LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

Data Management Test Plan

William O'Mullane, John Swinbank, Mario Juric, Frossie Economou, Gabriele Comoretto

LDM-503

Latest Revision: 2018-11-21

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
	2017-01-13	First draft	William O'Mullane
1.0	2017-06-30	First approved release.	William O'Mullane
1.1	2017-07-04	Minor cleanups for review. Approved in RFC-358.	W. O'Mullane
1.2	2017-07-19	Milestone alignment with plan	W. O'Mullane
	2018-06-18	Milestone alignment with PMCS and derived from it	J.D. Swinbank
1.3	2018-06-18	Approved in RFC-492	J.D. Swinbank
1.4	2018-07-17	Correct test specifications for LDM-503-04, 04b, 05, 08b. Link Pipelines releases to LDM-564. Approved in RFC-501.	J.D. Swinbank
1.5	2018-11-13	Include updated document tree. Update milestones for month end September 2018. Describe approach to managing tests in Jira.	W. O'Mullane; J.D. Swinbank; G. Comoretto
XXX	20XX-XX-XX	Updated description and link to test plan & report for LDM-503-09a.	J.D. Swinbank

Document source location: `https://github.com/lsst/LDM-503`

Contents

Data Management Test Plan

1 Introduction

This document provides an introduction to and overview of the approach to verification and validation which bas been adopted by the LSST Data Management Subsystem. Broadly, this approach consists of three aspects:

- *Verification* that the Data Management system as delivered meets the requirements placed upon it;
- *Validation* that the system as delivered meets the needs of the scientific community;
- *Rehearsing* the sustained operation of the system in operational scenarios.

This documentation describes how LSST Data Management is addressing each of these three requirements, and describes a series of high-level milestones and the accompanying test schedule. In addition, it briefly discusses the software development infrastructure that has been developed to support all three of these aspects of testing.

1.1 Objectives

We describe the test and verification approach for Data Management and describe various constraints and limitations in the testing to be performed. We also describe the program of rehearsals which will be undertaken to demonstrate the sustained operation of the Data Management system, and the validation exercises which will be performed on the partially and fully integrated system. We do not describe all tests in detail; those are described in dedicated test specifications for major components of Data Management.

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 verification exercisess in order to demonstrate the system's end-to-end capability against its design specifications. A large amount of informal science verification and validation will be done in the the teams and documented in technical notes; in this test plan we are looking for verification of the broader system, demonstration of its *operability* — i.e. whether it can be run every day for the 10 year planned survey with a reasonable level of operational support – and to validate its capability to meet the scientific expectations of the community.

1.4 Applicable Documents

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

LPM-55	LSST Quality Assurance Plan
LDM-148	DM Architecture
LDM-294	DM Project Management Plan
LDM-639	DM Acceptance Test Specification

1.5 References

[1] **[LDM-555]**, Becla, J., 2017, *Data Management Database Requirements*, LDM-555, URL https://ls.st/LDM-555

[2] **[LDM-533]**, Bellm, E.C., 2017, *Level 1 System Software Test Specification*, LDM-533, URL https://ls.st/LDM-533

[3] **[LDM-538]**, Butler, M., Parsons, J., Gower, M., 2018, *Raw Image Archiving Service Test Specification*, LDM-538, URL https://ls.st/LDM-538

[4] **[LSE-79]**, Claver, C., The LSST Commissioning Planning Team, 2017, *System AI&T and Commissioning Plan*, LSE-79, URL https://ls.st/LSE-79

[5] **[LSE-29]**, Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2017, *LSST System Requirements (LSR)*, LSE-29, URL https://ls.st/LSE-29

[6] **[LSE-30]**, Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2018, *Observatory System Specifications (OSS)*, LSE-30, URL https://ls.st/LSE-30

[7] **[LDM-540]**, Dubois-Felsmann, G., 2018, *LSST Science Platform Test Specification*, LDM-540, URL https://ls.st/LDM-540

