

Agile software development in an earned value world: a survival guide

Jeffrey Kantor^a, Kevin Long^b, Jacek Becla^c, FrossieEconomou^a, Margaret Gelman^d, Mario Juric^e, Ron Lambert^a, K. Simon Krughoff^e, John D. Swinbank^f, and XiuqinWu^g

^aLSST Project Management Office, Tucson, AZ, U.S.A.

^bLonghorn Industries, U.S.A.

^cSLAC National Laboratory, Menlo Park, CA, U.S.A.

^dUniversity of Illinois at Urbana-Champaign, U.S.A.

^eUniversity of Washington, Seattle, WA, U.S.A.

^fPrinceton University, Princeton, NJ, U.S.A.

^gInfrared Processing and Analysis Center, California Institute of Technology, Pasadena, CA, U.S.A.

ABSTRACT

Agile methodologies are current best practice in software development. They are favored for, among other reasons, preventing premature optimization by taking a somewhat short-term focus, and allowing frequent replans/reprioritizations of upcoming development work based on recent results and current backlog. At the same time, funding agencies prescribe earned value management accounting for large projects which, these days, inevitably include substantial software components. Earned Value approaches emphasize a more comprehensive and typically longer-range plan, and tend to characterize frequent replans and reprioritizations as indicative of problems. Here we describe the planning, execution and reporting framework used by the LSST Data Management team, that navigates these opposite tensions.

Keywords: Project Management, Software Engineering, Agile Process, Earned Value

1. INTRODUCTION

The Large Synoptic Survey Telescope (LSST)^{*} project is a proposed large-aperture, wide-field, ground-based telescope that will survey half the sky every few nights in six optical bands. The 8.4-meter telescope will be located in the Andes mountains near La Serena, Chile.^[1] The 3.2 Gpixel camera will take 6.4 GB images every 15 seconds, resulting in 15 TB of new raw image data per night. The focal plane consists of 189 science CCDs with 16 channels each to achieve a readout of all data in 2 seconds. After taking into account unusable nights, slew time, and other factors, this will produce 7 PB of new data each year to be processed.^[2]

The LSST Program is composed of four subsystem projects as well as an overall Program Management Office, Systems Engineering effort, and Commissioning project, each of which requires project management and control. The subsystem projects are:

^{*} <http://lsst.org>

- Telescope and Site
- Camera
- Data Management
- Education and Public Outreach

This paper focuses on the management and software development processes for the LSST Data Management System (DMS)^[3] but software development in some other parts of the LSST project follow similar processes. Increasingly, “Big Science” astronomy and astro-physics projects are no longer treating the data management aspect as an afterthought to be done once the telescope is in place, and LSST definitely falls into this class. LSST features a large data management project by optical astronomy standards, processing and archiving over 70 petabytes of image data and producing over 20 petabytes of catalogs annually, and generating 2 million transient alerts per night. Over the 6-year construction and commissioning phase, the DM project is estimated to require 600,000 hours of engineering effort. In total, the DMS cost is approximately 60% hardware/system software and 40% labor.

LSST Data Management combines: large-scale, multi-year equipment, communications, and services procurements; off-the-shelf and customized middleware; custom scientific, algorithm-intensive software; all integrated into a complex system. Managing the entire effort requires more formal mechanisms, and if, as LSST is, the project is funded by International and/or Federal entities, that includes the requirement to perform Earned Value Management (EVM)[†]. EVM requires significant up-front planning of the entire effort, to create a resource-loaded, scheduled baseline plan, against which progress is tracked and analyzed in terms of cost and schedule. EVM also discourages purely “level of effort” planning, where the plan only indicates a team working for some time period and deliverables are not specified.

Meanwhile, software development best practice is currently to employ an “agile process”[‡], which typically implements iterations of short, intensive “sprints” to develop the software incrementally, and to address emerging requirements and iterative design. Agile process does not encourage long-term up front planning, rather it focuses on the near-term, with the longer-term aspects left “fuzzy” or level of effort against a prioritized backlog.

So, the question this paper attempts to address is “How to marry these two in such a way as to get the benefit of both?” There have certainly been prior attempts to address this question, as a search of the web will attest, and the difficulty in making this “match” is well documented.^[4] In LSST, a fundamental requirement is that the Agile – EVMS integrated process is fully EIA-748² compliant. Sadly, many of the attempts to integrate coming from the Agile community re-invent the definition of EVM in ways that are fundamentally non-compliant, and as such are non-starters for LSST.

2. LSST DATA MANAGEMENT EVM PLANNING PROCESS

Since the DM software elements are a mix of off-the-shelf system software and adapted middleware, plus custom applications, each element has many interfaces to the other elements. In addition, the data quality requirements of LSST necessitate that some of the algorithms require some advancement over the current state of the art, and all of them require scalability to the LSST data volumes. Due to complexity of the system and to support EVM, a multi-level Work Breakdown structure has been created, as shown in Figure 1. Each institution in the DM team is typically responsible for 1 or more Level 2 WBS elements, and each institution has a Technical/Control Account Manager (T/CAM) responsible for planning, estimating, monitoring and EVM reporting for that Level 2 WBS element.

[†] Earned Value Management Systems ANSI/EIA-748-C Intent Guide, National Defense Industrial Association (NDIA), April 29 2014

[‡] https://en.wikipedia.org/wiki/Agile_software_development

Application Layer (LDM-151)

- Scientific Layer
- Pipelines constructed from reusable, standard “parts”, i.e. Application Framework
- Data Products representations standardized
- Metadata extendable without schema change
- Object-oriented, python, C++ Custom Software

Middleware Layer (LDM-152)

- Portability to clusters, grid, other
- Provide standard services so applications behave consistently (e.g. provenance)
- Preserve performance (<1% overhead)
- Custom Software on top of Open Source, Off-the-shelf Software

Infrastructure Layer (LDM-129)

- Distributed Platform
- Different sites specialized for real-time alerting, data release production, peta-scale data access
- Off-the-shelf, Commercial Hardware & Software, Custom Integration

