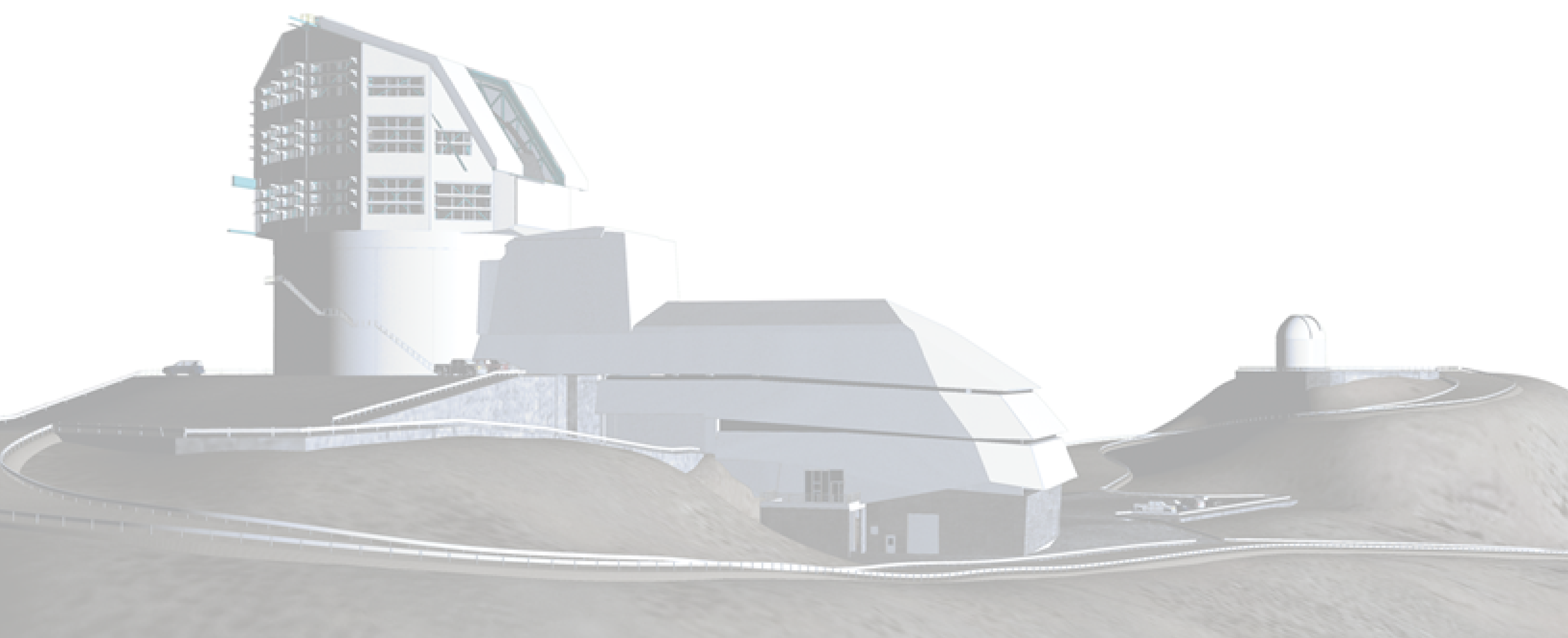
Vera C. Rubin Observatory
Data Management

Rubin Operations Work Management and Budget Planning

William O'Mullane, Amanda Bauer, Robert Blum, Phil Marshall, Cathy Petry, Ranpal Gill

RTN-005

Latest Revision: 2023-02-17



Abstract

This document describes the budget and work planning process, and provided guidelines for the management of effort, in Vera C. Rubin Observatory Operations. It lays out the annual cycle of milestone setting, development of departmental budget guidance, ground-up spending plan “sandboxing,” and work planning, and introduces (in an appendix) the suite of Rubin Operations Planning Tools. It describes the process for planning and executing agile-based work, including tasks that are carried out with regularity (nightly or monthly, etc), long-term development work that iteratively incorporates user feedback, and level of effort activity. The process for measuring progress towards an annually-planned schedule is described, as well as the budget planning cycle and the Rubin planning tools used.

Change Record

Version	Date	Description	Owner name
1	2021-11-05	Initial version.	William O'Mullane
1	2023-02-17	Extension to include budget planning, FY23 process updates.	Phil Marshall & Cathy Petry

Document source location: `https://github.com/rubin-observatory/rtn-005`

Contents

Rubin Operations Work Management and Budget Planning

1 Introduction

This document provides a guide to the Vera C. Rubin Observatory approach to work management and annual planning. See the operations proposal Rubin Directors Office (RDO)-018 for a description of the full scope and high level goals of the program. There is no formal Earned Value Management System (EVMS) required from the funding Agencies (the National Science Foundation (National Science Foundation (NSF)) and the Department of Energy (DOE) Office of Science) so activities are planned in detail on a semi-annual basis and effort towards that schedule of activities is tracked through an agile process.

The annual planning process starts by reviewing, revising, and adding major milestones to the next fiscal year plan. These are centered around releasing data to the public and major maintenance to the telescope system once we enter the phase of full survey operations, and it's these major milestones that are tracked and reported on in the annual NSF's National Optical-Infrared Astronomy Research Laboratory; `https://nationalastro.org` (NOIRLab) Program Operations Plan (Project Operating Plan (POP)) and SLAC National Accelerator Laboratory (SLAC) Field Work Proposal (FWP). Then, on a six-month cycle, the Leadership Team builds a series of "epic" milestone activities that are discrete pieces of work within Departments and Teams to collectively deliver the high level milestones. Teams record their day to day work in JIRA and overall progress is monitored automatically through Smartsheet and reported to the Agencies through our managing organizations.

This framework allows the multidisciplinary Rubin teams to operate the facility and generate nightly data products while continuously improving efficiency of workflows, as well as iteratively responding to user community feedback on a longer timescale to maximize the scientific benefit of annual data releases. Examples include optimizing the observing strategy as the survey progresses, improving algorithms in response to the user community feedback, and other incremental work needed to produce the annual data releases.

In this document, we lay out the procedural details for how we define and carry out annual plans, effectively track work progress to ensure delivery of milestones, maintain visibility in our workflows, remain responsive to change, and offer staff the ability to innovate and collaborate.

2 Organizational Structure

2.1 Rubin Operations Leadership

Rubin Observatory is a Program of NSF's NOIRLab. The Rubin Observatory Director is Robert Blum, the Rubin Deputy Director for NOIRLab is currently under recruitment, and the Rubin Deputy Director for SLAC is Phil Marshall. They are the first point of contact for all issues regarding project management within Rubin Observatory Operations.

The Head of Operations is Ranpal Gill and the Program Coordinator is Cathy Petry. They monitor the budget and maintain details within the NOIRLab accounting system. They assist in developing the annual Program Operating Plan (POP), tracking milestones and reporting on progress.

On the SLAC side, Christine Soldahl is the Business Manager who handles similar tasks.