[8] **[LSE-61]**, Dubois-Felsmann, G., Jenness, T., 2018, *LSST Data Management Subsystem Requirements*, LSE-61, URL https://ls.st/LSE-61

[9] **[LDM-554]**, Dubois-Felsmann, G., Ciardi, D., Mueller, F., Economou, F., 2018, *Science Platform Requirements*, LDM-554, URL https://ls.st/LDM-554

[10] **[LDM-639]**, Guy, L., 2018, *DM Acceptance Test Specification*, LDM-639, URL https://ls.st/LDM-639

[11] **[LDM-148]**, Lim, K.T., Bosch, J., Dubois-Felsmann, G., et al., 2018, *Data Management System Design*, LDM-148, URL https://ls.st/LDM-148

[12] **[LDM-552]**, Mueller, F., 2017, *Qserv Software Test Specification*, LDM-552, URL https://ls.st/LDM-552

[13] **[LDM-564]**, O'Mullane, W., Economou, F., Jenness, T., Loftus, A., 2018, *Data Management Software Releases for Verification/Integration*, LDM-564, URL https://ls.st/LDM-564

[14] **[LDM-294]**, O'Mullane, W., Swinbank, J., Jurić, M., DMLT, 2018, *Data Management Organization and Management*, LDM-294, URL https://ls.st/LDM-294

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

[16] **[DMTR-111]**, Swinbank, J., 2018, *LVV-P15 Test Plan and Report*, DMTR-111, URL https://ls.st/DMTR-111

[17] **[LDM-534]**, Swinbank, J.D., 2017, *Level 2 System Software Test Specification*, LDM-534, URL https://ls.st/LDM-534

1.6 Definitions, Acronyms, and Abbreviations

Acronym	Description
ATM	Adaptavist Test Management
CCB	Change Control Board
CI	Configuration Item
DAC	Data Access Center
DAQ	Data AcQuisition (system)
DAX	Data Access Services
DM	Data Management
DMCCB	DM Change Control Board
DMS	Data Management Sub-system
DRP	Data Release Production
EFD	Engineering Facilities Database
EPO	Education and Public Outreach
FITS	Flexible Image Transport System
HSC	Hyper Suprime-Cam
I&T	Integration and Test
ID	Identifier (Identification)
ISR	Instrument Signal Removal
JIRA	issue tracking product (not an acronym, but a truncation of Gojira, the Japanese name for Godzilla)
LPM	LSST Project Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSP	Low System Priority
LSST	Large Synoptic Survey Telescope
NCSA	National Center for Supercomputing Applications
OPS	Operations
PDAC	Prototype Data Access Center
PMCS	Project Management Control System
PSF	Point Spread Function
QA	Quality Assurance
QC	Quality Control
Qserv	Proprietary LSST Database system
RFC	Request for Comments
SPR	Software Problem Report

SUIT	Science User Interface and Tools
UX	User Experience
VCD	Verification Control Document
WCS	World Coordinate System
WISE	Wide-field Survey Explorer
p	pico; SI units prefix for 1E-12
s	second; SI unit of time

2 Verification Tests

We regard the system as being successfully completed when all of the high level requirements placed upon it, as defined in LSE-61 — the *Data Management System Requirements* — have been verified. The approach which will be taken to verifying each requirement is described in LDM-639, the *DM Acceptance Test Specification*. This test specification covers all aspects of the tests, as described in Section 2.3. Any given requirement may have multiple test cases associated with it in the specification, and these tests will be phased to account for incremental delivery depending on the need for certain functionality at a specific time. We anticipate that this phasing will align with the priorities assigned to requirements in LSE-61 — that is, that high priority requirements will be verified first — but this is not, in itself, required.

In addition to the high level requirements on the overall Data Management system, lower-level requirements documents describe requirements placed upon specific parts of the system (for example, LDM-554 provides requirements on the LSST Science Platform, and LDM-555 on the DM database system). Each of these requirements documents is accompanied by a test specification (LDM-540 and LDM-552 in the case of the previous examples; see also Table 1), with the relationship between them being the same as for the high-level requirements. These lower-level requirements documents and test specifications are defined on an as-needed basis: some Data Management components are adequately specified and tested by the high-level documentation.

