LARGE SYNOPTIC SURVEY TELESCOPE

# Large Synoptic Survey Telescope (LSST)

# Data Management Test Plan

**William O'Mullane, Mario Juric, Frossie Economou**

**LDM-503**

**Latest Revision: 2017-05-24**



## 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

1 Introduction 1
1.1 Objectives 1
1.2 Scope 1
1.3 Assumptions 1
1.4 Applicable Documents 1
1.5 References 2
1.6 Definitions, acronyms, and abbreviations 3
2 Test Items 4
3 Roles and Reporting 4
3.1 Pass/Fail Criteria 5
3.1.1 Key Performance Metrics 6
4 Constraints and Limitations 6
4.1 Requirements Traceability Constraints 7
4.1.1 Scientific 7
4.1.2 Computational 7
4.1.3 KPMs 8
4.2 Functional Requirements 8
4.3 Interfaces 9
5 Master Schedule 9

**6 Validation Tools** **10**

6.1 Introduction . . . . . 11

6.2 Data Comparison Tools . . . . . 11

6.3 Data Transformation Tools . . . . . 11

6.4 Analysis Tools . . . . . 12

**7 Unit and Integration Tests** **12**

7.1 Approach . . . . . 12

7.2 Test Coverage . . . . . 12

7.3 Component and Integration Test Specification . . . . . 12

**8 Validation Tests** **14**

8.1 General strategy . . . . . 14

8.2 Test Designs . . . . . 14

8.2.1 Test Design DM-Data Management-SYS-X . . . . . 14

8.3 Test Case Specification . . . . . 15

8.3.1 Test Case DM-Data Management-SYS-X-1 . . . . . 15

**9 Science Validation** **15**

9.1 Definition . . . . . 16

9.2 Schedule and Execution . . . . . 16

9.2.1 Schedule . . . . . 16

9.2.2 Execution . . . . . 17

9.3 Deliverables . . . . . 17

9.4 Organization and Resources . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
9.4.1 Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

# Data Management Test Plan

## 1 Introduction

In this document we outline the verification and validation approach for LSST Data Management. In addition we outline some of the high level test milestones.

### 1.1 Objectives

We describes the test and verification approach for DM and describe 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 but leave that to dedicated test plans.

### 1.2 Scope

This provides the approach and plan for all od DM. It covers interfaces to DM but nothing outside of DM. This document will be updated in response to any requirements updates.

### 1.3 Assumptions

We will run large scale Science Validations. 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 broad validation and specifically *operaability* i.e. can we run this system everyday for a long period of time (years).

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

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

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

[7] **[LDM-502]**, Nidever, D., Economou, F., 2016, *The Measurement and Verification of DM Key Performance Metrics*, LDM-502, URL https://ls.st/LDM-502

[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 generated from the on-line Gaia acronym list:

| Acronym | Description |
|---|---|
| API | Application Programming Interface |
| CAM | CAMera |
| CU | Coordination Unit (in DPAC) |
| DM | Data Management |
| DMCCB | DM Change Control Board |
| DMLT | DM Leadership Team |
| DPAC | Data Processing and Analysis Consortium |
| DPC | Data Processing Centre |
| DRP | Data Release Production |
| ESA | European Space Agency |
| 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) |
| LCR | LSST Change Request |
| LSST | Large-aperture Synoptic Survey Telescope |
| NCSA | National Center for Supercomputing Applications |
| OPS | OPerationS |
| OSS | Operations Support System |
| QA | Quality Assurance |
| QC | Quality Control |
| SP | Software Product |
| SPR | Software Problem Report |
| SST | Space Surveillance Telescope |
| SUIT | Science User Interface Team |
| SVN | SubVersioN |
| TBD | To Be Defined (Determined) |
| WBS | Work Breakdown Structure |
| WISE | Wide-field Survey Explorer |

## 2 Test Items

The test items covered in this test plan are Data Management and its constituent components:

- All the product - from KT diagrams
- Interfaces
- Procedures like Data release

## 3 Roles and Reporting

Tester report issues through Jira, but what other mechanisms will be used?

Who directs OPS rehearsals .. ?

Reports on rehearsals .. issues and

Handling failures - time lines for fix.

> **Note**
>
> Not my section but a thought: System Engineering intends to capture commissioning tests through JIRA testing addons/plugins such as Kanoah. We could use some of these to capture functional tests for repeatability between operation rehearsals (they might be overkill for most tests though). –FE

> **Note**
>
> Note that downstream text refers to “Software Review Board”. We don’t have such an entity so we need to either identify an existing entity or define the constitution of that Board.