Rubin Operations has four operational Departments in addition to the RDO (Rubin Director's Office): Rubin Observatory Operations (ROO) (Rubin Observatory Operations), Rubin Data Production(Obsolete use Rubin Data Management (RDM)) (RDP) (Rubin Data Production), Rubin system PerFormance (RPF) (Rubin System Performance), and REO (Rubin Education and Public Outreach). Each operational Department is led by an Associate Director (Associate Director (AD)).

2.2 Annual Reporting

The POP is a defined process for NOIRLab, where annually the next fiscal year's POP is developed and reporting on progress of the current year's POP to the NSF is done quarterly with a final annual progress report. Rubin Operations considers the POP to be a Rubin activity, which informs both NOIRLab and SLAC leadership of the annual Rubin activity including milestones and budget. For NOIRLab, the Rubin POP is integrated and delivered to NSF for the next fiscal year at the end of the current fiscal year. For SLAC, the POP informs SLAC's annual planning, which culminates in a Field Work Proposal (Field Work Proposals (FWP)) for all SLAC High Energy Physics activity including Rubin.

The FWP is delivered in June of the current fiscal year, and covers the federal budget request for the next two fiscal years. The FWP is previewed to SLAC management and Department of Energy (DOE) in February in advance of the final delivery in June. Because the POP for NSF lags the FWP for DOE by one year, and their submission dates differ by several months, Rubin does high level planning in early Quarter two (Q2) of the fiscal year (January and February) and plans 2 years ahead, as required for the SLAC FWP. Detailed activity planning, including defining smaller chunks of work as lower level milestones, continues though the year in advance of the next year. This detailed activity planning is the subject of this document.

2.3 Work Breakdown Structure

The Work Breakdown Structure (WBS) is a hierarchical description of Rubin Operations from an activity-based perspective. It provides a useful structure to organize Rubin Operations and plan annual work around. Rubin is level 1 of the WBS, the departments are level 2, and teams within the departments are level 3 and in the case of Program Operations, level 4. Individual roles in operations are defined at the lowest level.

This table shows the level 2 and level 3 elements of the Rubin Operations work breakdown structure.

L2 WBS	Lens 3 (L3) WBS	Description
1		Rubin Director's Office
	1.1	Director's Office
	1.2	Safety
	1.3	Program Operations[1]
	1.4	In-Kind Program Coordination
	1.8	Legacy Survey of Space and Time
	1.10	Sustainability
	1.11	Site Protection
	1.12	Rubin Site Protection
2		Rubin Observatory Operations
	2.1	Observatory Operations Management
	2.2	Observatory Science Operations
	2.3	Observatory Software
	2.4	Summit Operations
	2.5	Nighttime Operations
	2.6	Engineering

L2 WBS	Lens 3 (L3) WBS	Description
3		Rubin Data Production
	3.1	Data Production Management
	3.2	Infrastructure and Support
	3.3	Data and Processing Architecture
	3.5	Algorithms and Pipelines
	3.6	Service Quality and Reliability Engineering
	3.7	DevOps Support
	3.8	Data Security
4		Rubin System Performance
	4.1	System Performance Management
	4.2	Verification and Validation
	4.3	Community Engagement
	4.4	Survey Scheduling
	4.5	Systems Engineering
5		Rubin Education and Public Outreach
	5.1	Education and Public Outreach (EPO) Management
	5.2	EPO Technical
	5.3	Education
	5.4	Outreach

Detailed work is planned in advance of each fiscal year at the team level. Team leads will work with department associate directors to develop plans for activities that address specific milestones, projects, and level of effort activity. Progress towards the highest level milestones is reported regularly throughout the fiscal year to SLAC, Association of Universities for Research in Astronomy (AURA), NSF and DOE.

2.4 Activity Types

There are two types of activities (or epics) that are planned: activities that result in a deliverable, and level of effort (or support) activities. Progress can be tracked on activities with a deliverable by computing the fraction of the effort that is complete (percent complete) based

[1] Program Operations is made up of several groups at level 4 that are not presented here but are available for activity planning and budgeting.

on completed stories linked to the epic. Progress on Level of Effort (LOE) work is assumed to progress proportionally with the passage of time.

LOE activities include attending meetings, reporting on milestones, or taking part in other activities which do not directly map to a deliverable or product. This may be particularly the case for technical managers or others in leadership roles. In general, we strive to minimize the fraction of effort which is devoted to LOE activities and favor those which are more directly accountable. However, in certain cases such as operations and maintenance of telescope and facility systems, pipelines or other systems, LOE is perfectly acceptable. Technical staff in Chile at the summit facility may spend a much more significant fraction of time as LOE. As an example, a first-order estimate is that developers will spent 30% of their time on LOE type activities, and the remaining 70% of their effort is planned and tracked against well-defined deliverables.

3 Estimating Effort

3.1 Basic Assumptions

Rubin Operations assumes that a full-time individual works for a total of 1,800 hours per year: this figure is *after* all vacations, sick leave, etc are taken into account. The Rubin Operations partners, SLAC and NOIRLab, may have different definitions for tracking their staff time; Rubin Operations uses 1,800 hours per year as a fiducial value for effort estimation purposes.

In general, staff in Rubin Operations roles at a given expected full-time equivalent (Full-Time Equivalent (FTE)) effort level are expected to devote that fraction of their total work time to Rubin Observatory .

Staff in “scientist” or “engineer” roles can allocate up to 2% of their time to training, 2% to administrative activities and 1% to outreach and Diversity, Equity, and Inclusion (DEI) activities.

Staff in “scientist” roles are expected to spend 20% of their time on personal research (see the Rubin Operations Plan for details). That is, scientists are expected to devote 1,440 hours per year to operations activity, and the remainder of their time to personal research.

Personal research time is charged to a NOIRLab’s Research and Science Services (square root of the sum of the squares (RSS)) project code and is prorated for staff who are fractionally allocated to Rubin Operations. Training, administrative, and outreach and DEI time is charged to either RSS or Early Science (ES) depending on where staff will move into NOIRLab (RSS or ES). Functional Managers will ensure the proper project codes are available on your timecard.

Rubin expects to pay the full rate for any scientist or engineer who contributes full-time or fractionally to operations. This is handled through indirect rates at NOIRLab and direct charges to research accounts at SLAC. Science time is included in the subcontracts of our partners at affiliated institutions through indirect charges similar to the case for NOIRLab.

In Data Production, the base assumption is that 30% of an individual's Rubin Observatory operations time (i.e. 540 hours/year for a full-time developer, 432 hours/year for a full-time scientist) are devoted to overhead for regular meetings[1], ad-hoc discussions and other interruptions. This work is counted as LOE. It is actively encouraged to allocate less than 30% of an individuals time to LOE where that is possible.

Assuming no variation throughout the year, we therefore expect 105 hours of productive work from a developer, or 84 hours from a scientist, per month. Note that this is averaged across the year: some months, such as those containing major holidays, will naturally involve less working time than others: the remainder will necessarily include more working time to compensate. For other staff, the LOE will be higher but include much more day to day activity than for the developer case.

Rather than working in hours, our JIRA based system uses Story Points (SP), with one SP being defined as equivalent to four hours of effort (half a day's work) by a competent developer.

Thus, we expect developers and scientists to produce 26.25 and 21 SPs per *average* month respectively.