In general, in order to avoid duplication of test definition and execution, when a requirement is flowed down to lower-level requirements, and therefore tested at lower level, the high level test activity may be just an inspection. Detailed high level test cases are required if an LSE-61 requirement is not decomposed into lower-level component requirements.

Although individual test cases may be executed at any time, it is anticipated that major testing campaigns will be undertaken to demonstrate the successful completion of major milestones in the Data Management construction effort. The schedule for these milestones is shown in Section 5, while Section 6 provides further details as to the contents of each one. In addition, each low level component owner can define specific test campaigns to verify compliance with the relevant requirements. For example, such a test campaign may be associated with a software release made for a specific purpose.

2.1 Managing and Reporting Test Execution

As described above, requirements and test specifications are provided in baselined documents. These documents provide curated views on the Jira *LSST Verification and Validation* project which underlies the LSST-wide test effort. The Jira system provides "test scripts" that testers will follow when carrying out tests, and tracks information about test execution and results. This information enables us to generate reports on the execution of each test, and ultimately to build a Verification Control Document (VCD; see Figure 1). The VCD will provide the verification status of each DM requirement (in terms of the fraction of test cases pertaining to that requirement which have been successfully executed).

2.2 Components Under Test

The components of the DM system are outlined in LDM-294 and described in detail in LDM-148; a summary is shown in Figure 2. The test specifications covering these components are shown in Table 1, but note that, at time of writing, the document tree is being refactored and document numbers are not currently available for all components.

In addition, the test plan presented in this document covers:

- The external interfaces between Data Management and other LSST systems. These are described in DocuShare collection 5201. These verification activities will contribute to the global coverage of LSST requirements, but will not be included in the generation of the DM VCD.
- Operational procedures, such as the Data Release Production process. These are addressed by means of *operations rehearsals* which constitute some of the level two mile-

