LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

Data Management Test Plan

William O'Mullane, Mario Juric, Frossie Economou

LDM-503

Latest Revision: 2017-06-13

Abstract

This is the Test Plan for Data Management. In it we define terms associated with testing and further test specifications for specific items.

Change Record

Version	Date	Description	Owner name
D	2017-01-13	First draft	William O'Mullane

Contents

Data Management Test Plan

1 Introduction

In this document we lay out the verification and validation approach for LSST Data Management. In addition we outline some of the high level test milestones in Section 6 and our planned schedule for demonstrating interim verification status.

1.1 Objectives

We describe the test and verification approach for DM and describe various constraints and limitations in the testing to be performed. We also describe the validation tests to be performed on the partially and fully integrated system. We do not describe all tests in details; those are described in dedicated test specifications for major components of Data Management. Here we outline the required elements for those specifications as well as the tools we use to for continuous verification.

1.2 Scope

This provides the approach and plan for all of Data Management. It covers interfaces between Data Management and components from other LSST subsystems but nothing outside of Data Management. This document is change-controlled by the DMCCB and will be updated in response to any requirements updates or changes of approach.

1.3 Assumptions

We will run large scale Science Validations in order to demonstrate the system's end-to-end capability against its design specifications. A large amount of informal science validation will be done in the the teams and documented in technical notes; in this test plan we are looking for validation of the broader system and specifically *operability* i.e. whether we can run this system every day for the 10 year planned survey with a reasonable level of operational support.

1.4 Applicable Documents

When applicable documents change a change may be required in this document.

LPM-55	LSST Quality Assurance Plan
LDM-294	DM Project Management Plan
LDM-148	DM Architecture

1.5 References

[1] **[LSE-29]**, Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2016, *LSST System Requirements*, LSE-29, URL https://ls.st/LSE-29

[2] **[LSE-30]**, Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2016, *LSST System Requirements*, LSE-30, URL https://ls.st/LSE-30

[3] **[LSE-61]**, Dubois-Felsmann, G., 2016, *LSST Data Management Subsystem Requirements*, LSE-61, URL https://ls.st/LSE-61

[4] **[LPM-17]**, Ivezić, Ž., The LSST Science Collaboration, 2011, *LSST Science Requirements Document*, LPM-17, URL https://ls.st/LPM-17

[5] **[LSE-163]**, Juric, M., et al., 2017, *LSST Data Products Definition Document*, LSE-163, URL https://ls.st/LSE-163

[6] **[LDM-148]**, Kantor, J., Axelrod, T., 2013, *Data Management System Design*, LDM-148, URL https://ls.st/LDM-148

[7] **[LDM-240]**, Kantor, J., Juric, M., Lim, K.T., 2016, *Data Management Releases*, LDM-240, URL https://ls.st/LDM-240

[8] **[LDM-294]**, O'Mullane, W., DMLT, 2017, *Data Management Project Management Plan*, LDM-294, URL https://ls.st/LDM-294

[9] **[LPM-55]**, Sweeney, D., McKercher, R., 2013, *Project Quality Assurance Plan*, LPM-55, URL https://ls.st/LPM-55

[10] **[LSE-63]**, Tyson, T., DQA Team, Science Collaboration, 2011, *Data quality Assurance Plan: Requirements for the LSST Data Quality Assessment Framework*, LSE-63, URL `https://ls.st/LSE-63`

1.6 Definitions, acronyms, and abbreviations

The following table has been manually generated pending automatic generation from the online LSST acronym list:

Acronym	Description
AP	Alerts Production
API	Application Programming Interface
CAM	CAMera
CI	Configuration Item
CU	Coordination Unit (in DPAC)
DAC	Data Access Center
DAQ	Data AcQuisition (system)
DAX	Data Access Services
DBB	Data BackBone
DM	Data Management
DMCCB	DM Change Control Board
DMLT	DM Leadership Team
DPC	Data Processing Centre
DRP	Data Release Production
EFD	Engineering Facilities Database
HSC	Hyper Suprime-Cam
ICD	Interface Control Document
JIRA	issue tracking product (not an acronym, but a truncation of Gojira, the Japanese name for Godzilla)
KPM	Key Performance Metric
LCR	LSST Change Request
LSST	Large-aperture Synoptic Survey Telescope
NCSA	National Center for Supercomputing Applications