The Software Review Board will meet once a full run of all Test Cases has been performed, and subsequently after a complete run of all outstanding Test Cases.

**Note**

We don't actually have a software review board, not sure what the equivalent would be? is it an in-system, in-project or independent body?

### 3.1 Pass/Fail Criteria

Pass/Fail criteria vary slighty depending on the type of test being performed.

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.

### 3.1.1 Key Performance Metrics

**Note**

Given the incomplete 1:1 match between Key Performance Metrics listed in LDM-240 (which was a spreadsheet) and LSE-30 (aka OSS), we could theoretically have a situation where we pass our KPMs but fail an OSS metric. I think this is unlikely, but we would need to complete the OSS flowdown to be able to demonstate that to a skeptic. If we re-flowdown and come up with KPM 2.0s though, this section would stand as written. We just have to surrender the pedantic point that they wouldn't be "Key" at that point, they would be **the** performance metrics.

**Note**

There is an LCR open for flowing down some of the performance metrics from LSR to OSS.

Key Performance Metrics

# 4 Constraints and Limitations

**Note**

Wil: Describes the limitations and the constraints which apply to CU level tests of the system. lack of computing resources may mean that datasets are smaller or that full accuracy cannot be achieved. Explain what must be validated in the DPC tests.

- 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.

## 4.1 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

### 4.1.1 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).

**Note**

Flowdown is not complete, TJ is working on this.

### 4.1.2 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 eg. 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.

### 4.1.3 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. [TJ is working on this]. In particular, the non-science KPMs only exist in LDM-240 *(spreadsheet/old plan)*.

> **Note**
> While verification was part of the SQuaRE WBS we prepared a KPM verification plan at the request of System Engineering - LDM-502. This work is now being led by Wil now I guess?

## 4.2 Functional Requirements

Functional requirement are not explicitly called out as such. They are captured in LSE-61 (DMSR).

> **Note**
> When SQuaRE prepared the verification plan for SysEng, we were directed not to include functional requirements and limit ourselves to KPM. In general functional requirements are easy to verify by simply undertaking to perform the required functions in eg. operational rehearsals so maybe we could just say that?

### 4.3 Interfaces

There is an implicit, but not explicit, need to verify interfaces to other subsystems. The ICDs describing external interfaces are curated in Docushare Collection 5201.

> **Note**
>
> Integration used to be a Tucson role; I believe this is being led by the currently vacant Integration Scientist role? or whoever conducts the operation rehearsals?

> **Note**
>
> Internal interfaces: currently we have no definitions and hence they are not verifiable presumably. If we did, I would propose: I

## 5 Master Schedule

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

| Date/Freq | Location | Title, Description |
|---|---|---|
| Nightly | Amazon | **Nightly Tests**<br>Run all automated tests on all DM packages automatically. |
| Weekly | Amazon | **Integration tests**<br>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<br>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**<br>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**<br>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**<br>Validate AP alert generation stack performance on several DECam and HSC datasets. Begin continuous integration testing. |
| S18 | NCSA? | **AP system validation**<br>Validate AP alert distribution and mini-broker system fed by live or simulated live data. |
| 2018 | NCSA | **Spectrograph Data acquisition**<br>. . . Do we need a test BEFORE THIS? |
| 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). |
| 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. |
| 2020 | NCSA | **Operations Rehearsal Data Release (Commissioning Data)** |
| 2021 | NCSA | **Operations Rehearsal Data Release (Regular Data).** |

# 6 Validation Tools

### 6.1 Introduction

To evaluate the correctness of the generated data and the systems performances a set of tools may be developed or used. These tools will provide the means to facilitate the validation tasks. Following subsections describe the various tools that can used in the Data Managementvalidation (e.g. data comparison tools, analysis tools, etc).

### 6.2 Data Comparison Tools

This type of test tools are used to manage products in terms of:

- Comparison of a product generated during a test execution w.r.t. the relevant reference product
- Non regression verification comparing output products generated by different versions of the same system
- Measurement of quality degradation due to perturbed inputs

It allows:

- Product analysis
- Decoding of generated product allowing to read the most significant data of the product itself
- Visualisation of the values of a single selected field
- Apply an accuracy to the comparison
- Comparing specific parts of the products
- Filtering using flags values

### 6.3 Data Transformation Tools

These tools allow the data to be transformed to other formatted data.