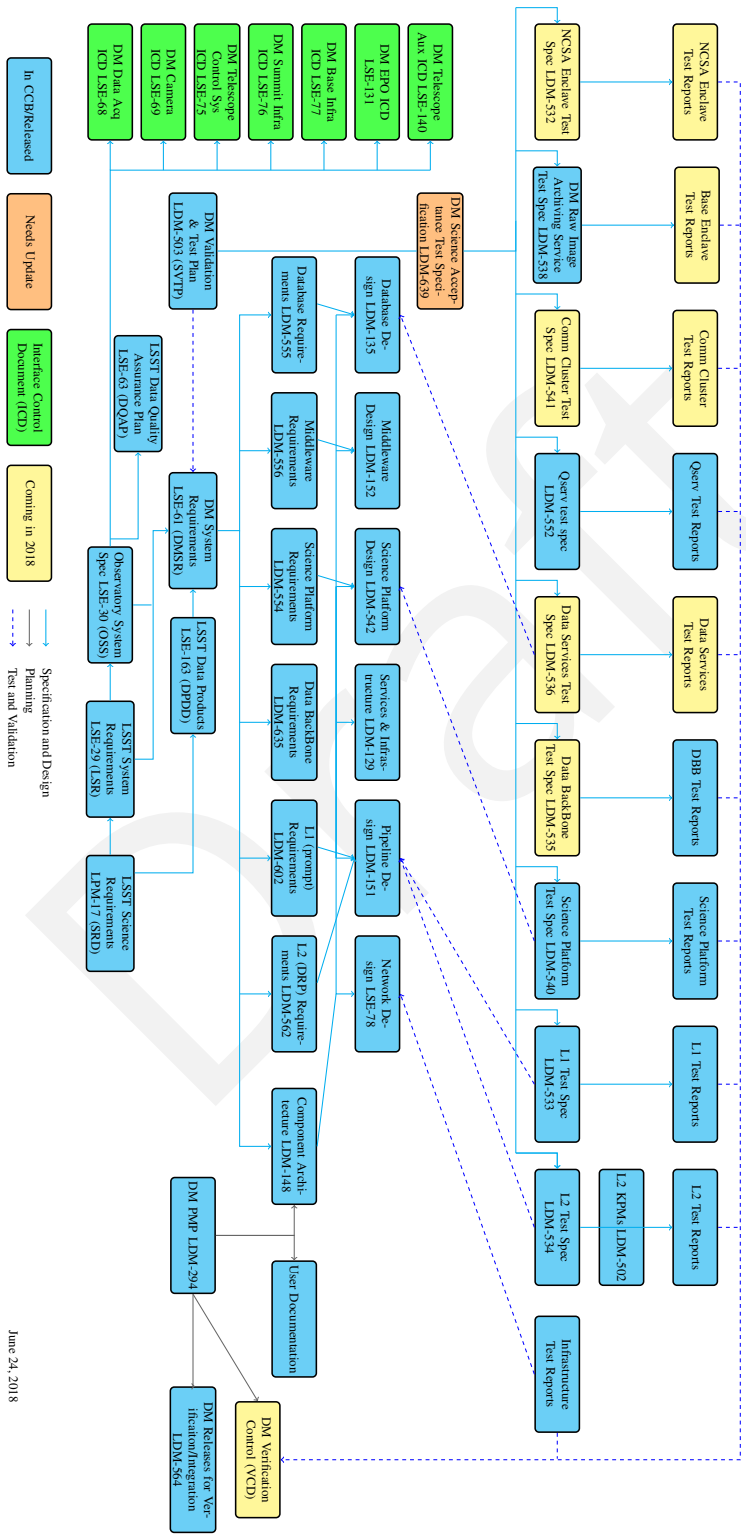

FIGURE 1: The Data Management document tree.

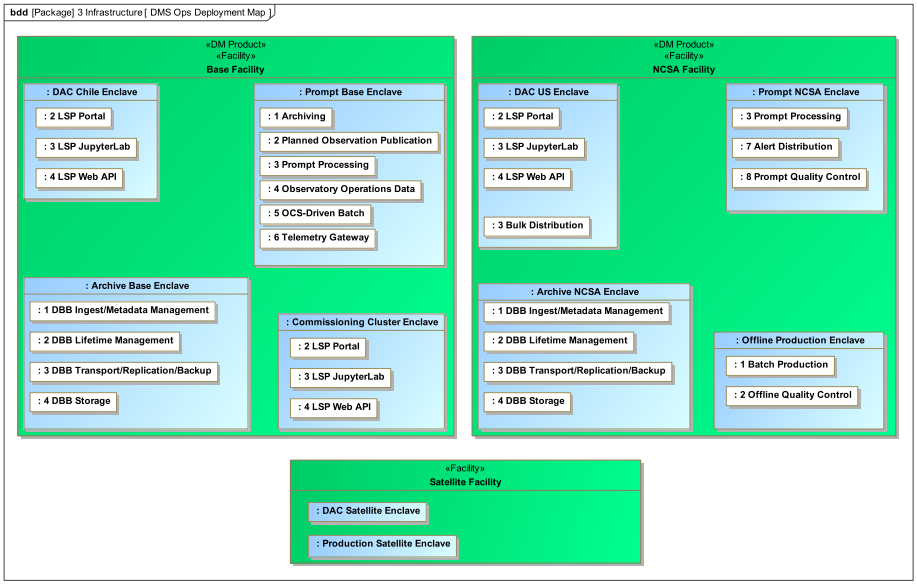

FIGURE 2: DM components as deployed during Operations. For details, refer to LDM-148.

stones.

Table 1: Requirements and test specifications for components of the Data Management system. Most components are covered by the high-level requirement and specification documents LSE-61 and LDM-639; those for which lower-level documents are available are listed individually. A cyan background indicates that a test specification is currently available; yellow, that one is being drafted at time of writing; orange, that the existing test specification is under revision.