3.2 Special Cases

[1]"Meetings" include, for example, scheduled weekly team meetings, stand-ups, etc; major conferences or project meetings involving preparation, travel time, etc should be scheduled in advance and allocated System PerFormances (SPs).

	Hours		SPs
	Per year	Per month	Per month
Full-time Developer	1800	105	26.25
Full-time Scientist	1440	84	21.00

TABLE 3: Expected working rates for developers and scientists. Technicians and engineers follow the same rates as developers.

3.2.1 Newcomers

New or inexperienced developers, even when devoting their full attention to story-pointed work, will likely be less productive than their more experienced peers. In this case, the ratio of hours to SPs increases, but the number of hours remains constant.

Note that specific activities related to "onboarding" and getting up to speed with operations can be ticketed as regular work. For example, working through tutorials, reading documentation, and so on are all activities which can earn SPs.

3.2.2 Team Leads and other Leadership Roles

Individuals in leadership roles may find it necessary to assign a larger fraction of their time to LOE type work, and therefore spend fewer hours generating SPs. The ratio of hours to SPs remains constant, but the number of hours decreases.

4 Long Term Planning

4.1 Timeline for Annual Planning

Our basic annual timeline is laid out below; for more detail on the budget planning and sandboxing, see Section 8.

- December-March: accumulate top-down input for next year budget and schedule.
  - Update lab-wide budget models (Resource Allocation Sheet (RAS)) from Staffing and Nonlabor plans, for quarterly forecasting (December, March).

- Update budget tools with any major changes, for AURA (federally) Negotiated Indirect Cost Rate Agreement (NICRA) proposal. Incorporate any changes in SLAC overhead rates and escalations.
- Adjust major milestone dates as events unfold.

- March
  - Semi-annual process of planning epics and L3s for next six months
- April-May: NOIRLab POP process and SLAC FWP development kick off mid-April, with major milestones and budget due end of May.
  - Early April: set next Financial Year (FY) major milestones
    - ADs review and/or add/edit major milestones
    - Discuss in Ops Exec
    - Implement changes in Smartsheet and JIRA
  - Late April: set next FY (and FY+2) budget:
    - Early April: freeze planning tools (WBS and Cost Calculator)
    - Joint agency operations status review (Joint Operations status Review (JOR))
    - RDO issues departmental budget guidance based on costed and reviewed plan.
  - May: next FY Sandboxing:
    - Early May: Prepare and issue sandbox workbooks, one per department.
    - Early May Department management teams facilitate team-level sandbox pitches (including Lens 2 (L2)/L3 milestons, changes to spending plans)
    - Late May: Sandboxing Workshop, for all ops teams and Director's Office.
- June-July: implement sandboxed changes.
  - Implement proposed changes to Staffing Plan, Risk Register, JIRA/Smartsheet L2/L3 milestones
  - End of June: Final Labor and Nonlabor budget for NOIRLab POP
  - Update lab-wide budget models (RAS) from Staffing and Nonlabor plans, for quarterly forecasting.
  - July: derive next FY contract and Memorandum Purchase Order (MPO) SOWs.
- September

- Update lab-wide budget models (RAS) from Staffing and Nonlabor plans, for quarterly forecasting.
- Semi-annual process of planning epics and L3s for next six months
- Planning for 2nd half may trigger major milestone changes needed in next FY:
  * Collect requests, study feasibility in Smartsheet
  * Discuss at Ops-Exec, import changes into JIRA and Smartsheet

4.2 Components of Annual Planning

The authoritative, high-level summary of the long-term planning system may be found in any POP process document.

Here we expand upon the details of that system. The plan for Pre-Operations and Full Survey Operations is embodied in:

1. A set of *milestones*, each of which represents the delivery of a major aspect of Rubin Operations, availability of specific functionality, or maintenance event for the telescope system. Milestones are planned in Smartsheet and then officially defined in a JIRA milestone issue.
2. A series of *epics* describe major pieces of work. Epics are associated with concrete, albeit high-level, deliverables or outcomes that culminate in the achievement of the above milestones, and have specific resource loads (staff assignments story point values) and end dates. All epics are linked to the milestone they are created to help deliver, although some epics might exist without linking to a milestone (level of effort or emergent work epics, for example).
3. A visualization of progress on work done towards achieving milestones is captured in Smartsheet, which directly tracks progress by rolling up issues that are completed inside of JIRA epics that work together to deliver a given milestone.

Milestones are allocated to one of three levels, defined as follows:

Level 1 These are at the full observatory level and are owned by the Directors Office. Examples are the completion of a Data Preview, the beginning of nightly observations for the full survey, or the delivery of an annual Data Release. Level 1 milestones are achieved by the culmination of effort defined by a set of Level 2 and Level 3 milestones. Level 1 milestones *may* be reported to the agencies as defined by the annual POP .

Level 2 These reflect effort within a Department and are owned by an Associate Director, or are cross-Department commitments. As such, they must be defined in consultation with the Director's Office. Level 2 milestones are achieved by the culmination of effort defined by a set of Level 3 milestones. Some Level 2 milestones *may* be reported to the agencies as defined by the annual POP.

Level 3 These are internal to a particular Department and assigned to a team and can therefore be specified by a single team lead.

Some of these milestones are exposed to external reviewers; it is important that these be delivered on time and to specification. Level 1 and 2 milestones are under change control once they are defined and described in a JIRA Milestone issue. Note the change control process is under development as a Pre-Operations activity.

Level 3 milestones are defined for use within Departments and not required to go under project change control, but properly adhering to the plan is important: your colleagues in other teams will use these milestones to align their schedules with yours, so they rely on you to be accurate.

Epics should work to achieve milestones, i.e., they may be blocking issues on the milestones. When a detailed description of work for a given epic is known, it is described in JIRA. It should then be assigned to the appropriate cycles.

Progress is tracked toward achieving milestones in Smartsheet by monitoring completed story points on linked issues in JIRA epics and rolling up the total progress. To ensure success, JIRA epics must be completely detailed out prior to a full 6-month cycle and total effort should be estimated out for an entire fiscal year of effort, as detailed below. All milestones should appear in JIRA with a milestone issue type as the source of truth.

4.3 Planning Research Work

In order for Rubin Observatory to reach its science goals, new algorithmic or engineering approaches must sometimes be researched. It is appropriate to budget time for this research work in planning packages.

4.4 Epic-Based Long Term Plans

As long as they have not been scheduled for the current cycle, these epics can be freely created and changed at any time, without any sort of approval process.

Fine grained planning of this sort can be useful for "bottom-up" analysis of the work to be performed and validation of the resources needed to implement a particular planning package. Thinking through the plan in this way can help in building up a detailed plan in a flexible, agile way, while also ensuring that scope, cost and schedule are carefully controlled.

4.5 Defining the Schedule with Milestones

Rubin Milestones are defined as JIRA issues of type "milestone". As indicated above, the Director (or their designate) defines the L1 milestones, the Associate Directors (ADs) define their departments' L2 milestones, and the Team Leaders and ADs define the L3 milestones for their teams.