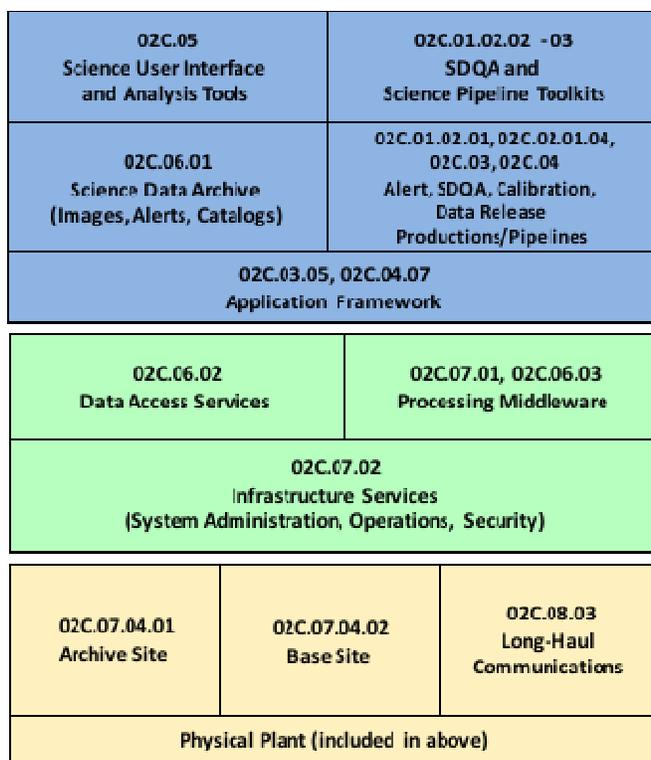


Figure 1: Data Management System Architecture and WBS

The LSST Management and System Engineering process shown in Figure 2 flows down LSST system requirements and operations plans, plus LSST project-level risks and milestones to create the DM System requirements. From this set of requirements, the DM team has created a set of design documents covering the DM System as a whole, and subsystem level designs for the infrastructure (hardware and systems software), the middleware (software for parallel processing, database, data transfer, etc.), and the applications (software pipelines, algorithms, data structures, user interfaces and tools).

For these reasons, the software is being developed via a series of prioritized, incremental releases, with functionality, performance, and data quality increasing over time. Every release cycle, the T/CAM of each major area of work (Level 2 WBS) maps the development work in that area into 1 to 3-month development activities. The T/CAM then captures the inter-dependencies for critical path scheduling, and finally loads the activities with the resources needed to perform the work. This plan is captured in the LSST Project Management Control System (PMCS), where the full EVM process is managed.

As the release cycle proceeds, the T/CAM creates an even more detailed plan for each activity, adding more fine-grained development tasks and very fine-grained development steps, the latter as short as 2 – 20 days. It is in this last part of the plan where the Agile Process is most apparent, which will be discussed in a later section below.^[5]

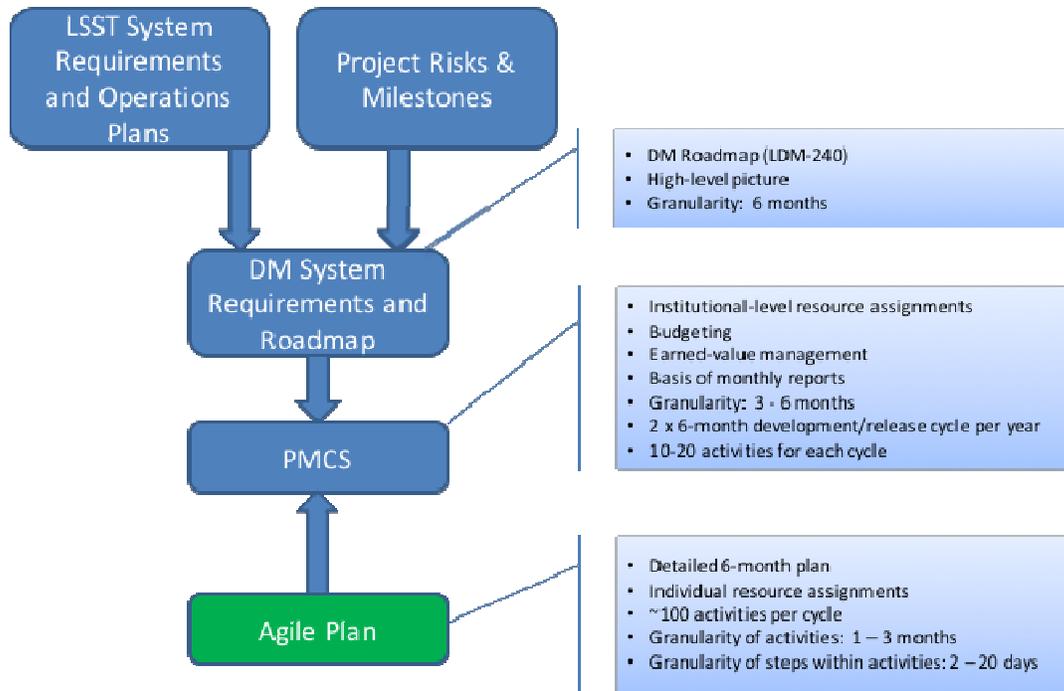


Figure 2: Data Management EVM Planning Process/Levels

A "Roadmap"^[6] has been developed which defines the major milestones and key performance metrics (KPM) that will be achieved in each release and is depicted in Figures 3 and 4. The granularity of the Roadmap is Level 3 or greater WBS element milestones or KPM, matrixed by incremental release cycle of 3 – 6 months.

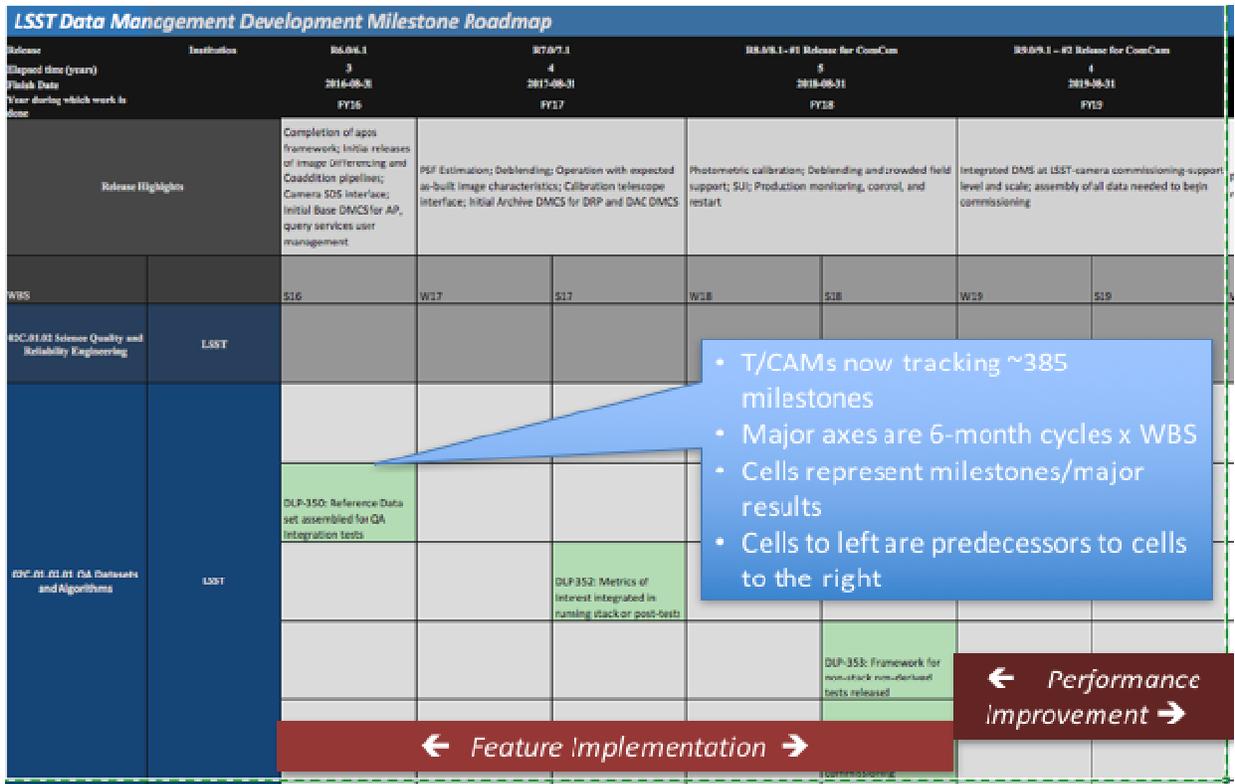


Figure 3: LSST Software Development Roadmap Milestones(LDM-240)