	Specification	
Component	**Requirement**	**Test**
Data Backbone	LDM-635	LDM-535
LSP Services	LDM-554	LDM-540
Alert Distribution service	LDM-638	
Archiving service	LDM-638	LDM-538

Prompt Processing service	LDM-638	LDM-533
Alert Distribution Software		LDM-533
Alert Production Software	LDM-602	LDM-533
DR Production Software	LDM562	LDM-534
Distributed Database (Qserv)		LDM-552
Commissioning Cluster Enclave		LDM-541
DAC Enclaves		LDM-539
Offline Production Enclave (NCSA)		LDM-532
Data Management Acceptance	LSE-61	LDM-639

The tests associated with each milestone may encompass more than one component of the DM system. The relevant components must therefore be included in the description of the milestone in Section 6.

2.3 Test Approach Overview

This section gives an overview of the approach, facilities and documents involved in the verification process.

2.3.1 Tools

Properly understanding the test management system requires some familiary with the tooling involved.

MagicDraw

MagicDraw[1] is the standard requirements modeling tool in use by LSST; it is where all requirements are ultimately defined. The LSST Systems Engineering team use MagicDraw to track verification of the entire LSST system; it is therefore imperative that all results generated by DM are collected here.

Jira and Adaptavist Test Management

Jira[2] is the issue tracking and management system in use across LSST. Adaptavist Test

[1] https://www.nomagic.com/products/magicdraw
[2] https://jira.lsstcorp.org/

Management[3] augments Jira with the capability to manage verification activities.

Syndeia

Syndeia[4] is used to synchronize the MagicDraw and Jira systems.

Extraction Scripts

The LSST team has written a set of scripts to extract information from Jira and format it as test specifications, test plans and reports, and the verification control document.

In general, those who wish to understand the current status of DM should need only to interact with baselined test specifications, test reports and the verification control document: use of the tools described above should not be necessary.

Product owners and individuals carrying out tests will interact with Jira.

Only members of the DM Systems Engineering Team (LDM-294) will interface directly with MagicDraw.

2.3.2 Requirements and Test Objects

This section provides an overview of the key concepts and vocabulary used in the test system.

Requrement

Requirements are defined in MagicDraw and then synchronized with Jira. For change control and for distribution to the wider project, they are extracted from MagicDraw to baselined documents. The level of change control applied to each requirement depends on which document it appears in: high level requirements appear in LSE-61 and are subject to project-level change control, while lower level requirements appear in LDM-series documents and are managed by the DM-CCB.

Verification Element

Each requirement is decomposed into one or more verification elements. A verification element is an aspect of the reuqirement which can be independently tested. Verification elements are created and updated in MagicDraw and synchronized with Jira; in Jira, it appears as a normal issue with a specific type.

[3] https://www.adaptavist.com/atlassian-apps/test-management-for-jira/
[4] http://intercax.com/products/syndeia/

Test Case

A test case is the definition of a procedure to be executed to test the related verification elements. A single test case may test many verification elements. Test cases are represented as special objects in Jira provided by the ATM system.

Test Cycle

A test cycle is list of test cases to be carried out in a particular order under a specified environtment to achieve some specific goal. Each test cycle may contain only one instance of a particular test case: re-executing a test case with (e.g.) a different configuration must be done in a separate cycle. Test cycles are represented as special objects in Jira provided by the ATM system.

Test Plan

A test plan defines the overall plan for achieving some particular goal, such as verifying a software release or completing a milestone (Section 6). Each test plan may include several test cycles. Test plans are represented as special objects in Jira provided by the ATM system.

Test Player

The Test Player is an interactive Jira tool which provides the tester (Section 2.4) with instructions and collects responses while a test case is being executed.

Software Problem Report

A Software Problem Report (SPR) describes a software failure or bug encountered when executing a test case. SPRs are represetned by Jira tickets in the “Data Management” Jira project.

