Data Management Planning and Reporting Process

Data Management Planning Process

As the Data Management progresses from R&D to Construction, the planning process has become more formal. For the primary work plans during R&D, the following process has been used:

1. From overall flow-down of System Requirements to DM, overall LSST Program Schedules, and requirements analysis and design activities within DM, a ?DM Risk Register has been created that assesses the main programmatic and technical risks as understood at this time. These risks are ranked according to impact and likelihood to establish an exposure, which is the primary means of prioritizing mitigation activities.

2. The ?DM Roadmap "bins" planned mitigation activities into "coarse-grained" 6-month periods. These activities encompass all major DM R&D work in all areas (DM System Management and Engineering, Applications, Middleware, Infrastructure).

3. The DM Project Manager creates and maintains data challenge release and management/system engineering project plans that encompass the entire Roadmap and add the next level of detailed work planning. In addition, the plans cover activities specific to preparing and conducting formal design reviews. The granularity of the data challenge release activities in this plan are 1 to 3 months each, non-release activities are finger grained from 1 week to 1 month. Each activity includes the activity description, expected start and end dates, and resources assigned (with primary resource listed first). Inter-activity dependencies are also captured to allow for critical path method scheduling and analysis. Starting with the Winter 2013 release, these plans have been moved from MS Project to the LSST Project Management Control System (PMCS) and are regularly exported in pdf form as ?LSST PMCS Plans

4. At the start of each 6-month period (aka each new Winter or Summer Release) the entire DM team accomplishes a very detailed scoping and planning exercise to get the most detailed work plan, with individual activity granularity from 1 week to 1 month. This process is done via meetings, interactive communications, and requirements/design reviews, outside the PMCS. Then plan is incorporated into the PMCS. Once work starts on the release, this level of plan is tracked on a weekly basis, and the status is rolled up to the PMCS (see DM Reporting Process below). This plan is known as a Release Plan.

The PMCS actually includes all activities to be performed during the current release period, even those that are not directly part of the data production and release (e.g. LSST and Agency Review activities). While the plan includes both release and other activities, the plan is coded in PMCS so that those activities that are part of the data production and release can be easily identified in filtered views of the plan.

The diagram below depicts the release planning process:

The release plan covers all activities associated with producing a data challenge release (non-release activities are maintained in the PMCS plans as described above). A release plan will include management, simulation, applications, middleware, infrastructure, integration and test, data quality, and production activities.

Data release plans have several major "phases" of work as follows, and the definitions in the Guideline sections apply, i.e.:

- Scoping/Planning: This occurs as a single, release-wide 1 month up front effort, and then is adjusted monthly throughout the release
- Design: This occurs on each major piece of the release (software elements design, data definition/acquisition, infrastructure definition) as we schedule it, and is adjusted monthly until the "feature freeze" before integration. It occurs iteratively and in parallel with implementation.
- Implementation: This occurs on each major piece of the system "piece-meal", i.e. the major piece is really made up of many smaller pieces, each of which gets implemented in an order defined by the developer/team. At the end of an implementation activity, the given software element is functional and unit tested, documented, and merged to the master, but it is not yet integrated.
- Integration: This is defined into 3 sub-levels, and ends with a code freeze. It occurs on major pieces of the software as elements finish implementation and enter into integration:
    - ♦ Stack integration Test: This is the first level of integration, and just ensures that the software can be run integrated into the stack and build environment and produces "not insane" data results. These are nightly runs that typically process very small, pre-defined data sets. They typically run once for each of multiple versions of the stack: the last released (stable), the current beta, the most recent integration (latest ticket branch merges), and the latest development (master/head, i.e. the bleeding edge).
    - ♦ Data Quality and Algorithmic Test: These are weekly and/or specialized runs that typically process small to medium sized data sets and are specifically oriented at ensuring the data outputs are acceptable from a data quality standpoint. They typically run some subset of the stack versions run in the nightly integration runs.
    - ♦ Performance Test: These are mini-production runs that typically process medium size data sets and are specifically designed to ensure we can do production at the volume we need, and in time to meet the schedule. These also regression test that the middleware still works with the stack changes and at volume. They typically only run the current beta or latest integration versions of the stack.
- Production: This is divided into sub-levels, and occurs one or more times on the entire release once all the parts are integrated
    - ♦ Large-scale production runs
- Data Quality Analysis and Report
    - ♦ Post-production data analysis, including moving data into user interfaces and documentation
    - ♦ Final report

This might sound somewhat waterfall due to the fact that language requires things to be described sequentially, but it really isn't . Not all parts of the code move through these phases at one time, nor do we lock them in stone when we first define them, so this is more iterative than waterfall. The point to having the terminology above is to distinguish between certain types of activities and to define when a given activity in the plan is started or complete.