L1, L2 and most L3 milestones are defined as part of the annual planning cycle, and prior to the year in which the work associated with them is due to be carried out. ADs and Team Leaders communicate their milestones to the Program Coordinator, who enters them into Smartsheet and then creates a JIRA issue of type "milestone" for each one.

During the year, it is sometimes necessary to create new milestones (primarily at Level 3) that were omitted during the earlier planning phase. In this case, the team leader or AD may create the JIRA milestone directly, and alert the Program Coordinator to it for inclusion in the Smartsheet.

The following JIRA fields must be filled out when defining a new milestone:

- The `Milestone Level`, "1," "2," or "3."
- The `Summary` is also the milestone title, and is a concise description of when the milestone is reached. Example: *"V5 WBS workbook and Preliminary Cost Calculator implementation complete."*
- The `Milestone Activity` field is equivalent to the activity that will take place in order to produce the deliverable and meet the milestone. It should contain a sentence outlining the activity to be performed in order for the milestone to be reached and the deliverable to be produced. In the POP document tables this is the "Activity." Examples include *"Deliver Data Preview 0.1 (DP0.1)"* (an L1 milestone) and *"DP0.1 Data Release: science-ready catalogs released from the Interim Data Facility (IDF)"* (an L2 milestone that belongs to it).
- The `Deliverable` is a very terse list of the deliverables needed to reach the milestone. Example: *"V5 WBS workbook and Preliminary Cost Calculator."*
- The `Description` text should contain more information detailing the scope of activity needed to complete the milestone. Example: *"Upgrade the WBS activity, labor and non-labor plans from V4 to V5 in order to capture a United States (US) Data Facility (DF) at SLAC, a United Kingdom (UK) DF, and any other modifications needed, and estimate the corresponding budget."* Note that while only a subset of L1 and L2 milestones are actually listed in the NOIRLab Program Operations Plan (POP), Rubin adopts the same structure for all its milestones.
- The `Due Date` is the latest date in the future by which the milestone needs to be reached. This date should be before or the same as the milestone's parent milestone's "Milestone Due Date" as shown in the Smartsheet, or the "Due Date" of the parent milestone in JIRA.
- In the `Linked Issues` field, create a *"blocks"* link to the parent milestone to reflect how that milestone contributes to achieving that milestone.

The Program Coordinators will ensure that the milestones that have been defined are correctly arranged in the Smartsheet, so that their epics appear nested beneath them.

5 Short Term Work Planning

Short term planning is carried out in blocks referred to as cycles, which (usually) last for six months. Before the start of a cycle, milestones are confirmed by the Director's Office, listed in

Smartsheet, and detailed in the Milestone issue. Any team member can find the milestones in JIRA.

5.1 Defining The Plan

5.1.1 Scoping Work

The first essential step of developing the short term plan is to produce an outline of the program of work to be executed. In general, this should flow directly from the long term plan (§4), ensuring that the expected planning packages are being worked on and milestones being hit.

While developing the cycle, please:

- Do not add *artificial* padding or buffers to make the schedule look good;
- Do budget appropriate time for handling bugs and emergent issues;
- Reserve time for planning the following cycle: it will have to be defined before this cycle is complete;
- Leave time for other necessary activities, such as cross-team collaboration meetings and writing documentation.
- Per the cycle cadence, ensure that new development will conclude (or, at a minimum, be in a releasable state) in time for the end of cycle release.

Obviously, ensure that the program of work being developed is achievable by your team in the time available: ultimately, you will want to compare the number of SPs your team is able to deliver (§3) with the sum of the SPs in the epics you have scheduled (§5.1.2), while also considering the skills and availability of your team. It is better to under-commit and over-deliver than vice-versa, but, ideally, aim to estimate accurately.

5.1.2 Defining Epics

The plan for a six month cycle fundamentally consists of a set of resource loaded epics defined in JIRA. Each epic loaded into the plan must have this minimum set of fields filled in:

- A concrete, well defined deliverable *or* be clearly described as a "bucket" or "emergent work" (§5.1.4);
- The `Component` field set to the appropriate Department;
- The `Story Points` field set to a (non-zero) estimate of the effort required to complete the epic in terms of SPs (see §3).
- In the `Linked Issues` field, create a *"blocks"* link to the parent milestone to reflect how that milestone contributes to achieving that milestone.
- The `Due Date` field set to the appropriate date, which does not exceed the due date of the milestone it is labeled to achieve.
- The `label` field is set to identify the fiscal year during which the work will be done. Examples are Financial Year 23 (FY23) or Financial Year 24 (FY24).

The fields above are required to have values entered because they define the connection to Smartsheet where effort-tracking for the full project is done. Other fields in the epic can also be filled in as needed.

Be aware that:

- An epic may only be assigned to a single cycle. It is not possible to define an epic that crosses the cycle boundary (see §5.2 for the procedure when an epic is not complete by the end of the cycle).
- Indeed, where possible management activities *should* be scheduled as epics with concrete deliverables in this element rather than being handled as LOE.
- The epic should be at an appropriate level of granularity. While short epics (a few SPs) may be suitable for some activities, in general epics will describe a few months of time. Epics allocated multiple hundreds of story points are likely too broad to be accurately estimated.

Although it is possible—indeed, encouraged—to set the `assignee` field in JIRA to the individual who is expected to carry out the bulk of the work in an epic, this does not provide sufficient granularity for those cases when more than one person will be contributing.

5.1.3 Scheduling Research Work

As discussed in §4.3, research is sometimes required to meet our objectives. However, it is not a natural fit to our usual planning process, as it is speculative in its nature: it is often impossible to produce a series of logical steps that will lead to the required result. We acknowledge, therefore, that scheduling an epic to deliver some particular new algorithm based on the results of research is impossible: we cannot predict with any confidence when the breakthrough will occur.

We therefore schedule research in timeboxed epics: we allocate a certain amount of time based on the resources available, rather than on an estimate of time to completion. However, note that these timeboxed epics should still provide concrete deliverables: they are not open-ended "buckets" as discussed elsewhere.

5.1.4 Bucket Epics

Some work is "emergent": we can predict in advance that it will be necessary, but we cannot predict exactly what form it will take. The typical example of this is fixing bugs: we can reasonably assume that bugs will be discovered in the codebase and will need to be addressed, but we cannot predict in advance what those bugs will be.

This can be included in the schedule by defining a "bucket" epic in which stories can be created when necessary during the course of a cycle. Make clear in the description of the epic that this is its intended purpose: every epic should either have a concrete deliverable or be a bucket.

Bucket epics have some similarities with LOE work. As such, we acknowledge that they are necessary, but seek to minimize the fraction of our resources assigned to them. If more than a relatively small fraction of the work for a cycle is assigned to bucket epics, please consider whether this is really necessary and appropriate.

5.2 Closing the Cycle

Assuming everything has gone to plan, by the end of a cycle all deliverables should be verified and the corresponding epics should be marked as done. Marking an epic as done asserts that the concrete deliverable associated with the epic has been provided.

Epics which are in progress at the end of the cycle cannot be closed until they have been completed. These epics will spill over into the subsequent cycle. It is *not* appropriate to close an in-progress epic with a concrete deliverable until that deliverable has been achieved: instead, a variance will be shown until the epic can be closed. Obviously, this will impact the labor available for other activities in the next cycle. (This does not apply to bucket epics (§5.1.4), which are, by their nature, timeboxed within the cycle).