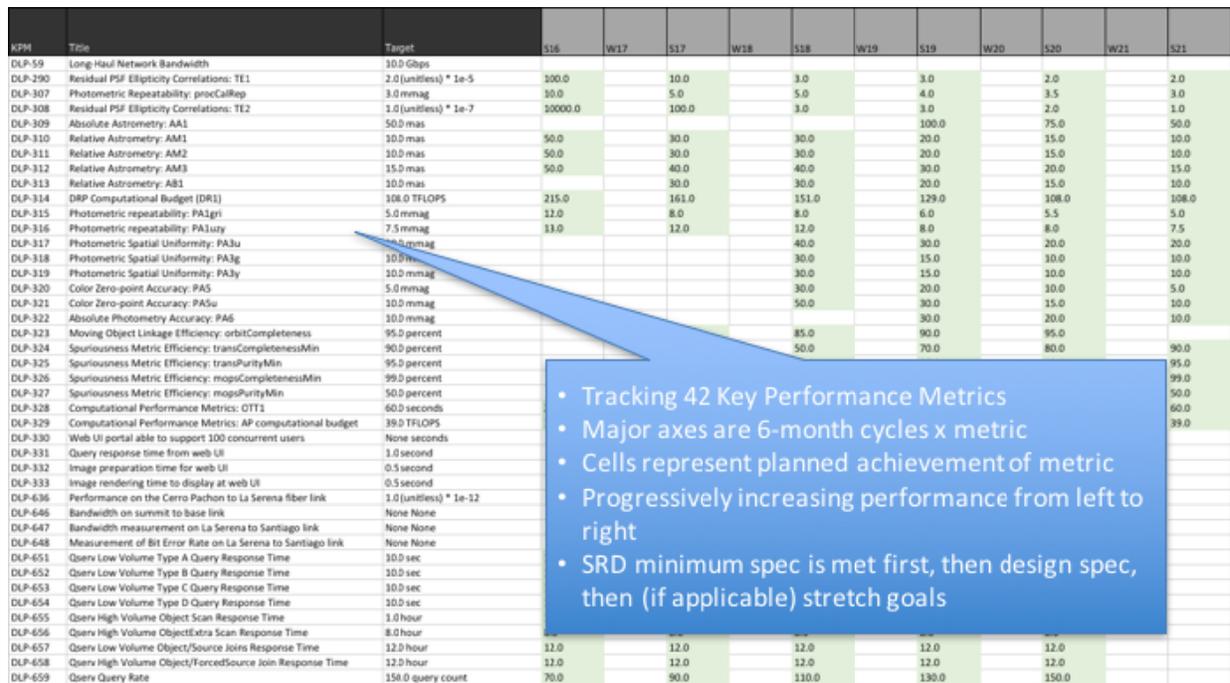


Figure 4: LDM-240 DM Key Performance Metrics

3. ORGANIZATION

Tools

The LSST EVMS depicted in Figure 5 encompasses the processes and technical systems that integrate and quantitatively measure project cost and schedule performance against a project execution plan known as the Performance Measurement Baseline (PMB). Measuring and maintaining the PMB uses all of these tools in conjunction and under strict guidelines.

The EVMS technical systems which are used to reach this goal include the following:

- CASNet – AURA Financial System
- Primavera Project Planner (V8.3) - Schedule
- Cobra (V5.1) - EVMS
- Docushare – Document Control, Document Repository
- Drupal – LSST Change Control Board Interface
- Risk Register - Internally developed software used to manage risks and opportunities
- eCAM – Internally developed software used as the T/CAM’s interface to all cost and schedule data
- JIRA⁵ – Software project tracking with agile tooling for software teams

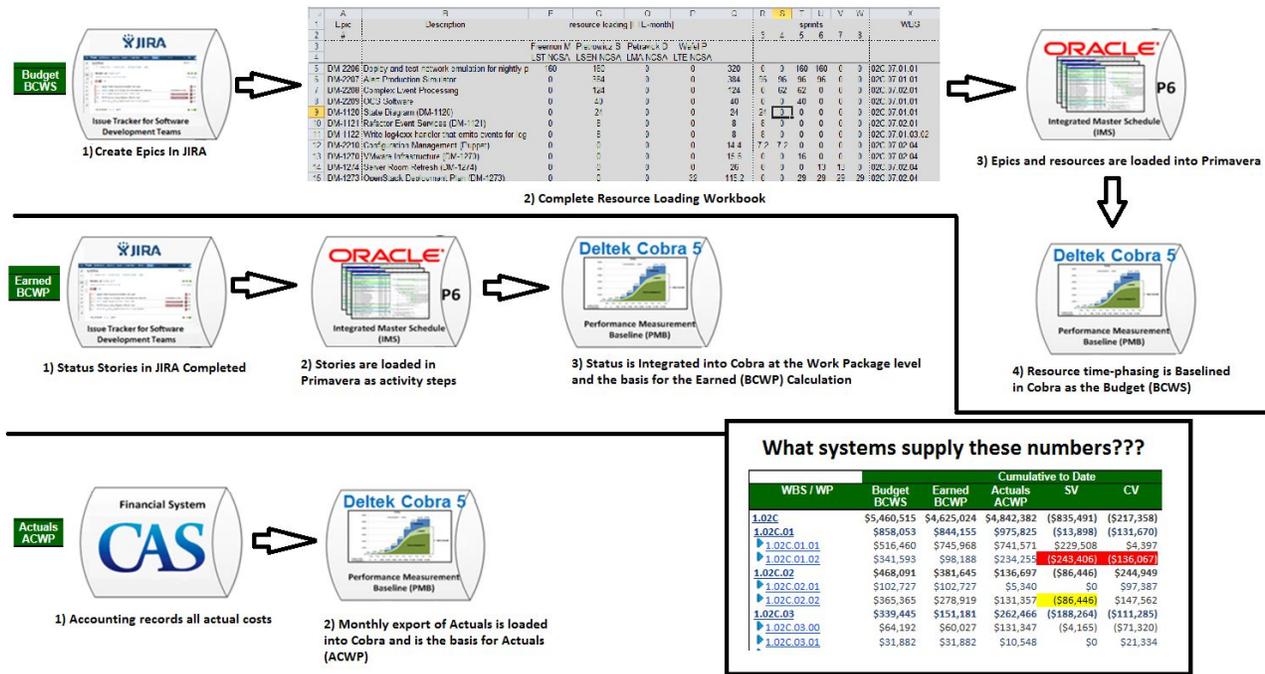


Figure 5 LSST PMCS tool set

Work Breakdown Structure

The LSST WBS is a product-oriented, hierarchical structure that identifies the hardware, software, services, and all other deliverables required to achieve the LSST Project. The LSST WBS is the primary structure for managing

⁵ <https://jira.lsstcorp.org/secure/Dashboard.jspa>

performance against the PMB and is the framework for defining and assigning work, developing schedules, estimating and budgeting, and controlling changes. The WBS elements that comprise LSST control accounts are defined at level 4 but in some cases details in the IMS are at a lower level.