OCS	Observatory Control System
OPS	OPerations
PDAC	Prototype Data Access Center
QA	Quality Assurance
QC	Quality Control
QSERV	In-house LSST Database system
SP	Software Product
SPR	Software Problem Report
SUIT	Science User Interface Team
SV	System Verification
TBD	To Be Defined (Determined)
TCT	Technical Control Team (Obsolete - now DMCCB)
UX	User InterFace widget
WBS	Work Breakdown Structure
WISE	Wide-field Survey Explorer

2 Roles and Reporting

Each test specification must make clear who the *tester* is.

Testers report issues through Jira and also write a test report. The test reports will be used to populate the verification control document see Section **??**. At this time we are aware of some plans in System Engineering to use Jira plugins to track commissioning tests - we have to see how this might work for DM verification.

Operations rehearsals require someone to direct them, these are more about the process than tests. The rehearsal can not be directed by the Operations Manger since that person has a major role in the rehearsal. Some external individual must be found to perform this role.

Tests and procedures will sometimes fail - a test specification may be rerun several times to get it correct.

For large scale tests and rehearsals there should be a Test Review Board nominated to write

up the findings as well as decide on timescales for rerunning part or all of a test in case of failure.

3 DM Verification Approach

We intend to verify DM requirements by testing of DM components in a fairly standard engineering manner. For each high level component a test specification will be produced defining a set of tests related to the requirements for the component. These specifications are represented on the top of Figure 1. Any given requirement may have several tests associated with it in the specification, the tests may be phased with the need for certain functionality at a specific time.

The test spec will cover all aspects of the test as outlined in Section 3.2. These high level test specifications may call out individual lower level test specification.

As we execute tests we will gather test reports on the Pass/Fail of the individual tests related to specific requirements. This Information will allow us to build a Verification Control Document (VCD) (right of Figure 1). The VCD will provide a % verification of each requirement in DM (rolled up to OSS requirements also). Figure 1 currently calls for a report from each test spec - this may be captured directly in Jira.

The DM components are outlined in LDM-294 and detailed in LDM-148. At a high level these components are represented in figure Figure 2. Based on those components we can see the set of Test Specifications needed in Table 2. That table does not contain all of the document numbers yet.

3.1 Test Items

The test items covered in this test plan are:

- Data Management and its primary components for testing and integration purposes. These are listed in Table 2. All components listed in orange and yellow have specifications in the corresponding documents listed. Major sub-components in white may have individual test specifications or be addressed in the component they are under depend-

TABLE 2: Components from LDM-148 with the test specifications to verify them.

Component	Testing Spec
NCSA Enclave	LDM-532
- L1 System	LDM-533
- - L1 Prompt Processing	???
- - L1 Alert Distribution	???
- - L1 Alert Filtering (mini Broker)	???
- - L1 Quality Control	???
- - L1 OCS Batch Processing	???
- - L1 Offline Processing	???
- L2 System	LDM-534
- - L2 QC	???
- - L2 Data Release	???
- - L2 Calibration Products	???
Data Backbone	LDM-535
- DBB Data Services	LDM-536
- - DBB QSERV	???
- - DBB Databases	???
- - DBB Image Database/Metadata Prov	???
- - DBB Data Butler Client	???
- DBB infrastructure	LDM-537
- - DBB Tape Archive	???
- - DBB Cache	???
- - DBB Data Endpoint	???
- - DBB Data Transport	???
- - Networks	???
Base Enclave	LDM-538
- - Prompt Processing Ingest	???
- - Telemetry Gateway	???
- - Image and EFD Archiving	???
- - OCS Driven Batch Control	???
Data Access Center Enclave	LDM-539
- - Bulk Data Distibution	???
- - Science Platform	LDM-540
- - Science Platform JupyterLab	???
- - Science Platform Portal	???
- - DAX VO+ Services	???
Commisioning Cluster Enclave	LDM-541
- - SuperTask ..	