Be aware that if a planned epic is not closed it may impact the completion of the milestone it contributes to. Epics related to milesteones must be completed in order for the milestone to also be completed.

6 Execution

Having defined the plan for a cycle following §5, we (RDP and RPF) execute it by means of a series of month-long sprints. In this section, we detail the procedures teams are expected to follow during the cycle.

6.1 Detailing Work

6.1.1 Issue Types

There are two JIRA issue types that are used for planning work on epics: Story and Bug

6.1.2 Defining Stories

Epics have already been defined as part of the cycle plan (see §5.1.2). However, the epic is not at an appropriate level for scheduling day-to-day work. Rather, each epic is broken down into a series of self-contained "stories". A story describes a planned activity worth between a small fraction of a SP and several SPs (more than about 10 is likely an indication that the story has not been sufficiently refined). It must be possible to schedule a story within a single sprint, so no story should ever be allocated more than 26 SPs.

The process for breaking epics down into stories is not mandated. In some circumstances, it may be appropriate for the technical manager to provide a breakdown; in others, they may request input from the developer who is actually going to be doing the work, or even hold a brainstorming session involving the wider team. This is a management decision.

It is not required to break all epics down into stories before the cycle begins: it may be more appropriate to first schedule a few exploratory stories and use them to inform the development of the rest of the epic. However, do break epics down to describe the stories which will be worked in an upcoming sprint (§6.2) before the sprint starts. When doing so, you may wish to leave some spare time to handle emergent work (discussed in §5.1.4). Every epic should contain *at least* one story with non-zero story points assigned.

Note that there is no relationship enforced between the SP total estimated for the epic and the sum of the SPs of its constituent stories. It is therefore possible to over- or under-load an epic. This will have obvious ramifications for the schedule. After execution is complete, comparing the total number of SP on planned stories in an epic to the number of SP on the epic itself affords the opportunity to refine time estimates going forward.

6.1.3 Receiving Bug Reports

Members of the project who have access to JIRA may report bugs or make feature requests directly using JIRA. As discussed in §6.4, technical managers should regularly monitor JIRA for relevant tickets and ensure they are handled appropriately.

Our code repositories are exposed to the world in general through GitHub. Each repository on GitHub has a bug tracker associated with it. Members of the public may report issues or make requests on the GitHub trackers. Per the Developer Workflow, all new work must be associated with a JIRA ticket number before it can be committed to the repository. It is therefore the responsibility of technical managers to file a JIRA ticket corresponding to the GitHub ticket, to keep them synchronized with relevant information, and to ensure that the GitHub ticket is closed when the issue is resolved in JIRA.

The GitHub issue trackers are, in some sense, not a core part of our workflow, but they are fundamental to community expectations of how they can interact with the project. Ensure that issues reported on GitHub are serviced promptly.

In some cases, the technical manager responsible for a given repository is obvious, and they can be expected to take the lead on handling tickets. Often, this is not the case: repositories regularly span team boundaries. Work together to ensure that all tickets are handled.

6.1.4 Emergent Work

On occasion, work arises that is not anticipated and therefore not planned. Epics for this type of work will have been set up every cycle so stories should be linked there.

6.2 Sprinting

Each team organizes its work around periods of work called sprints. A sprint comprises a defined collection of stories which will be addressed over the course of the month. These stories are not necessarily (indeed, not generally) all drawn from the same epic: rather, while epics divide the cycle along logical grounds, sprints divide it along the time axes.

Broadly, executing a sprint falls into three stages:

1. Preparation.

   The team assigns the work that will be addressed during the sprint by choosing from the pre-defined stories (§6.1.2). Each team member should be assigned a plausible amount of work, based on the per-story SP estimates and the likely working rate of the developer (see §3).

   The process by which work is assigned to team members is a local management decision: the orthodox approach is to call a team-wide meeting and discuss it, but other approaches are possible (one-to-one interactions between developers and technical manager, managerial fiat, etc).

   Do not overload developers. Take vacations and holidays into account. The sprint should describe a plausible amount of work for the time available.

2. Execution.

   Daily management during the sprint is a local decision. Suggested best practice includes holding regular "standup" meetings (see §6.5), at which developers discuss their current activities and try to resolve "blockers" which are preventing them from making progress.

   Stories should be executed following the instructions in the Developer Guide as regards workflow, coding standards, review requirements, and so on. It is important to ensure that completed stories are marked as done: experience suggests that this can easily be forgotten as developers rush on to the next challenge, but it is required to enable us to properly track progress as per §**??**.

When completing a story we do not change the number of SPs assigned to it: the SP total reflects our initial estimate of the work involved, not the total time invested. However, we should *also* record the true SPs expended on the issue. This makes it possible to review the quality of our estimates at the end of the sprint. Each individual, with guidance from their Team Lead, should use this information as they strive to improve the accuracy of their planning and estimating.

Avoid adding more stories to a sprint in progress unless it is unavoidable (for example, the story describes a critical bug that must be addressed before proceeding). A sprint should always stay current and should be up-to-date with reality; if necessary, already scheduled stories may be pushed out of a sprint as soon as it is obvious it is unrealistic to expect them to be completed.

3. Review.

   At the end of the sprint, step back and consider what has been achieved. What worked well? What did not? How can these problems be avoided for next time? Was your estimate of the amount of work that could be finished in the sprint accurate? If not, how can it be improved in future? Refer to the burn-down chart for the sprint, and, if it diverged from the ideal, understand why.

   Again, the form the review takes is a local management decision: it may involve all team members, or just a few.

We use JIRA's Agile capabilities to manage our sprints. Each Team Lead is responsible for defining and maintaining their own agile board. The board may be configured for either Scrum or Kanban style work as appropriate: the former is suitable for planned development activities (e.g. Science Pipelines development); the latter for servicing user requests (e.g. providing developer support).

6.3 Closing Epics

An epic is considered complete and may be marked as done when:

1. It contains at least one completed story;

2. There are no more incomplete storys defined within it;
3. There are no plans to add more storys;
4. (If applicable, i.e. it is not a bucket, as defined in §5.1.4) its concrete deliverable has been achieved.

"Bucket" epics should be closed at the end of the time box (i.e. end of half fiscal year and end of fiscal year). Note that it is not permitted to close an epic without defining at least one story within it. Empty epics can never be completed.

6.4 Jira Maintenance

At any time, new tickets may be added to JIRA by team members. Please remind your team of the best practice in this respect (RFC-147). It is the responsibility of technical managers to ensure that new tickets are handled appropriately, updating the schedule to include them where necessary.

It is required that the `Team` field be set to the appropriate team (RFC-145). This indicates which manager is responsible for seeing that the work is completed successfully. Available teams, and the associated managers, are listed in the Developer Guide; generally speaking, they align with the the work breakdown structure described in §2.3. Where there is uncertainty about which team should be responsible for a particular ticket, the "Data Production Management" team may be used to indicate that the AD of Data Production is responsible for assigning the work.