The WBS dictionary is a narrative attached to the WBS that describes the scope, deliverables, and associated key milestones of each work element identified. The WBS dictionary defines each element to at least the control account level in terms of the content of the work to be performed.

The WBS has been added to JIRA as a custom field associated with Epics, Meta-Epics, and Milestones. This is a required field that ensures the element in JIRA is associated with the correct WBS in the PMCS tools.

Organizational Breakdown Structure

The LSST Organizational Breakdown Structure (OBS) shown in Figure 6 is a hierarchical structure that defines the Institutions/major contracts where the work will be planned and controlled. The OBS identifies the accountability, responsibility, management, and approvals of all authorized work scope. While the OBS is not directly called out in JIRA it is possible to infer the OBS based on the “Team” code that is assigned to an Epic.

Code	Description
1.	LSST Construction Phase
1.01	LSST
1.02	SLAC
1.03	IPAC
1.04	NCSA
1.05	UW
1.06	Princeton
1.07	NOAO
1.08	Adler
1.09	UCD
1.10	Arcadis
1.11	UA
1.12	(unassigned)
1.13	RAL
1.14	Purdue
1.15	PFLOW
1.16	REUNA

Figure 6 LSST OBS Structure

Control Accounts

As shown in Figure 7, a control account (aka cost account) is a management control point at which budgets (time-phased resource plans) and actual costs are summarized and compared to earned value for management control purposes. A control account is a natural management point for planning and control since it represents the work assigned to one responsible organizational entity for a single program WBS element. A control account manager maintains responsibility for an individual control account and all technical, cost, and schedule elements in work Packages below it.

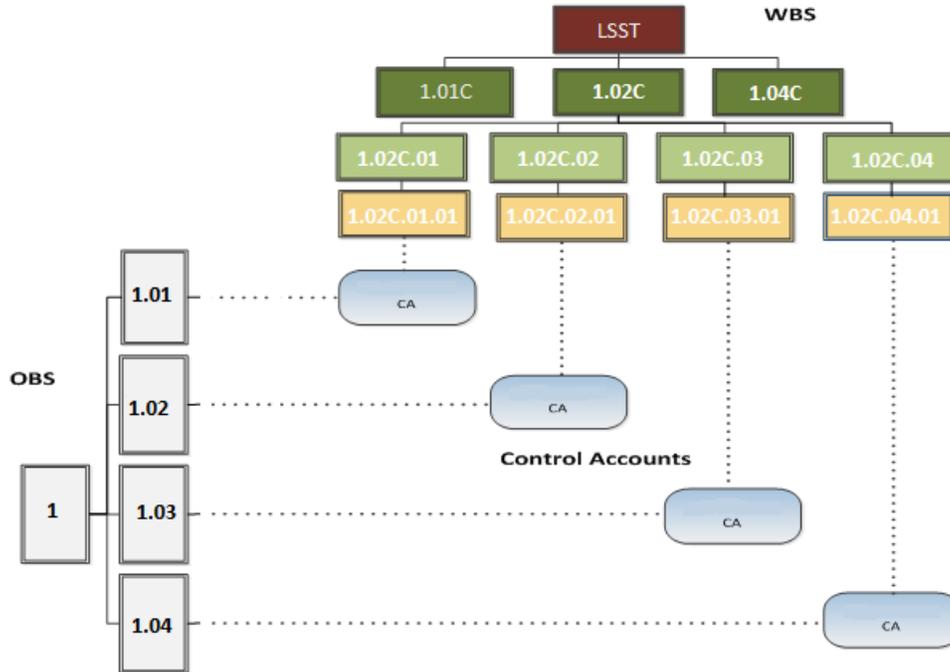


Figure 7 Graphical Representation of Control Accounts.

Work Packages

LSST Work Packages are the intersection of the current cycle and a WBS element (whereas planning packages are the intersections of future cycles and WBS elements). Work Packages are defined at the level directly under the Control Account and specifies what work is planned, measures progress on that work, and computes the associated earned value. Actuals for LSST will be aggregated and loaded at the work package level. Once Epics from JIRA are imported into the IMS and are resource loaded they will be assigned to the current active Work Package for their assigned WBS. Many activities can be assigned to one Work Package which roll up to represent the total dollars and hours assigned to the Work Packages. If changes need to be made to the baseline resulting from rolling wave planning, a change request will be prepared and submitted for management review and approval. Work Packages are configured using the following criteria:

- Has a limited duration within a reasonably short time span
- Has scheduled start and completion dates
- Has resource requirement separated (e.g., labor, material, contracts) in a way that allows the EV reporting process to accurately measure progress
- Has a budget or an assigned value expressed in hours and/or dollars
- Reflects the way in which work is conducted and has meaningful work products
- Has a one to one correlation to an accounting charge number in the LSST accounting system
- Uses a single EV method

Planning Packages

LSST Planning Packages are defined at the level directly under the Control Account and specifies what work is planned. Planning Packages are reserved for future activities that cannot be clearly defined when the project baseline is set. Work that is beyond the current detailed planning period will reside in planning packages until they are converted to detailed work plans per a rolling wave process. The Planning Packages consist of a work scope, schedule, and time-phased budget normally at a higher level than individual Work Packages. Planning Packages do not require the detail found in Work Packages since, by definition, such details are not known. The Data Management Long Range Planning (DLP) project specifies the key milestones that are tied to planning packages in the IMS.

Technical/Control Account Managers (T/CAMs)

The LSST T/CAMs are responsible for the planning and management of the technical scope, schedule, and budget for assigned control accounts. They will provide timely input to the Project Manager in the formats described in this plan and will keep the project management staff informed of their work progress and issues or concerns, risk assessment, tracking methods, variance analysis, and estimate-to-complete/estimate-at-complete management processes.

Milestones

All significant development stages of each Planning Package are tracked through milestones. Each Planning Package has at least one milestone, used to mark the completion of a given Package. It is not uncommon to have multiple milestones per Package to track progress along the way. Each milestone has a description, and due date; it has no duration, and no resources assigned to it.

Critical Milestones are defined as Level 1 milestones (NSF reporting) and are watched and reported on monthly in the LSST monthly construction report. Level 2 milestones are coded to be managed at the project level, and Level 3 and 4 are watched at the WBS subsystem and T/CAM levels respectively.