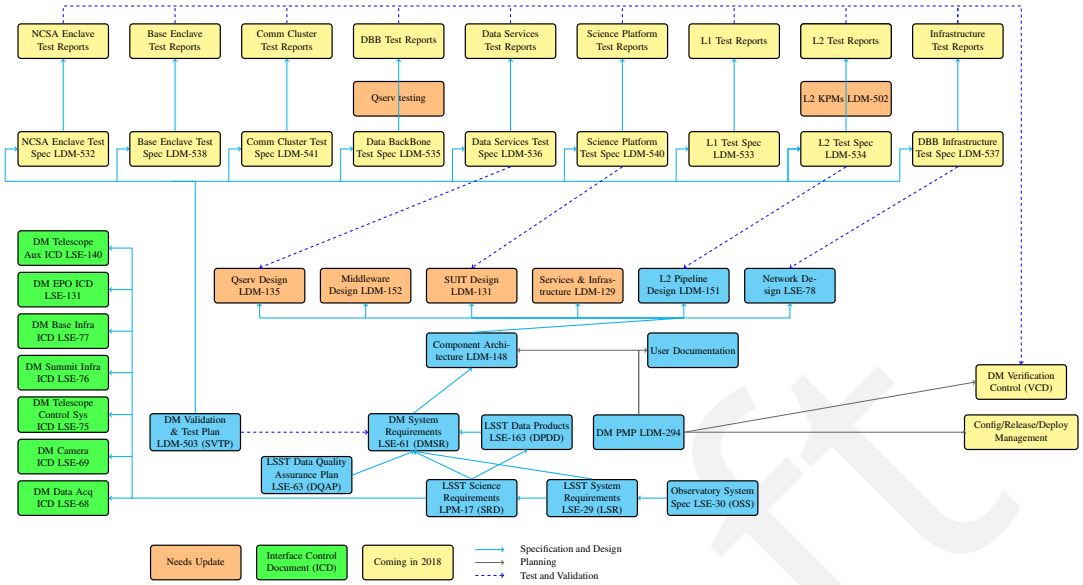

FIGURE 1: Documentation tree for DM software relating the high level documents to each other. (from LDM-294

ing on applicable factors such as whether they are scheduled for testing at the same time and/or whether they share architectural components or are largely distinct.

- The external interfaces between Data Managementand other sub-systems. These are described in [Docushare collection]
- Operational procedures like Data Release Process and Software Release Process. [We don't have a list]

3.2 Testing Specification document format

The Testing Specification documents are drawn in conjunction with the LSST System Engineer. In all cases they include:

- A list of components being tested within the scope of the Test Specification Document.
- A list of features in those components that are being explicitly tested.
- How those features related to identified requirements for that component
- A description of the environment in which the tests are carried out (eg. hardware platform) and a description of how they differ from the operational system in tests prior to