Please regularly monitor JIRA for incomplete tickets and update them appropriately. Where tickets describe bugs or other urgent emergent work which cannot be deferred, refer to §5.1.4.

6.5 Coordination Standup

7 Tracking Progress and Standard Reporting Cycle

7.1 Tracking Progress toward Milestones

Progress on completing epics is visualized in Smartsheet. Smartsheet lists all Level 1 through Level 3 milestones in a gantt-chart style view. Each Level 1 milestone is achieved by completing a series of Level 2 and/or Level 3 milestones. Smartsheet tracks Story Points marked as complete in individual JIRA epics in real time. Progress on individual milestones is shown as the weighted total of Story Points within each epic contributing to the successful completion of the milestone.

7.2 Reporting Cycle

High level milestone progress will be reported to SLAC and NOIRLab regularly. NOIRLab reports will flow quarterly (or monthly) to the NSF. Rubin will show progress on all Lens 1 (L1) milestones and any L2 milestones called out in the POP.

8 Annual Budget Planning

Rubin Operations are funded at approximately equal levels by the NSF and DOE. NSF funding is requested by NOIRLab through its 5-year renewal proposal process, and annual budgets submitted as part of the NOIRLab Program Operations Plan (POP). DOE funding is requested by SLAC in an annual Field Work Proposal (FWP). In this section we describe the annual cycle of budget development and spending planning.

This section defines and formalizes a ground-up process to prioritize the annual scope of the program, review and scrub budgets and performance, and develop, using Rubin's planning tools, a consolidated and detailed, cost estimate informed by the schedule milestones.

It also describes our approach to risk-aware budgeting, and a process for reviewing and approving requests beyond baseline targets.

8.1 The annual budget planning cycle

Figure 1 illustrates the phases of Rubin Operations budget development and spending planning throughout the year, as laid out in Section 4.1. In April, following several months collecting high level and external budget information and implementing changes requested during the previous summer, the Director's Office sets major (Level 1 and some Level 2) milestones, and issues departmental budget guidance for the next two fiscal years. (Ideally, the JOR would be held in early April, to provide external expert oversight of the plan.) The major milestones and budget are inserted into the draft NOIRLab POP and SLAC FWP. (Only the budget for the coming fiscal year is required for the POP.)

In May, the Departments and Teams hold a series of meetings to set their minor (remaining Level 2 and Level 3) milestones, and "scrub" their budgets, milestones, and risks, in order to propose their next-year spending plans (to the Director's Office) by the end of May. We provide each department with a "sandbox" workbook to work up during this period. We initialize this sandbox with the department's labor and non-labor plans (that were used in setting the budget), the current staffing plan, the progress against milestones in the current fiscal year, and the major milestones for the coming fiscal year. We then invite them to "scrub" their plans, by working this sandbox up with analysis of recent performance, and any changes they want to propose for the next fiscal year, based on the minor milestones and associated activities they want to prioritize. We also invite them to scrub their part of the risk register, update their risk response plans in concert with their planned activities, and anticipate any requests beyond target they may need to make.

Typically, each team works on its part of the sandbox mostly independently, and then presents their scrubbed plan to the department's management team around mid-May in a "pitch" session. (We provide a template slide deck for each team's pitch, to ensure that all aspects of the sandboxing/scrubbing exercise are addressed.) The teams then address the feedback they have been given by the management team, and then they present their scrubbed plans to the other teams in the department, the Directorate and Program Operations teams from the Director's Office, and the teams in the other departments, at the sandbox workshop at the end of the month. This sandbox workshop at the end of May / beginning of June gives all the departments, including the Director's Office, a chance to review, and probe, the ground-up planning in each team. (In practice, only the four technical departments' management teams are likely to sit through the entire workshop, although it is organized inclusively so that anyone could choose to review all the scrubbed plans.)

In most cases a team's proposed changes in effort distribution, staffing, and purchasing will either involve only small deviations from budget guidance, or stay within the "box" given to them. Teams may also coordinate their proposed changes so as to keep the department's overall spending plan within its budget. (This coordination may be facilitated by the department's management team, e.g. following the pitch sessions.)

Following the Sandbox Workshop, the Director's Office adjusts and finalizes the overall spending plan by the end of June, in time for the submission of the NOIRLab POP, and SLAC's DOE budget briefing. This June period of post-sandbox spending plan development and cost calculation involves the Directors Office staff making updates to the planning tools (Appendix B), with review by the departmental management teams.

We implement the changes to the spending plan, once captured in the Rubin planning tools, in NOIRLab's and SLAC's lab-wide budget models in July. Statements of work for the next fiscal year's contracts and MPOs are straightforwardly derived from the scrubbed plans, for their July deadlines.

In September, the teams then plan work (for the first half of the new fiscal year) against their minor milestones, incorporating any community input they may have collected at the annual project and community workshop (Project Community Workshop (PCW)) in August.

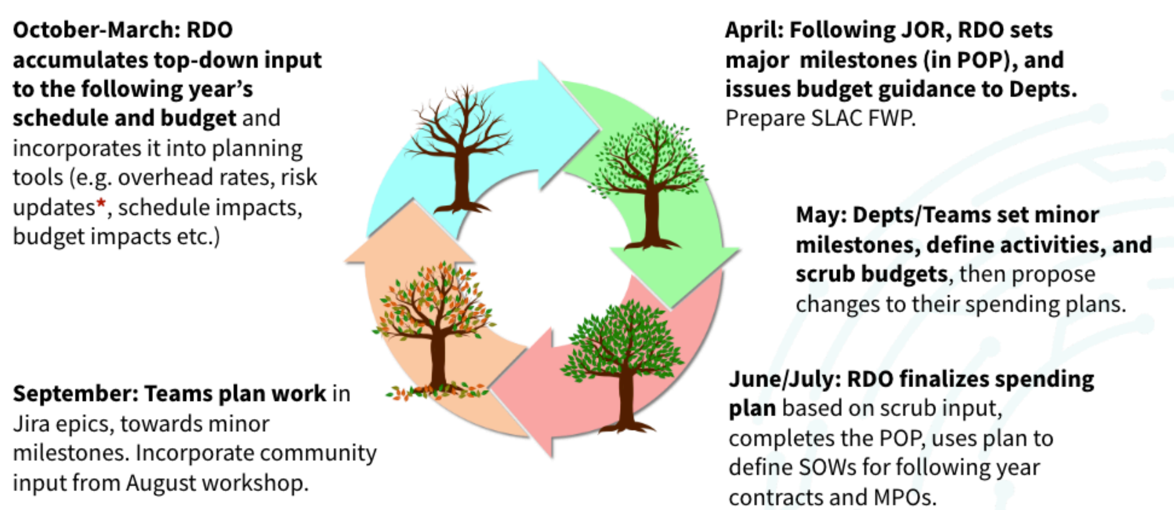


FIGURE 1: Annual budget planning cycle.

8.2 Risk-aware budgeting and Requests Beyond Target

The ground-up spending plan developed during the annual cycle is to cover operations work defined by the milestone activities, as well as any regular work captured in bucket epics. This work includes mitigation of risks held by the teams, as described in the response plans in the risk register. Any risks that are realized during the year will require action that is not budgeted for. Additional spending to address realized risks is enabled by Rubin's request beyond target (Request Beyond Target (RBT)) process, whereby reserve funds are released for use by the departments in response to a specific mid-year request made in a custom Jira ticket.