Each activity within a Planning Package that involves a cross-team dependency is required to end with a milestone; this includes activities related to delivering software components, hardware, and services. Relationships between milestones, as well as between milestones and planning packages are captured in JIRA: milestones typically block planning packages, milestones can also relate to other milestones.

Figure 8 depicts the relationships between all of the PMCS Schedule, EVM, and JIRA Agile objects described in this section, including control accounts, work packages, planning packages, and milestones.

Schedule Planning Object	EVM Object	JIRA Agile Object
Work Breakdown Structure Level 4	Cost Account	"WBS" Custom Field
"Cobra WP" Activity Code Field	Work Package/Planning Package	Cycle, Fix Version
Activity	N/A – Many to one relationship between activities and Work Packages	Epic
Activity Step	Sum of weights used to calculate % Complete	Story, Bug, Improvement
Milestone		Milestone
Milestone (KPM)		Key Performance Metric
Assignee	Resource	Assignee

Figure 8 Field mapping across the PMCS

4. PLANNING, SCHEDULING, AND BUDGETING

Integrated Master Schedule

The Integrated Master Schedule (IMS) is a network of tasks linked from program start through program finish, reflecting the interdependencies between tasks and milestones. The IMS is the foundation of the performance measurement baseline (PMB) used to track progress, forecasts, and changes throughout program execution. The IMS enables the Critical Path Method that is used to identify the program critical path, as well as driving paths to major interim events or deliverables.

The IMS is subject to configuration control as part of the baseline. The change control process ensures that elements of the IMS, PMB, and technical baseline are kept synchronized. The schedule is baselined and status is posted in the Primavera scheduling tool (P6). The time-phased work packages are identified and used to build the planned value (PV) profiles in the PMB and the schedule status drives the EV reporting against PV based on physical percent complete and milestone completions.

Time Phased Budgets

The assignment of budgets to scheduled activities in the IMS produces a time phased budget plan. The resource loaded IMS is integrated into Cobra and as part of the PMB it allows budget performance to be measured. The Integrated budget in Cobra is defined as the Budget at Completion and is subdivided into control accounts, planning packages, and work packages.

Earned Value Techniques

Each established work package will have one corresponding Earned Value Technique (EVT). Any Epic from JIRA will be assigned an EVT of Percent Complete as the stories assigned to the Epic make it very easy to quantify the work that has been performed, and what is remaining. The following EVTs are used by LSST:

A – Level of Effort

This EVT assumes that when a work package starts, its progress will not deviate from the original budget spread. There are no limitations upon the applicability of this technique for measuring progress, but it is most suitable for only a small number of work packages that are by their nature unmeasurable. By definition, the value earned by an open work package using this EVT is equal to its to-date budget.

C - Percent Complete

Used to manually enter the completion status of the work package in percent each status period. When using method C the subordinate tasks in the schedule should have associated activity steps to help quantify the work performed.

K – Planning Package

This EVT results in always calculating an earned value of zero for the item. Use this EVT if one does not want the work package to earn any of its budget, regardless of its status.

Executing the Rolling Wave Plan

The estimated effort for future work exists in the time phased plan as 6-month cycle planning packages along with estimated DLP milestone completion dates. Planning packages were originally baselined from individually estimated via UML-based specifications. These planning Packages are converted to detailed work as planned in JIRA in the rolling wave process.

The Activities (epics) loaded into the IMS typically have durations on the order of 1 – 3 months. These Epics are estimated individually via developers through the Agile process. Each Epic is further detailed in steps (stories) of 2 – 20 days each. The development team T/CAMs plan each 6-month cycle in terms of Epics, Stories, Bugs, etc. in JIRA which doesn't require CCB approval. The T/CAMs and individual developers only work within JIRA using the agile process during the cycle. This allows them real-time access to record progress and plan future work while data input into the PMCS is controlled by project controls (monthly for status and as needed for change control). Resource estimation and assignment to Epics is done outside of JIRA (in Excel) and is integrated into PMCS. The T/CAMs complete a spreadsheet to assign resources and hours to the specific Epic(s)/Activity(s) being imported for the upcoming cycle as shown in Figure 9.