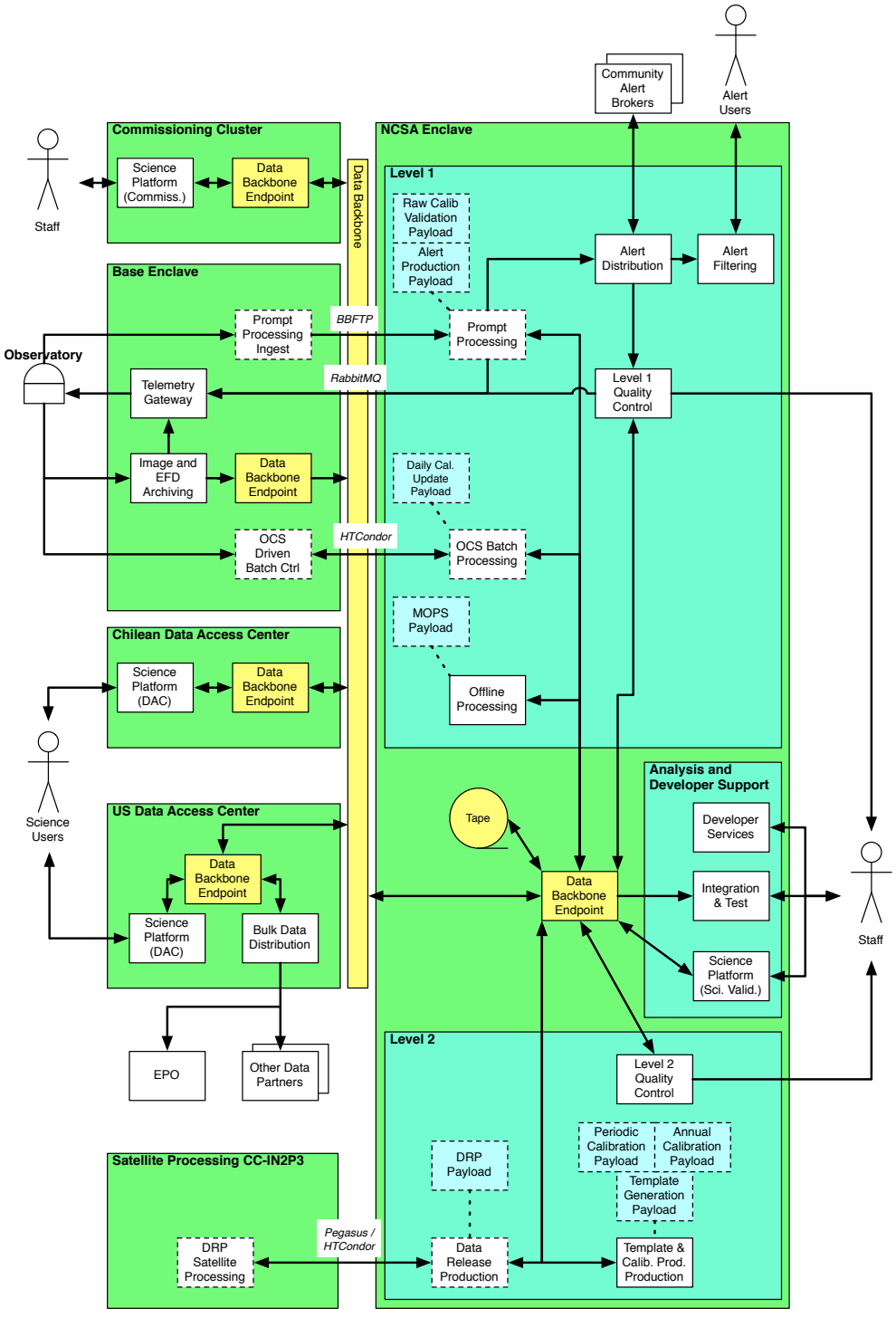

FIGURE 2: DM components as deployed during Operations. Where components are deployed in multiple locations, the connections between them are labeled with the relevant communication protocols. Science payloads are shown in blue. (from LDM-148)

final integration (eg. interfaces that may be mocked without affecting that component's testing).
- The inputs (eg. data, API load) that are to be used in the test
- Pass-fail criteria (eg. metrics to be met)
- How any outputs that are used to determine pass/fail (eg. data or metrics) are published (made available).
- A Software Quality Assurance manifest listing (as relevant) code repositories, configuration information, release/distribution methods and applicable documentation (such as installation instructions, developer guide, user guide etc.)

4 Pass/Fail Criteria

A Test Case will be considered "Passed" when:

- All of the test steps of the Test Case are completed and
- All open SPRs from this Test Case agreed in Software Review Board are considered non-critical.

A Test Case will be considered "Partially Passed" when:

- Only a subset of all of the test steps in the Test Case are completed but the overall purpose of the test has been met and
- Any critical SPRs from this Test Case agreed in Software Review Board are still not closed.

A Test Case will be considered "Failed" when:

- Only a subset of all of the test steps in the Test Case are completed and the overall purpose of the test has not been met and
- Any critical SPRs from this Test Case agreed in Software Review Board are still not closed.

Note that in LPM-17 science requirements are described as having a minimum specification, a design specification and a stretch goal. We preserve these distinctions where they have been

made in, for example, the verification framework and automated metric harness. However for the purposes of Pass/Fail criteria, it is the design specification that is verified as having been met for a test to pass without intervention of the Software Review Board.

In the event that a requirement is failing its design specification and the minimum specification is invoked, this is an LSST project level issue and is escalated beyond the scope of this plan.