Standard terminology for financial reserves is that "management reserve" describes funds held back to pay for the unknown unknowns, while "contingency reserve" is used for known unknowns (i.e. the risks in the risk register). Rubin Operations' two funding agencies offer differing guidelines on how reserve should be managed. DOE allows reserves to be built up and carried forward at SLAC, and used as both contingency and management reserve. DOE High Energy Physics (HEP) holds additional reserves centrally, which can also be used as both contingency and management reserve. NSF NSF Division of Astronomical Sciences (AST) allows line item budget over-estimation by up to 10%, to allow for increases in cost. These generate carry-forward (along with any other under-spending that may occur), which we use as contingency reserve. NSF holds management reserve centrally, to handle unexpected events.

The cost impact analysis of the Rubin Operations risk register informs the level of contingency reserve needed. Residual (post-mitigation) cost exposure for each risk is computed by taking the product of the residual probability of the risk being realized (in any given year) and the likely annual cost of addressing that risk. The total cost exposure is then the sum, over all risks in the register, of these products. While Rubin holds a number of potentially very costly risks, these are generally very unlikely to occur.

Currently, the total cost exposure, and hence the needed level of contingency reserve, is expected to be approximately 15% of the total annual operating budget at each operations partner. Included in the risk register is the risk of federal funding allocations being reduced, with a likely cost (if realized in any given year) of some $1.5M per operations partner. (At SLAC, for example, this would cover about 7% of the operating expense – enough to pay the labor bills during a 1 month funding allocation delay.)

Reserve maintenance is accounted for in our federal budget requests, either as an explicit difference between cost and budget on the DOE side, or as line by line allowances for increases in cost on the NSF side. At both operations partners carry forward is maintained from one year's plan to the next, with it being spent down to zero either during the post-operations period (at SLAC), or before the end of each 5-year renewal period at NOIRLab. Each year a plan is described in the NOIRLab POP for how carry forward is allocated. (The restriction on carrying forward funds between renewal periods means that the level of reserve held at NOIRLab varies significantly over the course of the 5 years. Risks that are realized early in the 5-year period are more likely to need supplemental funding from NSF's centrally-held reserves than those that are realized later on, when the carry-forward has been built up. It may be possible to purposefully hold back scope so as to build up the carry-forward quickly at the start of a renewal period.)

An RBT is typically made by a department in order to address risks and/or opportunities as they are realized. If approved, the RBT leads to a draw on the reserve held at one of the operations partners. (At NOIRLab, carry-forward use must be declared a year in advance in the POP: this is why the sandboxing process includes a request that the operations teams pitch RBTs for the coming year at that stage.) The RBT itself is a Jira ticket submitted by a department's management team on behalf of the team that is addressing the risk. Review is by the Directorate team, as budget holders. The spending associated with an approved RBT is not captured in a change to the labor or non-labor plan (which together define the baseline spending plan), but instead is tracked as a variance relative to those plans.

9 Personnel

9.1 Staffing Changes

In addition to onboarding procedures at your local institution, please be aware of

- The Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope) (LSST) New Employee Onboarding material, and

and direct new recruits to them when they join your team[2].

The responsible hirere must also complete an onboarding form for the new recruit. When members of staff team leave the project, the Technical/Control (or Cost) Account Manager (T/CAM) should fill in an offboarding form.

[2] As per §3.2.1, remember that newcomers should be allocated SPs for working through this material.

10 Open issues

- Kanban for LOE operations work
- Need section on more procedural driven work on mountain and DF.

A References

B Glossary

AD Associate Director.

algorithm A computational implementation of a calculation or some method of processing.

Association of Universities for Research in Astronomy consortium of US institutions and international affiliates that operates world-class astronomical observatories, AURA is the legal entity responsible for managing what it calls independent operating Centers, including LSST, under respective cooperative agreements with the National Science Foundation. AURA assumes fiducial responsibility for the funds provided through those cooperative agreements. AURA also is the legal owner of the AURA Observatory properties in Chile.

AST NSF Division of Astronomical Sciences.

AURA Association of Universities for Research in Astronomy.

Business Manager The person responsible for all business activities of the LSST Project and the LSST Corporation; he or she serves as liaison to AURA CAS, develops and monitors contracts, and serves as the LSST Corporation Secretary.

cadence The sequence of pointings, visit exposures, and exposure durations performed over the course of a survey.

cycle The time period over which detailed, short-term plans are defined and executed. Normally, cycles run for six months, and culminate in a new release of the LSST Software Stack, however this need not always be the case.

Data Management The LSST Subsystem responsible for the Data Management System (DMS), which will capture, store, catalog, and serve the LSST dataset to the scientific community and public. The DM team is responsible for the DMS architecture, applications, middleware, infrastructure, algorithms, and Observatory Network Design. DM is a distributed team working at LSST and partner institutions, with the DM Subsystem Manager located at LSST headquarters in Tucson.

Data Release The approximately annual reprocessing of all LSST data, and the installation of the resulting data products in the LSST Data Access Centers, which marks the start of the two-year proprietary period.

DEI Diversity, Equity, and Inclusion.

Department of Energy cabinet department of the United States federal government; the DOE has assumed technical and financial responsibility for providing the LSST camera. The DOE's responsibilities are executed by a collaboration led by SLAC National Accelerator Laboratory.

DF Data Facility.

Director The person responsible for the overall conduct of the project; the LSST director is charged with ensuring that both the scientific goals and management constraints on the project are met. S/he is the principal public spokesperson for the project in all matters and represents the project to the scientific community, AURA, the member institutions of LSSTC, and the funding agencies.

DOE Department of Energy.

Earned Value Management System A set of tools, techniques and procedures which are used to implement a EVM approach to project management.

Education and Public Outreach The LSST subsystem responsible for the cyberinfrastructure, user interfaces, and outreach programs necessary to connect educators, planetaria, citizen scientists, amateur astronomers, and the general public to the transformative LSST dataset.

element A node in the hierarchical project WBS.

epic A self contained work with a concrete deliverable which my be scheduled to take place with a single cycle and WBS element.

EPO Education and Public Outreach.

ES Early Science.

EVMS Earned Value Management System.

FTE Full-Time Equivalent.

Full-Time Equivalent A unit equivalent to one person working full time for one year with normal holidays, vacations, and sick time. No paid overtime is assumed.

FWP Field Work Proposals.

FY Financial Year.

FY23 Financial Year 23.

FY24 Financial Year 24.

HEP High Energy Physics.

IDF Interim Data Facility.

JIRA issue tracking product (not an acronym but a truncation of Gojira the Japanese name for Godzilla).

JOR Joint Operations status Review.

L1 Lens 1.

L2 Lens 2.

L3 Lens 3.

LOE Level of Effort.

LSST Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope).

monitoring In DM QA, this refers to the process of collecting, storing, aggregating and visualizing metrics.

MPO Memorandum Purchase Order.