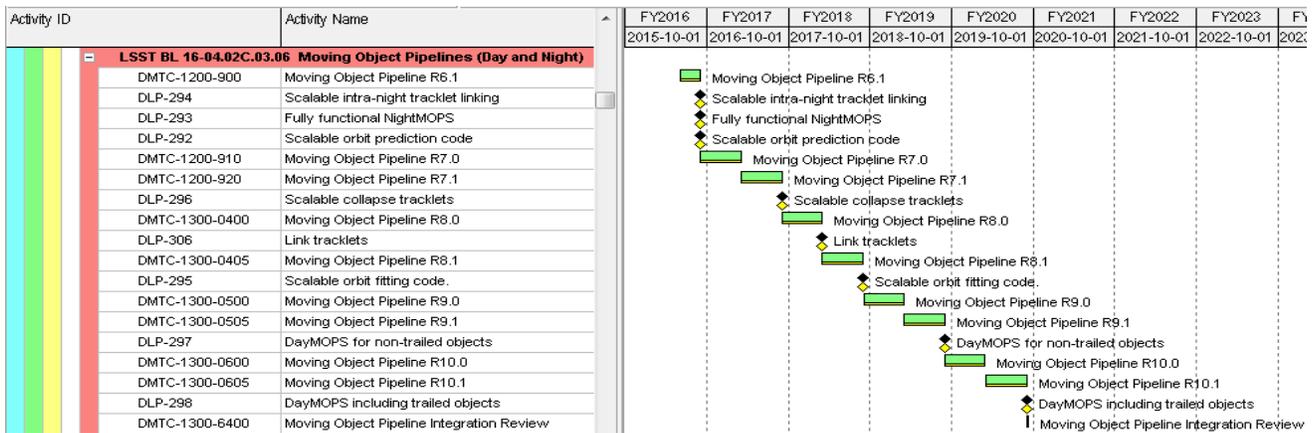


Figure 10 DM PMCS Prior to 6-month rolling wave plan

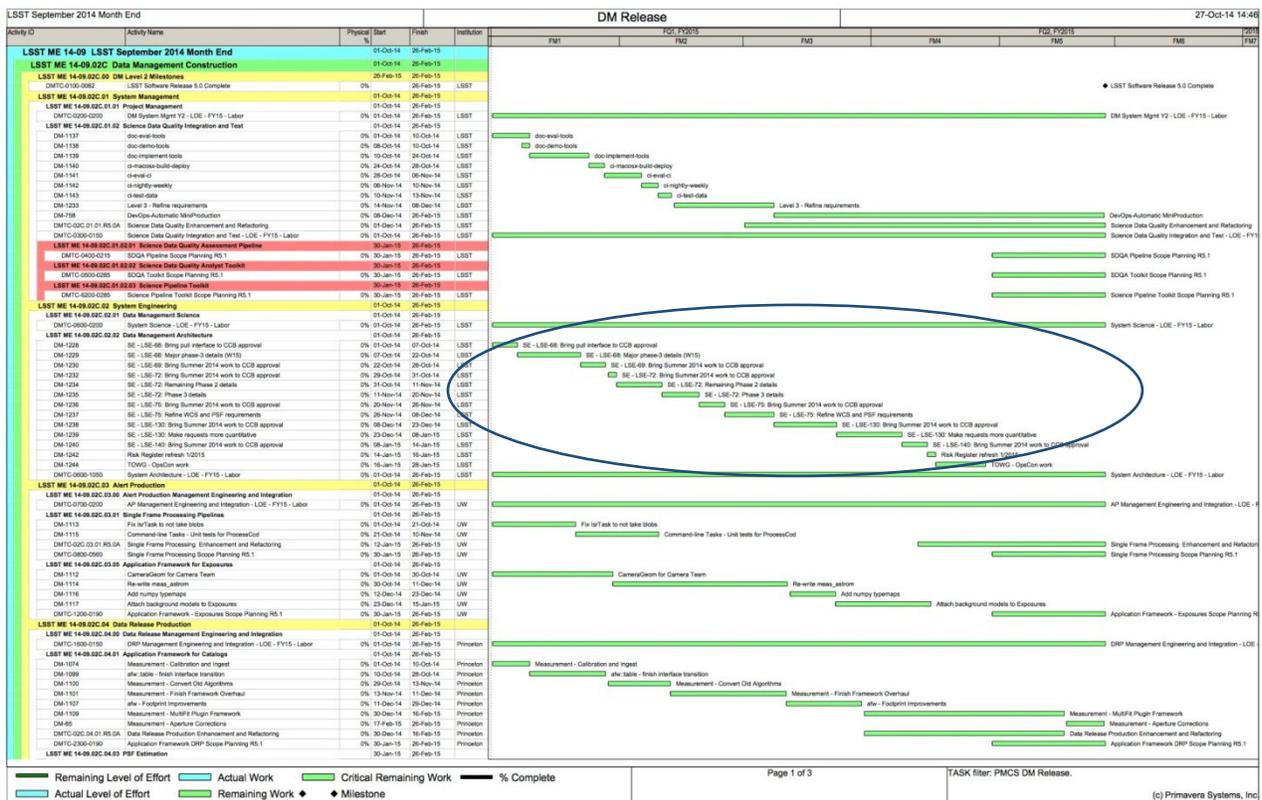


Figure 11 Activities loaded from JIRA Epics

5. PERFORMANCE/FORECAST MEASUREMENT, ANALYSIS

Once the Epics have been loaded into the baseline the team executes the work that has been planned. During the execution phase status is collected from JIRA and loaded into the PMCS. This method of collecting status is very efficient as developers can work in a tool (JIRA) that provides real time interaction and doesn't require learning a new system like Primavera to record progress.

Performance Measurement

On at least a monthly basis (as a minimum) JIRA is updated with the status of the remaining and in-progress activities. This is the status that is loaded into the PMCS which includes:

- Actual start dates for activities begun during the status period (the day recorded in JIRA when the state changes to "In Progress").
- Actual finish dates for activities completed during the status period (the day recorded in JIRA when the state changes to "Done").
- Actual finish dates for milestones accomplished during the status period.
- Physical percent complete of activities by collecting status of the associated stories on the Activity.

All of this information is recorded in JIRA and is used as the primary method for measuring the progress of software development work at LSST.

Every month, the Project Controls Specialist exports plan updates and status information from JIRA into the PMCS. When developers record progress in JIRA they set the status (Done, In Progress, To Do, etc.) on the stories assigned to an Epic. A Story is either Not Done or Done (0% – 100%), there are no partial completions recorded on stories for EV measurement. Each story that is marked complete will contribute to the total % Complete of the Activity. When these Stories and any additional that may be added during a cycle are all marked Done and the status is imported to PMCS, the Epic will show 100% complete. It is likely that a subset of stories may have not yet been planned for an In Progress Epic. In this case, a step is added in Primavera called "Remaining Stories" which is the delta of the estimated stories on the epic and the actual planned stories. This helps prevent case of having negative earned value and new stories are added to an Epic.

The mechanism for exporting status uses the standard Excel interface from JIRA as shown in Figure 12. The story export from JIRA is filtered to contain all issues that are assigned to an Epic and are of a type of Improvement, Story, or Bug. This data is then transferred to an Excel template which uses Vlookup functions to relate Epic and story data from JIRA with cost and schedule data from Primavera. This enables the tracking of performance to baseline dates and dollars solely in Excel.

The mechanism used to import the stories into the IMS employs an internally developed custom tool that allows importing activity steps into Primavera. Without this tool there is unfortunately no Primavera interface to import activity steps and it would be an arduous effort to manually load the steps each month. Once the steps have been imported EV is analyzed in the PMCS, and variances are calculated.

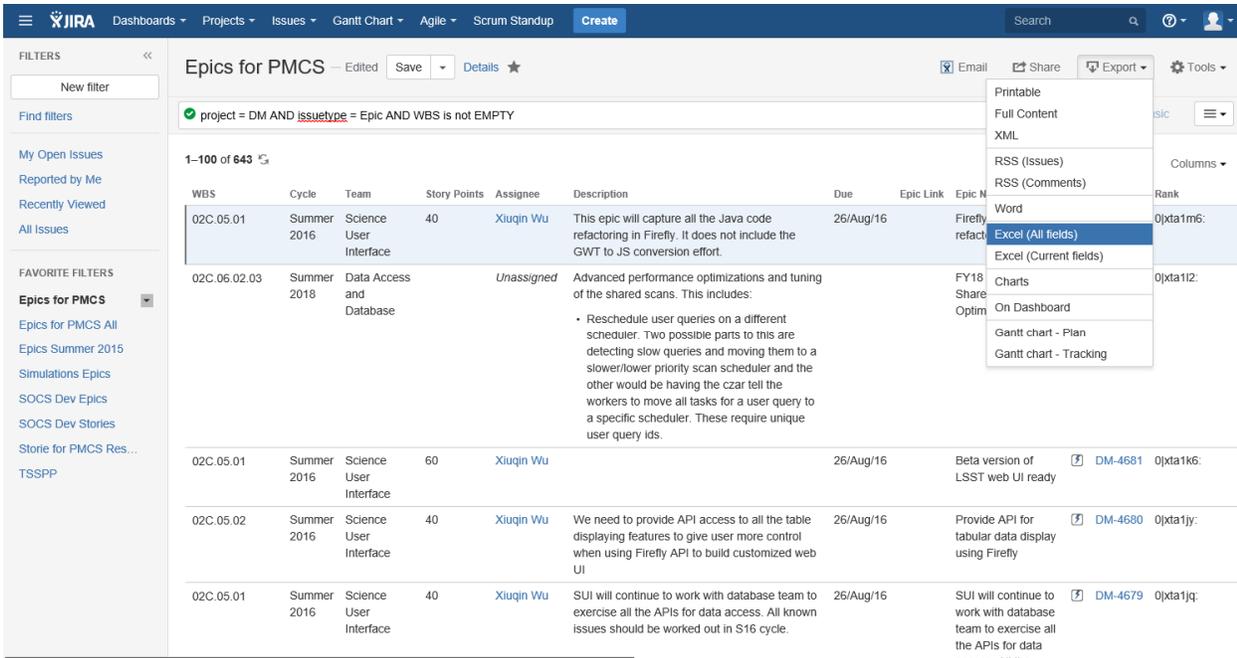


Figure 12 JIRA Interface to Export Status

EV is analyzed in Deltek Cobra, which integrates directly with Primavera. This integration establishes a many to one relationship between activities in the IMS to a single work package in Cobra. The EV for a Work package is calculated in Cobra by determining the earned amount for each activity in relation to its budget and % complete. The sample data in Figure 13 shows the flow from Primavera budget (top) and status to the integration in Cobra