5 Constraints and Limitations

- Verification is being done on the basis of precursor data sets such as HSC, and eventually with engineering data from the LSST arrays. These are just a proxy for full-focal-plane on-site LSST data.
- Metric measurements and operational rehearsals during construction may not involve critical operational systems that are still in development. For example, while computational performance is being measured, computationally dominant algorithmic steps such as deblending and multi-fit are only modeled, since they have not yet been implemented; operational rehearsals are done without the factory LSST workflow system; etc.

5.1 Requirements Traceability Constraints

Note

I felt a summary of the current state of play being verified could be useful to Wil. We don't have to leave it in the final document –FE

5.1.1 Scientific

Some science requirements are captured in LSE-29 (aka LSR) and flow down to LSE-30 (aka OSS) ; some also exist in LSE-163 (aka DPDD) and will flow down in LSE-61 (aka DMSR).

5.1.2 Computational

There are requirements in LSE-61 (aka DMSR) which captures the LSE-30 (OSS) requirements that DM is responsible for. *In practice LSE-63 (the QA document) has not been flown down to LSE-61*. These are:

- The primary computational performance flown down from LSE-29 (LSR) is OTT1 which is the requirement to issue an alert within 60 seconds of exposure end. DMS-REQ-0004 LSR-REQ-0101
- Another requirement flows down from LSE-29 is calculation of orbits within 24 hours of the end of the observing night DMS-REQ-0004 LSR-REQ-0104 L1PublicT
- There is a new (not yet baselined?) requirement for the calibration pipeline to reduce calibration observations within 1200 seconds
- A nightly report on data quality, data management system performance and a calibration report have to be generated with 4 hours of the end of the night DMS-REQ-0096 dqReportComplTime

Note that there are no computational requirements on individual technical components e.g.. data processing cluster availability, database data retrieval speeds, etc. There is an upper limit on acceptable data loss, and there is a network availability requirement.

5.1.3 KPMs

As a proxy for validating the DM system, LDM-240 (aka “the spreadsheet”) defined a set of Key Performance Metrics that the system could be verified against. KPMs were not formally flowed down from LSE-29 (LSR) through LSE-30 (OSS) although there is some overlap with LSE-29 requirements. In particular, the non-science KPMs only exist in LDM-240 *(spreadsheet/old plan)*.

5.2 Interfaces

We will verify external interfaces to other subsystems and selected major internal interfaces. The ICDs describing external interfaces are curated in Docushare Collection 5201.

6 Master Schedule

The schedule for testing the system until operations commence (currently 2022).

Date/Freq	Location	Title, Description
Nightly	Amazon	**Nightly Tests** Run all automated tests on all DM packages automatically.
Weekly	Amazon	**Integration tests** Basic Sanity check to make sure code compiles at no regressions have occurred and also pushing though a basic data set.
TBP	NCSA	Interface tests The interface tests have to be planned and documented in a separate test plan that should include tests for each two parties on an interface (2by2 tests) as well as tests for all parties. Some of these will be covered again in E2E tests but before that we should be confident they work. **This includes internal and external interfaces.**
TBP	NCSA + IN2P3	End to End Tests ?? Freeze software for Ops .. `https://confluence.lsstcorp.org/display/DM/Data+Processing+End+to+End+Testing` What is the status of these ?
F17	NCSA	**Science Platform with WISE data in PDAC** SUIT continues PDAC development, adding the WISE data, further exercising the DAX dbserv and imgserv APIs, and taking advantage of metaserv once it becomes available
F17	NCSA	**HSC reprocessing** Validate the data products withe LSST stack match or improve the HSC products - thus validating the stack. Validate the ops platform in NCSA. Validate some procedures like installing the stack, patches, starting, stopping production. Generate validation data set for weekly integration and other tests.
F17		**AP alert generation validation** Validate AP alert generation stack performance on several DECam and HSC datasets. Begin continuous integration testing.
Feb 2018	NCSA?	**Camera DAQ Integration Test** The data acquisition hardware should be available to DM 13th Feb. We should test it.