Data Management Reporting Process

The DM project (including all DM contracted institutions) is legally required to provide monthly status reports to LSST senior management, the LSSTC Board and AURA AMCL. These monthly reports are also provided to the funding agencies, and are audited annually by Federal auditors. Non-compliance with reporting requirements is cause for contract termination.

The DM project uses the following process for monthly reporting:

1. As discussed above, the DM Project Manager maintains data challenge release and management/system engineering plans in the LSST Project Management Control System (PMCS). Each activity in the plan includes the name of the activity, expected start and end dates, and assignees (with primary/lead person listed first).

2. The LSST Project provides a web-based tool for weekly status reporting against the plan. The status collected included % complete, actual and expected start and finish dates, and comments. The tool can be accessed at:

?LSST Status Reporting Tool

Instructions for the tool can be accessed via a link off the tool login page.

The standard weekly status reporting cycle is as follows:

Monday: Project Manager creates pre- and post-reschedule baselines of the plan in PMCS and publishes pdf files of the current fiscal year plans to ?LSST PMCS Plans.

Tuesday: DM team discusses status, issues, blockages, progress.

Tuesday - Thursday: DM Project Manager updates plan with revised activities, resource assignments, etc. Project Controls Specialist fixes bugs, adds enhancements, tests, and releases new version of status tool.

Friday - Sunday: All primary resources (i.e. lead individuals) report status via the tool. This status updates the PMCS plan with the associated information.

3. The DM Project Manager exports the list of recently completed activities into tables, sorted by institution, and creates an "extended" progress report, which the DM institutional leads add narrative to. The DM Project Manager posts this report in DocuShare. This report has been examined every year by Federal auditors to check that it shows work performed/progress by each contracted DM institution.

4. From the extended report the DM Project Manager creates a "summary report" for submission to senior management and for inclusion in the monthly report to the LSST Board and AMCL. These reports are also included in the quarterly and annual reports to the funding agencies. This summary report is also posted in DocuShare by the DM Project Manager.

This process satisfied both Federal auditors and LSST management. However, the process has always had some issues with consistency in the plans versus data challenge release work activities as they naturally evolve during the course of development, with regularity and consistency of reporting by the team, and with the ease of "rolling up" the detailed status provided by the team to the reports and plans.

We have adjusted the status process so that we get the accuracy, consistency, and status reports/plan integration needed for Earned Value Management, which will be required during the construction phase. We are now deciding how to adjust the process, which is depicted below:

Status Reporting Guidelines

In each status reporting Trac page, we have established "default" status milestones for various activity types:

Guide on Reporting Percentage Completed

For Software Design activities:

25% if requirements are defined;

50% if we have completed a Trac page and/or UML with design/comments

75% if we have completed a prototype implementation (if necessary)

100% if the design is reviewed and approved, ticket opened for implementation

For Software Implementation activities:

25% external interface defined

50% unit tests written

75% coded on branch and passes unit tests

100% code is documented, reviewed, and approved (merged to master)

For Software Integration activities:

50% packages are tagged, stack builds/runs in buildbot triggered runs (with errors)

75% packages are tagged, stack builds/runs in buildbot triggered runs (without errors)

100% packages declared and distribution-server-tagged, ready for next higher level Integration or Production runs

For other non-software, non-level of effort activities:

use 25% if the deliverable's specification is completed;

use 50% if the deliverable is implemented awaiting review/approval;

use 75% if the deliverable has been reviewed and is in revision;

use 100% if the deliverable is approved and posted to appropriate repository.

For level of effort activities:

% of duration elapsed is % complete

However, there has been some difficulty in applying these guidelines by the team, as it is sometimes the case that the process varies (e.g. unit test development occurs after initial coding). Another variation is that activities may sometimes encompass design, coding, and testing in one activity. In order to allow for these variations and improve consistency, the following guidelines are also relevant going forward:

As the Release Plan is developed, and individuals are assigned, the lead person is responsible for either adopting these milestones for that activity, or proposing another set that matches the activity at what they consider roughly 25, 50, 75, and 100% complete states. Alternatively, the activity can be subdivided into smaller activities of no more than 2 weeks, for which the states will be 0, 100% complete only. In any case, whether using the default milestones or developer-provided ones, the activities must encompass all design, development, unit test, and documentation associated with the overall activity.

In order to improve synchronization of the Release Plans in PMCS and the work as it is progressing, the plans will be revisited monthly. This allows for activities to be deferred, split, merge, emerge, etc.