Deviation

When testing establishes that it is impossible to meet a particular requirement, a Jira issue of type “Deviation” is filed. This ultimately represents a request to change the baseline to remove or relax the requirement in question. Such a change can only be made with the approval of the relevant (Project or DM) Change Control Board.

Test Campaign

A test campaign is the sum of all activities needed to plan, execute and report the testing carried out with a specific goal in mind (e.g. addressing a particular milestone). All information relevant for a test campaign is collected in the test plan and related test cycle(s). Each test plan and report refers to a specific test campaign.

More details on the various Jira objects and detailed instructions on their use are available on Confluence[5].

Workflows for the different types of objects are described in project-level Systems Engineering documentation[6].

2.3.3 Test Documents

Though all test information is contained in MagicDraw and Jira, it is important to have baselined test documentation in Docushare.

The **Test Specification** for a component collects all of the test cases that cover that particular component. Test specifications are subject to approval by the responsible change control board, and therefore from part of the project baseline. However, they are also living documents: as test cases are added or updated in Jira, new editions of the document will be produced, basedlined and provided through Docushare.

Test Specifications consist of a mixture of material which are directly written and curated in GitHub by the corresponding component owner, and sections which are automatically generated from the contents of Jira.

The **Test Plan and Report**[7] describes all the information related to a particular test cycle — that is, they describe the contents of the test cycle and the results of executing it. Since test plans and reports describe the results of a particular campaign, updates are limited to minor corrections (for example, to spelling). A new test campaign, even one which repeated the same test plan, would result in the creation of a new document to describe the new results.

2.3.4 Approval Procedure

Test specifications are part of the project baseline. As such, they must be approved by the relevant change control board (project-level for LSE-handled documents such as LSE-61; DM-level for LDM-handles).

[5] https://confluence.lsstcorp.org/display/DM/DM+Test+Approach
[6] `https://confluence.lsstcorp.org/display/SYSENG/LSST+Verification+Architecture`
[7] For historical reasons, test plan and report documents use the handle “DMTR”.

New test cases, which are not yet part of an accepted specification, or test cases which have been updated since the relevant specification was baselined, will be in the "draft" status. When the relevant specification is accepted, they should be moved to "approved" status, at which point the specification document is regenerated and placed in Docushare.

Test cases may also be removed from the baseline in an analagous process. These test cases should be marked as "deprecated".

Test plan and report documents are not change controlled (indeed, per §2.3.3 they should not be changed). However, they should be approved by whoever is responsible for requesting that the test be carried out. For high-level milestones (i.e., those listed in Section 6), that should be taken to mean the combination of the DM Project Manager and the DM Subsystem Scientist. Note that approval for these documents must be sought in two stages:

- The test plan should be approved before the test campaign is carried out;
- The results should be approved after the campaign has been completed.

2.3.5 The DM VCD

A global, project-wide verification control document will be derived from MagicDraw by the Systems Engineering team. However, per Section 2.1, a Data Management-specific VCD will also be provided. This will show the test coverage of each requirement its relationship to test cases and their execution. A conceptual example of the structure of the VCD is given in Table 2.

2.4 Roles and Personnel

Each test case is assigned an *owner*, who is responsible for defining and maintaining it.

Executing the test case is the responsibility of the *tester*, who may be different from the owner. A given test case may be executed as part of multiple test campaigns; each time, it may be the responsibility of the a different tester. Test cases are executed following the script provided in the Jira "LSST Verification and Validation" project (Section 2.3).

Table 2: An example of the structure of the Verification Control Document. Identifiers are intented to be illustrative only, and may not correspond to existing documents, requirements, verification elements, test cases, or test plans and reports. In this example, DMS-REQ-001 is a high-level DM requirement (described in LSE-61) which is associated with two verification elements: DMS-REQ-0001-V-01, which is tested by test cases LVV-T1 and LVV-T2, and DMS-REQ-0001-V-02, which is tested by the single test case LVV-T2. DMS-LSP-REQ-002 is a lower level requirement falling under DM change control (iit appears in a document with an "LDM" handle). It is covered by a single verification element, which is tested by two test cases.