S18	NCSA?	**AP system validation** Validate AP alert distribution and mini-broker system fed by live or simulated live data.
June 2018	NCSA	**DM ComCam system test** ComCam will be in Tucson July 24th, the DM system must be ready to deal with it.
Aug 2018	NCSA	**DM Camera data test** Partial camera data should be available to DM July 31st. We plan to test DM stack with it.
2018	NCSA	**Spectrograph Data acquisition** . . . Do we need a test BEFORE THIS?
Oct 2018	NCSA	**Operations rehearsal for commissioning** With TBD weeks commissioning (lets say a week) - pick which parts of plan we could rehearse. Chuck suggests Instrument Signal Removal should be the focus of this (or the next rehearsal).
Feb 2019	NCSA	**DAC validation** There is a project Milestone that DAC/DM/Networks are available March 15th. We need to run tests in Feb to show this is ready.
Oct 2019	NCSA	**Operations rehearsal #2 for commissioning** More complete rehearsal - where do the scientist look at quality data? How do they feed it back to the Telescope ? How do we create/update calibrations ? Exercises some of the control loops.
Jan 2020	Base	**Operations rehearsal #3 for commissioning** Dress rehearsal - Just like it will be April for the actual commissioning.
Dec 2020	NCSA	**Operations Rehearsal Data Release (Commissioning Data)**
March 2021	Base	**DM Software for Science Verification Test** Science verification starts April, DM software must be installed and validated prior to start of Science Verification (which should be called validation perhaps)
2021	NCSA	**Operations Rehearsal Data Release (Regular Data).**
Feb 2022	NCSA/Base	**Operations Rehearsal(s)** Rehearsals for real operations which start Oct 2022
Sept 2022	NCSA/Base	**Operations Rehearsal(s)** Full Dress rehearsal for real operations which start Oct 2022

7 Tools for Continuous Verification and Software Quality Assurance

7.1 Introduction

A number of tools and development practices are in use in Data Managementto ensure software quality and to verify requirements are met. These tools are used continuously (eg. to measure key performance metrics routinely) or periodically (eg. software release characterizations) and so will be well understood by the time the formal verification phase begins.

7.2 Continuous Integration and Unit Testing

Code is checked via a continuous integration (CI) service both for on-demand developer use and for verifying the quality of the master branch. Irrespective of supported platforms, we have a practice of verifying that the stack can run on at least 2 distinct operating systems/platforms as portability is often a good indicator of maintainability. The CI service also permits verification that the codebase runs with different third party dependencies; for example we test that the python code runs both under (legacy) Python 2.7 and (trailing edge versions of) Python 3. This reduces the foreseeable technical debt of porting to Python 3 for operations.

Unit testing policy is described in the DM Developer guide under Unit Test Policy.

Roles and responsibilities in this area include:

- The DM System Engineering Team team is responsible for approving dependencies and setting strategy such as the Python 3 portability
- The DM System Engineering Team is responsible for setting the Unit Test policy
- The SQuaRE team is responsible for developing, operating and supporting Continuous Integration Services
- The SQuaRE team determines platform release practice ICW the other teams and Architecture

Note

We do not have unit test coverage tooling for Python; this is coming with the planned switch to the pytest framework.

7.3 Code Reviews

DM's process requires that every story resulting in code changes to the stack is reviewed prior to being merged to master. This is both as code quality verification and also to ensure that at least one other team-member has some familiarity with a particular part of the codebase. DM's Code Review process is described in the DM Developer Guide under the section DM Code Review and Merging Process.

Roles and responsibilities in this area include:

- The DM System Engineering Team defines the development process and style guide including the code review standard.
- SQuaRE is responsible for supporting tooling to assist code review (eg. linters, JIRA-Github integration, etc).

7.4 Automated Requirements Verification and KPM Measurement

DM uses a harness for continuous metric verification. In the software development context this is used for:

- Calculating KPMs where available and alerting when they exceed specification
- A regression testing framework for any developer-supplied metric, with optional alerts when excursions occur from past values to verify that performance is not being degraded by new code or environments
- Visualizing these results and linking them back to build and pull request information
- Drill-down of those metrics in pre-defined visualization templates geared towards specific verification use-cases.
  Roles and responsibilities in this area include:

- The pipeline teams are responsible for providing some of the code and data to calculate the KPMs
- SQuaRE is responsible for developing and operating the continuous metric verification services
- Individual developers contribute non-KPM metrics as desired

8 Operations Validation

Operations Validation of the system is done though Oe rations Rehearsals (and or end to end tests). This may repeat some or all of a science validation exercises but in a more operational setting with a focus on operations.