### 6.4 Analysis Tools

Descriptions of the performance monitoring tools, profilers, test coverage programs... used in the Performance evaluation tests.
...

## 7 Unit and Integration Tests

### 7.1 Approach

Unit and Integration Tests will be automatically executed through the JUnit test framework. The descriptions of the test below are extracted from the test cases code and documentation.The results of Unit and Integration Test to be included in the Software Test Report will be generated automatically from the output of the execution of the tests by JUnit. A script will be provided to perform thees processing steps.

Module identification? (module tag in class header? mapping file?)

### 7.2 Test Coverage

Test coverage goal for unit and integration testing. Each class and public method shall have a JUnit test harness that may be labeled according to their purpose (e.g. I/O, individual class/method tests, software integration, data model integration etc.). Nominal and contingency tests should be clearly identified.

Interface coverage...

The tool [insert name of unit test coverage tool here] will be used to provide metrics on the code coverage by Unit Tests for Data Management and this metric will be provide in the Test Report.

### 7.3 Component and Integration Test Specification

TABLE 3: Components from LDM-148 with the test specifications to verify them.

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

# 8 Validation Tests

Validation of the system though Oerations Rehersals (and or end to end tests)

## 8.1 General strategy

Description of the general verification and validation strategy, decomposition into verification testing categories (e.g. science tests, SP external interface tests, algorithms interrelation and sequence). Assessed validation tests results shall be available over the software development duration: they are stored into SVN repository along with related input data, property-file, etc.

A subset of tests are run at DPC during software release qualification process, the results of DPC runs are compared with corresponding test outputs. During DPC integration tests, these assessed outputs will also allow to verify software non-regression.

## 8.2 Test Designs

### 8.2.1 Test Design DM-Data Management-SYS-X

#### 8.2.1.1 Objective

Explain the objective of this test design

#### 8.2.1.2 Features to be tested

- Component A
- Component B

#### 8.2.1.3 Features not to be tested

- Component C
- Component D

#### 8.2.1.4 Approach

Description of the approach to writing this test design

#### 8.2.1.5 Test Cases
List of test cases to be specified

| Test Case | Description |
|---|---|
| DM-Data Management-SYS-X-1 | Description of Validation Test |

## 8.3 Test Case Specification

### 8.3.1 Test Case DM-Data Management-SYS-X-1

#### 8.3.1.1 Testable Items
List the components to be tested in this test case

#### 8.3.1.2 Purpose
Explain the purpose of this test case

#### 8.3.1.3 Input Specification
Describe the inputs to this test (data, written procedures, etc.)

#### 8.3.1.4 Output Specification
Describe the outputs of this test

#### 8.3.1.5 Environment
Describe the environment (computing resources etc) required for this test.

#### 8.3.1.6 Inter-case dependencies
If this test in dependent on another test having been completed successfully (for input data for example), state that here.

#### 8.3.1.7 Test Procedure
Describe the procedure to be performed

#### 8.3.1.8 Test Verification
Describe how to verify if the test has been successful.

# 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 5). 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 5 (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

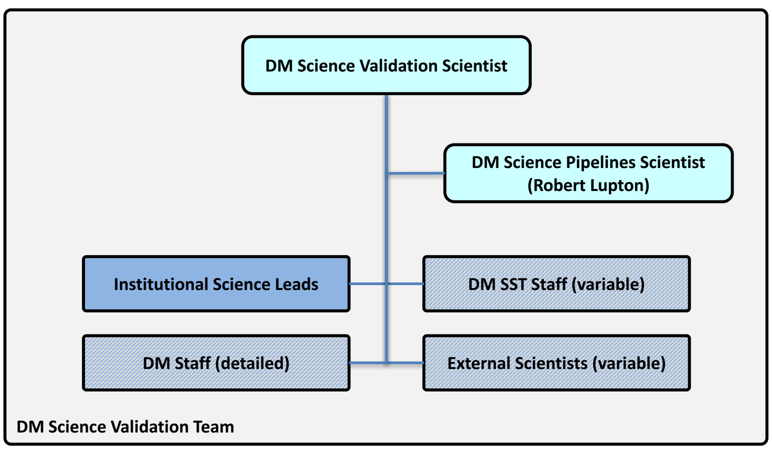


FIGURE 1: Organogram of the Data Management Science Validation Team. The group is chaired by the DM Science Validation Scientist, with the DM Science Pipelines Scientist and Institutional Science Leads making up the permanent membership. Depending on the SV activities being executed at any given time, 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 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.

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 5. 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.