Requirement	Verification Element	Test Case	Run Date	Test Status
DMS-REQ-0001 LSE-61	DMS-REQ-0001-V-01 LVV-1	LVV-T1 LDM-639	2011-11-11 DMTR-999	**Passed**
		LVV-T4 LDM-639	2011-11-11 DMTR-999	**Passed**
	DMS-REQ-0001-V-02 LVV-1	LVV-T2 LDM-639		**Not run**
DMS-LSP-REQ-0002 LDM-554	DMS-LSP-REQ-0002-V-01 LVV-3	LVV-T3 LDM-639	2011-11-11 DMTR-999	**Failed**
		LVV-T5 LDM-639		**Not run**

Testers have to report the test execution details into the corresponding fields provided in Jira by the "test player", so they can be used to generate test reports. The information captured in Jira will also be used to populate the Verification Control Document (see Section 2).

2.5 Success Criteria

Test cases will sometimes fail. A test case may be re-run several times until it passes, but the tester must log an explanation than indicates that any failures were understood (e.g. they were due to a fault that was fixed) or repeated sufficient times to ensure that passing the test was not a transient success.

Issues which cannot be resolved by the tester in the course of carrying out the test will be reported as "Software Problem Reports" (SPRs) through the Data Management ticketing system (the Jira "Data Management" project at the time of this document revision).

The SPR describes an issue with the component being tested. In some cases, that issue may simply be a "bug", for which a fix can be implemented as part of regular Data Management development. The test case can then be re-executed successfully.

In other cases, the SPR may be raised because the component under test is simply incapable of hitting the requirements placed upon it: it is not sufficiently fast, or accurate, or is in some other way deficient. Ultimately, it may be impossible (either due to resource constraints, or simply because the requirement is unrealistic) to resolve the SPR in such a way that the requirement can be met. In this case, an issue of type "Deviation" may be filed: this represents a request to change or relax the requirements. Product owners are responsible for reviewing SPRs relating to components which they are responsible for and filing Deviations when appropriate.

A test case cannot be regarded as passing while there are open SPRs preventing its execution. If the SPRs cannot be resolved in a timely fashion, the test case should be recorded as a failure. It may be re-executed as part of a fresh test campaign when the SPRs have been resolved.

2.6 Constraints and Limitations

2.6.1 Procedural and Technical Limitations

- The Data Management system must be verified before the complete LSST system can be completed. Verification is therefore carried out using precursor datasets [8], simulated data, and — where available — with engineering and pre-release data from the as yet incomplete LSST system.
- 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-epoch fitting may only be modeled, since they have not yet been implemented; operational rehearsals are done without the factory LSST workflow system; etc.

2.6.2 Requirements Traceability Constraints

The Data Management verification plan is based entirely on requirements captured in the DM System Requirements (LSE-61). It does not refer to higher level requirements documentation, such as the LSST System Requirements (LSE-29) or the Observatory System Specifications (LSE-30); rather, we assume that all higher level requirements have been correctly flowed down to DM. In practice, the Systems Engineering team continues to refine the flow-down of higher level requirements and issue updates to LSE-61; this test plan must both anticipate and be responsive to those updates.

3 Operations Rehearsals

The operability of the Data Management system is demonstrated through a series of operations rehearsals. Like verification tests , these rehearsals correspond to high level DM milestones (Section 5), and involve carrying out a specific set of activities under controlled conditions. As such, many of the considerations described in Section 2 also apply to rehearsals. However, the aim of the rehearsal is not to verify that the performance of the Data Management system meets some requirement, but to verify that it can be integrated and operated successfully, and to demonstrate and validate operational procedures.

[8] e.g. from Hyper Suprime-Cam; Section 6.4

The current schedule calls for six rehearsals to be carried out to test different aspects of the system (for example, one rehearsal addresses nightly operations, and another the production and curation of a data release). At time of writing, the activities to be undertaken as part of each operations rehearsal are currently being detailed. This schedule and scope of each exercise will be designed to align with the LSST Commissioning Plan (LSE-79).