9 Science Validation

9.1 Definition

We define **DM Science Validation** as **the process by which we assess the as-built Data Management system meets the needs of the scientific community and other identified stakeholders**.

We assess the projected and realized scientific usability of the system by periodically exercising the integrated system in a way that goes beyond synthetic unit and integration tests and verification of piece-wise requirements as described in previous sections. In other words, we *attempt to use the system in ways we expect it to be used by the ultimate users of the system, scientists*. An example may be performing a mock science study on the results of processing of precursor data, or performing a mock science-like activity (e.g., interactive analysis of time-domain datasets) on a partially stood-up service (e.g., the Notebook aspect of the LSST Science Platform). We record and analyze any issues encountered in such usage, and feed this information back to the DM Science and DM development teams.

Science Validation exercises are designed to close the design-build-verify loop, and enable one to measure the degree to which the requirements, designs, the as-built system, and future development plans continue to satisfy stakeholder needs. They also provide valuable feedback about modifications needed to ensure the delivery of a scientifically capable system.

Ultimately, SV activities transfer into commissioning SV activities and provide training to the future members of the Commissioning team.

9.2 Schedule and Execution

9.2.1 Schedule

DM SV activities are planned and prepared in a rolling wave fashion in parallel with development activities (on a 6-month cycle, or perhaps a year). The SV activities will typically be designed so as to exercise the capabilities of the system expected to be delivered at the end of a given development cycle. These follow a long-term roadmap of SV activities, linked to product delivery milestones in the DM's Construction Plan (see the table in Section 6). The Science Validation (SV) team guides the definition of goals of those activities, in close consultation with the DM Project Manager.

By their nature, SV activities will typically lag behind deliveries of the (sub)system being verified – ideally, they will commence immediately upon delivery. Preparatory SV activities (e.g., identification and acquisition of suitable datasets, identification of potential Science Collaboration resources to include on the activity, or development of activity-specific analysis codes) will commence as early as feasible. DM SV Scientist will coordinate the execution of all SV activities.

SV activities should aim to take no longer than two months to conclude, to enable rapid actionable feedback to DM Management and DM Subsystem Science.

9.2.2 Execution

Science Validation activities typically follow the successful execution of unit and integration test activities described in the previous sections, especially the larger "dress rehearsals" and "data challenges" as listed in Section 6 (Master Schedule).

Following successful service stand-up or data challenge execution (at integration and unit test level), the generated data products or integrated services are turned over to the SV team. The SV team performs additional tests and data analyses to exercise the integrated system and assess its quality relative to expectations for the current phase of construction. This

assessment is fed back to DM Subsystem Science and Systems Engineering teams to inform them about the status and needed improvements to the system.

Beyond reporting on the results, the SV team examines the tests or procedures developed in this phase and identifies those that are good new metrics of system quality and could be run in an automated fashon. These are fed back to the development teams for productizing and incorporation into the automated QC systems.

9.3 Deliverables

Key deliverables of Science Validation activities are:

- Reports on the assessed capability of the Data Management System to satisfy stakeholder needs. The assessments shall take into account the expected maturity of the system being tested.
- Recommendations for improvements and changes, both in the quality of as-constructed systems (i.e., what needs to be built differently or better, to make it more consistent with the system vision), as well as the overall system vision (i.e., recommendations on where the vision may need to be modified to fully respond to stakeholder needs).
- Measurements of performance metrics that do not lend themselves to easy automation (e.g., science activities requiring human involvement, like visual classification, or UX tests).
- Identification of new performance metrics to be tracked, including potential deliveries of code to the DM Construction and I&T teams for inclusion in automated quality control pipelines.
- Other deliverables as charged when chartering a particular SV exercise.

9.4 Organization and Resources

The DM Subsystem Scientist is accountable to the LSST Project Scientist for successful execution of DM Science Validation activities. This responsibility is delegated to the **DM Science Validation Scientist**, who leads the Science Validation (SV) team.