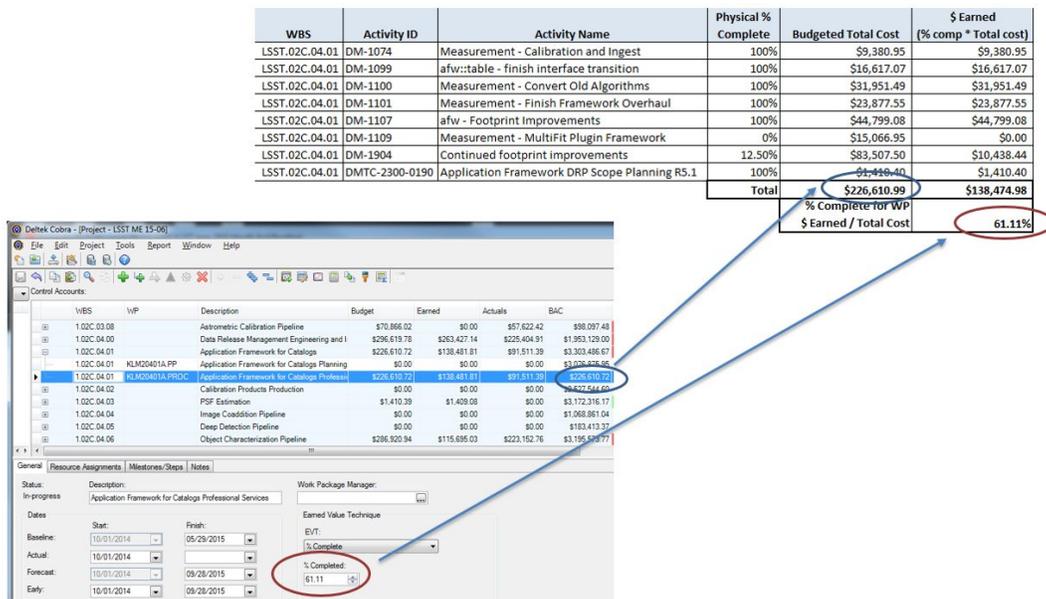


Figure 13 IMS to EVMS tie out

While Earned Value is very effective over the long term it does have some “lags” inherent in the system.

- 1 week to 1-month lag between updates in JIRA and in PMCS
- 1+ month lag between PMCS updates and monthly reports
- EV is “skewed” by late invoicing, requiring estimated actuals

We do have the ability to assess performance earlier outside of the EV tools by looking at trends in JIRA-based on Epics and Stories directly.

- Estimated Story Points vs. Planned Story Points
- Planned Story Points vs. Completed Story Points

6. REPORTING

Performance metrics generated from PMCS and JIRA are used in conjunction to assess performance, and ultimately to report to stakeholders on the health of the project. An orderly process is used to collect, review, report, and use the data generated by the system and is repeated monthly. This monthly reporting cycle is based on the accounting month which ends on the last day of each calendar month. These project status reports contain the following information:

- Budget summary
- CPI/SPI Trending
- Status of key milestones
- Progress narrative
- Baseline change control log
- EVMS data
- Variance explanations (when required)

Electronic systems have been developed at LSST to serve this reporting and analysis data directly from the Cobra and Primavera databases in a system that allows drilling from a high WBS level all the way to the activity and activity step level to facilitate simplified analysis and reporting.

The data generated from the PMCS are available in a web application known as eCAM, or electronic T/CAM Notebook, which is shown in Figure 14. eCAM facilitates this monthly analysis and reporting by showing all current period, cumulative, and at complete EVMS data for all WBS/CA/WPs for the entire project. Status indicators exist in eCAM to not only highlight costs that have tripped variance thresholds, but also date indicator lights to highlight the following situations;

- Red: Today’s date \geq forecast start/finish date and Today’s date \geq BL start/finish date
- Yellow: Forecast start/finish date $>$ BL start /finish date and today’s date $<$ BL Start/ BL Finish
- Green; Forecast start/finish date \leq BL start /finish date and today’s date $<$ BL Start/ BL Finish

eCAM will highlight all control accounts and work packages that have tripped the variance reporting threshold. T/CAMs are directed to populate an Explanation and corrective action at the control account level at a minimum. It is encouraged that this data is captured at the work package level and aggregated to the parent control account. Past narratives are evaluated each month to address issues that aren’t being resolved.

Project: LSST ME 15-06 CAM: Swinbank J WBS Level: All WBS: 1 - LSST May 2015 Month End Baseli Only Show Variances Hide WBS Summary Hide Completed Expand All

Near Term Responsibilities Procurement Milestone

EV Summary SV CV Selected Work Package: KLM20401A.PROC EVT: PctComplete CAM: Swinbank J Narrative Required

WBS / WP	Current Period								Cumulative to Date								At Complete			
	Budget BCWS	Earned BCWP	Actuals ACWP	SV	CV	SPI	CPI	Budget BCWS	Earned BCWP	Actuals ACWP	SV	CV	SPI	CPI	BAC	EAC	% Spent	% Complete		
1	\$70,168	\$35,688	\$123,809	(\$34,479)	(\$88,121)	0.51	0.29	\$782,330	\$511,436	\$503,884	(\$270,894)	\$7,552	0.65	1.01	\$15,362,042	\$15,263,820	3%	3%		
1.02C	\$70,168	\$35,688	\$123,809	(\$34,479)	(\$88,121)	0.51	0.29	\$782,330	\$511,436	\$503,884	(\$270,894)	\$7,552	0.65	1.01	\$15,362,042	\$15,263,820	3%	3%		
1.02C.04	\$70,168	\$35,688	\$123,809	(\$34,479)	(\$88,121)	0.51	0.29	\$782,330	\$511,436	\$503,884	(\$270,894)	\$7,552	0.65	1.01	\$15,362,042	\$15,263,820	3%	3%		
1.02C.04.00	\$50,855	\$21,028	\$47,306	(\$29,827)	(\$26,278)	0.41	0.44	\$296,620	\$263,427	\$225,405	(\$33,193)	\$38,022	0.89	1.17	\$1,953,129	\$1,873,603	12%	13%		
1.02C.04.01	\$0	\$0	\$22,682	\$0	(\$22,682)	0.00	0.00	\$226,611	\$138,482	\$91,511	(\$88,129)	\$46,970	0.61	1.51	\$3,303,487	\$3,243,856	3%	4%		
KLM20401A.PP	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$3,076,876	\$3,076,876	0%	0%		
KLM20401A.PROC	\$0	\$0	\$22,682	\$0	(\$22,682)	0.00	0.00	\$226,611	\$138,482	\$91,511	(\$88,129)	\$46,970	0.61	1.51	\$226,611	\$166,980	40%	61%		
1.02C.04.02	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$2,527,545	\$2,527,545	0%	0%		
1.02C.04.03	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$1,410	\$1,409	\$0	(\$1)	\$1,409	1.00	0.00	\$3,172,316	\$3,170,907	0%	0%		
1.02C.04.04	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$1,068,861	\$1,068,861	0%	0%		
1.02C.04.05	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$0	\$0	\$0	\$0	\$0	0.00	0.00	\$183,413	\$183,413	0%	0%		
1.02C.04.06	\$19,313	\$14,660	\$53,821	(\$4,653)	(\$39,161)	0.76	0.27	\$257,690	\$108,118	\$186,967	(\$149,572)	(\$78,849)	0.42	0.58	\$3,153,291	\$3,195,635	6%	3%		

