (Orch ability to read output provenance of previous wrappers? Pass value in tiny file?)
        - Use existing new QC stdout/stderr parsing at runtime into Oracle cluster for operator information. Not used for determination of whether to continue to next wrapper in current job.
            - Note: If program issues what should be a "status5", it shall exit with a non-zero exit code.
            - Note: Should allow a flag to turn off QC parsing.
    - Submit side processing framework changes

        - New file transfer technique (due to file/dir naming changes)
        - Writing of workflow plan instead of elf workflow.
            - Handle “loops” iterating same code over multiple CCDs/bands.
        - Post-job moving of files into new processing directory.
        - Ingest wrapper outputs (provenance and meta data).
    - Effort Estimates:
        - Fermi:
            - DES scientists 2 FTE months
            - Software designers 2 FTE for 0.2 months
        - NCSA:
            - 0.5 FTE for 4 months DBA
            - 1.5 FTE for 2 weeks design
            - 2 FTE for 5 months initial implementation

- Refactoring of non-wrapper codes:
    - Exposure-based changes i.e creation of astrostds catalogs
    - Specialized query codes
    - Handle new file and object ids (also catalog ingestion codes)
    - Standardization of header keywords
        - Removing paths
        - Formats
    - Estimates: covered in wrapper and framework estimates.

<u>Total Fermi Effort</u>

Scientist:	9.5 FTE-months
Software Designer:	5.2 FTE-months
Software implementer:	17.5 FTE-months

<u>Total NCSA Effort</u>

Software Designer:	1.2 FTE-months
Database developer:	2 FTE-months
Software implementer:	10 FTE-months