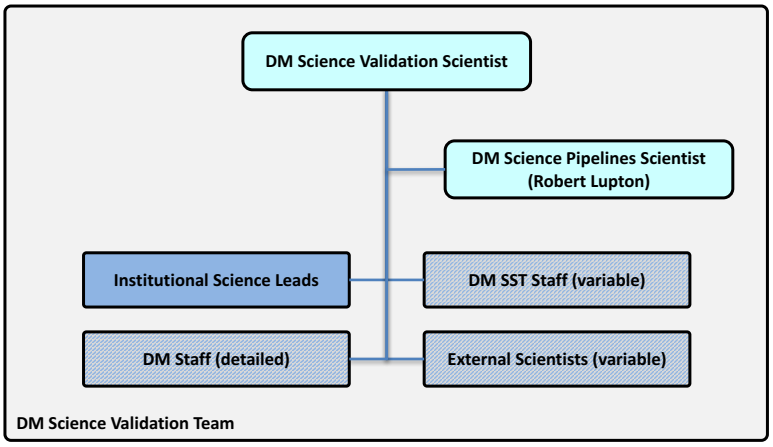

FIGURE 3: Organogram of the Data Management Science Validation Team. The group is chaired by the DM Science Validation Scientist, with the DM Science Pipelines Scientist and Institutional Science Leads making up the permanent membership. Depending on the SV activities being executed at any given time, the group may draw on additional temporary members from DM SST Staff, the broader DM Construction staff, as well as external scientists (e.g., Science Collaboration members committed to assisting SV goals). SV membership is reassessed on a cycle by cycle basis, with estimates incorporated in the long-term plan.

The SV team guides the definition of goals and receives the products of dress rehearsal activities, consistent with the long-term testing roadmap defined in Section 6. Decisions on strategic goals of SV exercises are made in close consultation and coordination with the DM Project Manager and Subsystem Scientist. The results of SV activities are reported to the DM Project Manager and Subsystem Scientist.

SV activities draw on resources of the DM System Science Team, but may also tap into the broader construction team if needed (and as jointly agreed upon with the DM Project Manager), as well as contributors from the LSST Science Collaborations. Additional members may added as needed, depending on SV activities being considered and based on the recommendation of the DM SV Scientist and resource constraints.

The SV Scientist, the DM Science Pipelines Scientist, and all Institutional Science Leads are ex-officio members of the SV Team. DM Project Scientist and Managers are not formal members, but monitor the work of the group.

9.4.1 Example

An example of a Science Validation activity may be as follows:

- Based on the long-term development roadmap and new capabilities expected to be delivered, the at the beginning of a 6-month cycle the SV Team defines the goals of a data challenge to be executed at the end of the cycle. For the purposes of this example, we assume a major new feature to be delivered is astrometric calibration and estimation of proper motions.
- A small data release production using HSC data is defined that should result in a data set sufficient to measure the size and orientation of velocity ellipsoids in the Galactic halo. If such measurement are a success, they would independently validate the newly added global astrometric calibration and proper motion measurement capability.
- At the end the development cycle, the Science Pipelines team delivers to the proto-Operations team a documented and internally tested set of DRP pipelines with the new capabilities as defined above. The pipelines pass all unit and small-scale integration tests. The proto-Operations team deploys and re-verifies the received pipelines in the I&T environment designed to closely mimic the production environment. They verify that the pipeline integrates well with the orchestration system and is capable of executing medium-to-large scale processing. The pipelines pass integration tests.
- The data challenge is operationally planned and executed by the proto-Operations team, including the execution of any predefined QA metrics. The data products and test results are turned over to the Science Validation team.
- The Science Validation team performs the analysis needed to achieve SV exercise goals (the measurement of vellocity ellipsoids, in this case).
- The results and conclusions derived from the data challenge are fed back to the DRP team, DM Project Management, and DM Subsystem Science; they may be used to assess the overall quality of the product, pass a formal requirement, and/or inform future construction decisions.
- Any newly developed but broadly useful tests are identified as such, and fed to the I&T team for inclusion into the battery of tests that are run on a regular basis.