Schedule Details Selected Work Package: KLM20401A.PROC

ID	Resource Description	Baseline Start	Baseline Finish	Forecast/Actual Start	Forecast/Actual Finish	% Comp.	Total Float	Planned Cost	Planned Hours	Target Duration	Remaining Duration
LSSE-PRINCETON - Sr. Software Engineer-PRINCETON	DM Sr. Software Engineer	11-Dec-14	29-Dec-14	20-Jan-15 A	31-Jan-15 A	100.0	0	\$44,799	166	11	11
DM-1107 - afw Footprint Improvements	DM Scientist						0				
LSC-PRINCETON - Scientist-PRINCETON	DM Sr. Scientist						0				
LSC-PRINCETON - Scientist-PRINCETON	DM Sr. Scientist						0				
LSSE-PRINCETON - Sr. Software Engineer-PRINCETON	DM Sr. Software Engineer						0				
DM-1109 - Measurement - MultiFit Plugin Framework	DM Sr. Software Engineer	30-Dec-14	16-Feb-15	01-Jul-15	14-Aug-15	0.0	-19	\$15,067	126	33	33
LSSE-UCD - Software Engineer - UCD	DM Software Engineer						0				
LSSE-PRINCETON - Sr. Software Engineer-PRINCETON	DM Sr. Software Engineer						0				
DM-1904 - Continued footprint improvements	DM Sr. Software Engineer	02-Mar-15	29-May-15	24-Mar-15 A	28-Sep-15	12.5	-19	\$83,508	296	64	63
LSC-PRINCETON - Scientist-PRINCETON	DM Scientist						0				
LSC-PRINCETON - Scientist-PRINCETON	DM Sr. Scientist						0				
DMTC-2300-0190 - Application Framework DRP Scope	DM Sr. Scientist	30-Jan-15	26-Feb-15	02-Feb-15 A	27-Feb-15 A	100.0	0	\$1,410	5	20	20
Total:							\$2,048	138,474			

Figure 14 eCAM showing JIRA Epics assigned to a Work Package

Outside of eCAM, JIRA performance can be measured and reported directly from the JIRA excel exports, as shown in Figure 15. By pivoting this data information can be grouped by any of the code fields used in JIRA and Primavera.

WBS	CAM	Key	Epic Name	COMPLETED	Stories Planned	Stories Complete	Remaining SP	Sum of SP % Variance Completed / Target Planned 20/50
02C.07.03	Gelman M	DM-1273	Plan OpenStack v1 Deployment	To Do	67	47	20	30%
02C.07.03 Total					67	47	20	30%
02C.07.03.08	Gelman M	DM-2239	Develop use cases for TOWG	To Do	9	1.5	7	82%
02C.07.03.08 Total					9	1.5	7	82%
02C.07.04	#N/A	DM-2225	LOE - S15 (sys admin)	In Progress	178	178	0	0%
02C.07.04 Total					178	178	0	0%
02C.07.04.05	Gelman M	DM-1268	Upgrade Rack KVMs	In Progress	11	6	5	45%
		DM-2211	Setup qserv prototype for qserv team	To Do	4	4	0	0%
		DM-2218	Base Data Center Requirements	Done	15	13.5	1	7%
02C.07.04.05 Total					30	23.5	6	24%
02C.07.04.06	Gelman M	DM-2224	Wide-Area Network Work	To Do	15	13	2	13%
02C.07.04.06 Total					15	13	2	13%
Grand Total					2491	1685.5	806	37%

Figure 15 JIRA Excel Report

Typical Analysis:

- We consider overall work rate to be relatively constant over a 6-month cycle, since we do not have large staffing changes during the cycle
- Progress should be roughly linear with time (but see reporting comment below)
- The SP Variance % of Completed/Planned is 37% which is ~4% slow versus expectation of 33% at 2/3 date, i.e. we are behind by 4% of total planned work
- Our experience is that we under-report progress until the last month, due to “conservative” assessments of completion, and having only done unit tests which don’t expose all problems
- We will both add SP in the last 2 months and complete them at a higher rate
- Based on the above, I expect us to slip 15% of S15 SP

7. CONCLUSION AND NEXT STEPS

The LSST EVMS and Agile integration provide a means to develop software using current best practice Agile methods while preserving the planning, performance monitoring/analysis, and reporting rigor required by EVM. By allowing the near-term rolling wave plan to be developed and executed in an Agile manner, but within the context of the overall EV process, a marriage of the two worlds is achieved.

While the process is now fully integrated, there are a number of steps where the exchange of data between the various tools in the EVMS is still somewhat manual, via data export and import, and a number of areas where additional reports and notifications would be useful. Future work in the system will be directed at improving the level of automation and support for additional analytical reports and notification.

8. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation through Cooperative Support Agreement (CSA) Award No. AST-1227061 under Governing Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

We thank Victor Krabbendam and Tim Jenness, AURA/LSST for reviewing the draft manuscript.

9. REFERENCES

- [1] Kahn, S., "Final design of the Large Synoptic Survey Telescope," in [Ground-Based and Airborne Telescopes IV], Hall, H. J., Gilmozzi, R., and Marshall, H. K., eds., Proc. SPIE 9906, in press (2016).
- [2] Kurita, N., Kahn, S., Stubbs, C., Ritz, S., Nordby, M., Riot, V., "Large Synoptic Survey Telescope camera design and construction", in [Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation], 2016, 9912, in press
- [3] Juric, M. et al., "The LSST Data Management System," in [Astronomical Data Analysis Software & Systems XXV], Lorente, N. P. F. and Shortridge, K., eds., ASP Conf. Ser. in press, arXiv:1512.07914, ASP, San Francisco (2016)
- [4] Boehm B., Turner, R. , "Management challenges to implementing agile processes in traditional development organizations", IEEE Software Volume:22 Issue:5 (2005).
- [5] Becla, J., et al, "LSST DM Project Management Tools Overview", [LDM-472](#)
- [6] Kantor, J., et al, "Data Management Releases", [LDM-240](#)