National Science Foundation primary federal agency supporting research in all fields of fundamental science and engineering; NSF selects and funds projects through competitive, merit-based review.

NICRA (federally) Negotiated Indirect Cost Rate Agreement.

NOIRLab NSF's National Optical-Infrared Astronomy Research Laboratory; `https://nationalastro.org`.

NSF National Science Foundation.

Operations The 10-year period following construction and commissioning during which the LSST Observatory conducts its survey.

PCW Project Community Workshop.

POP Project Operating Plan.

Q2 Quarter two.

RAS Resource Allocation Sheet.

RBT Request Beyond Target.

RDM Rubin Data Management.

RDO Rubin Directors Office.

RDP Rubin Data Production(Obsolete use RDM).

Resource Allocation Sheet Shows the detailed FTE loading to produce NOIRLab budgets.

Review Programmatic and/or technical audits of a given component of the project, where a preferably independent committee advises further project decisions, based on the current status and their evaluation of it. The reviews assess technical performance and maturity, as well as the compliance of the design and end product with the stated requirements and interfaces.

Risk The degree of exposure to an event that might happen to the detriment of a program, project, or other activity. It is described by a combination of the probability that the risk event will occur and the consequence of the extent of loss from the occurrence, or impact. Risk is an inherent part of all activities, whether the activity is simple and small, or large and complex.

ROO Rubin Observatory Operations.

RPF Rubin system PerFormance.

RSS square root of the sum of the squares.

Rubin Operations operations phase of Vera C. Rubin Observatory.

Safety The control of accidental loss.

Science Pipelines The library of software components and the algorithms and processing pipelines assembled from them that are being developed by DM to generate science-ready data products from LSST images. The Pipelines may be executed at scale as part of LSST Prompt or Data Release processing, or pieces of them may be used in a standalone mode or executed through the Rubin Science Platform. The Science Pipelines are one component of the LSST Software Stack.

seeing An astronomical term for characterizing the stability of the atmosphere, as measured by the width of the point-spread function on images. The PSF width is also affected by a number of other factors, including the airmass, passband, and the telescope and camera optics.

SLAC SLAC National Accelerator Laboratory.

SLAC National Accelerator Laboratory A national laboratory funded by the US Department of Energy (DOE); SLAC leads a consortium of DOE laboratories that has assumed responsibility for providing the LSST camera. Although the Camera project manages its own schedule and budget, including contingency, the Camera team's schedule and requirements are integrated with the larger Project. The camera effort is accountable to the LSSTPO..

SP System PerFormance.

story A JIRA issue type describing a scheduled, self-contained task worked as part of an epic. Typically, stories are appropriate for work worth between a fraction of a SP and 10 SP; beyond that, the work is insufficiently fine-grained to schedule as a story. While fractional SP are fine, all stories involve work, so the SP total of an in progress or completed story should not be 0.

Summit The site on the Cerro Pachón, Chile mountaintop where the LSST observatory, support facilities, and infrastructure will be built.

Systems Engineering an interdisciplinary field of engineering that focuses on how to design and manage complex engineering systems over their life cycles. Issues such as requirements engineering, reliability, logistics, coordination of different teams, testing and evaluation, maintainability and many other disciplines necessary for successful system development, design, implementation, and ultimate decommission become more difficult when dealing with large or complex projects. Systems engineering deals with work-processes, optimization methods, and risk management tools in such projects. It overlaps technical and human-centered disciplines such as industrial engineering, control engineering, software engineering, organizational studies, and project management. Systems engineering ensures that all likely aspects of a project or system are considered, and integrated into a whole.

T/CAM Technical/Control (or Cost) Account Manager.

timebox A limited time period assigned to a piece of work or other activity. Useful in scheduling work which is not otherwise easily limited in scope, for example research projects or servicing user requests.

UK United Kingdom.

US United States.

Validation A process of confirming that the delivered system will provide its desired functionality; overall, a validation process includes the evaluation, integration, and test activities carried out at the system level to ensure that the final developed system satisfies the intent and performance of that system in operations.

Verification The process of evaluating the design, including hardware and software - to ensure the requirements have been met; verification (of requirements) is performed by test, analysis, inspection, and/or demonstration.

WBS Work Breakdown Structure.

Work Breakdown Structure a tool that defines and organizes the LSST project's total work scope through the enumeration and grouping of the project's discrete work elements.

C Rubin Operations Planning Tools

The cost and schedule of Rubin Operations is derived using a suite of planning tools, that capture the logical flowdown from the observatory's operations requirements to its work breakdown structure (WBS) and major milestones, and from there to its staffing plan and budget.

Since the management of Rubin Operations is shared between two operations partners, NOIRLab and SLAC, whose staff are organized into integrated, multi-partner teams, these planning tools must be collaborative and shareable between institutions. Enabling ground-up development and delegated ownership of the overall plan leads to a further requirement that the planning tools be easy and intuitive to use. These two considerations have led to Rubin Operations' development of its planning tools as a set of custom-built, inter-connected Google sheets workbooks. The schedule itself is defined as a set of JIRA milestones, while work is planned using JIRA epics and stories; Smartsheet provides additional enterprise-level milestone planning and tracking capability.

The Rubin planning tools are as follows:

- The **Data Preview and Release Planning Tool (DPRPT) workbook** supports the derivation of Rubin's primary set of major milestones, the LSST data releases. It also captures the high level planning of the data release contents, a critical activity during the pre-operations phases "Data Previews."
- The **Work Breakdown Structure (WBS) workbook** contains the WBS for Rubin Operations, leading to the derivation of the departments, teams and groups that make up the operations organization. It also hosts the Labor and Non-labor plans, defining the labor roles and their needed effort profiles, and the non-labor items (equipment, services, etc), that each team needs to carry out its part of the work.
- The **Staffing Plan workbook** shows how the labor plan is being staffed. (The WBS Labor plan shows the *needed* FTE effort in each role, while the Staffing Plan shows the *planned* FTE effort from each team member in each role.) It imports dynamically the WBS Labor plan, and provides a number of cross-checks against it.
- The **Cost Calculator (CC) workbook** imports dynamically the WBS Labor and Nonlabor plans, computes the cost profile associated with each item, and produces various summaries needed for our budget requests. The Staffing Plan is not used directly, but an approximate average salary per role is estimated externally using the Staffing Plan as a guide, and then entered into one of the CC's data sheets. The CC's costing is therefore approximate: we estimate that it forecasts with better than 1

Each tool is internally documented with a README and a Change Record, and cell comments are used for discussion of changes at that level.

C.1 Updating and Versioning

These planning tools are being evolved continuously. A major update of the Staffing Plan, WBS, and Cost Calculator is carried out following each sandboxing exercise. Because they are all interconnected, they carry the same version number. Ideally, the joint operations review happens just before such an update is begun; a natural time to archive the current version and advance to the new version is when the plan is frozen for joint agency review. Mechanically, a copy of each workbook is made, with “ARCHIVED” at the end of its name, the connections are edited so that the archived Staffing Plan and CC workbooks import from the corresponding archived copy of the WBS and not the current version, and then all 3 archived copies are transferred to an Attic folder. The current versions are then renamed with the new version number, and edited until it is time to freeze them again.