Operations rehearsals require an *Rehearsal Coordinator* to oversee the process. This is a distinct role from that of the testers (Section 2.4, since they are (by definition) carrying out their operational roles during the rehearsal. For example, the rehearsal may not be directed by the Operations Manager, since that person has a major role in the rehearsal. An individual not involved in the rehearsal itself will be identified to perform this function.

4 Science Validation

4.1 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.

4.2 Schedule and Execution

4.2.1 Schedule

Unlike the verification and rehearsal activities, which correspond to high level milestones, validation 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. 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.

4.2.2 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 fashion. These are fed back to the development teams for productizing and incorporation into the automated QC systems.

4.3 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.

4.4 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.

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

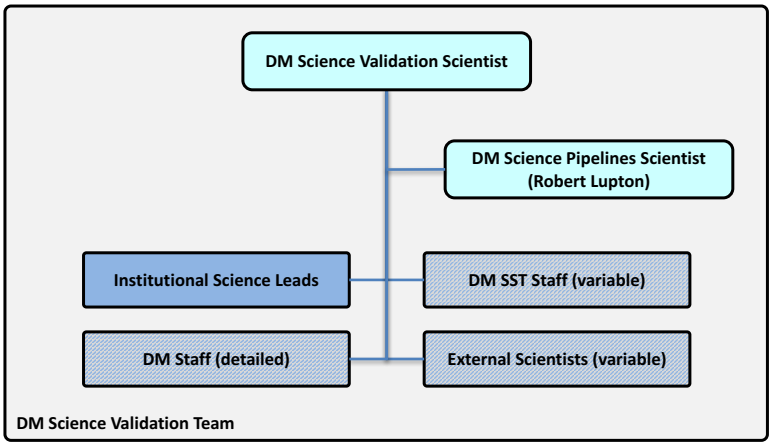

FIGURE 3: Organogram of the Data Management Science Validation Team. The group is chaired by the DM Science Validation Scientist, with the DM Science Pipelines Scientist and Institutional Science Leads making up the permanent membership. Depending on the SV activities being executed at any given time, 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.

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.

4.4.1 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 velocity 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.

5 Master Schedule

The schedule for testing the system until operations commence (currently 2022) is outlined in Table 3. Tests named following the pattern “LDM-503-NN” (LDM-503-01, -02, etc) correspond

to major [9] milestones of the DM project. These tests are closely tied to major integration events for the overall LSST system, as shown in Figure 4. This schedule also includes nightly and weekly tests which are executed throughout the construciton period by the continuous integration (CI) system (see `developer.lsst.io`).

Table 3: List of High Level integration tests for DM

ID	Date	Location	Title
LDM-503-NLY	Nightly	CI System	Nightly integration tests
LDM-503-WLY	Weekly	CI System	Weekly integration tests
LDM-503-01	2017-11-30	NCSA	Science Platform with WISE data in PDAC
LDM-503-02	2017-11-30	NCSA	HSC reprocessing
LDM-503-03	2017-11-30	NCSA	Alert generation validation
LDM-503-04	2018-06-01	NCSA	Aux Tel DAQ integration functionality test
LDM-503-04b	2018-06-01	NCSA	Aux Tel DAQ interface Integration Verification and Spectrograph Operations Rehearsal
LDM-503-05	2018-07-02	NCSA	Alert distribution validation
LDM-503-08b	2018-09-06	NCSA	Small Scale CCOB Data Access
LDM-503-06	2018-11-30	NCSA	DM ComCam interface verification readiness
LDM-503-07	2018-11-30	NCSA	Camera data processing
LDM-503-09	2018-11-30	NCSA	Ops rehearsal for commissioning #1
LDM-503-09a	2018-11-30	NCSA	Pipelines Release Fall 2018
LDM-503-08	2019-01-04	NCSA	Spectrograph data acquisition
LDM-503-10	2019-09-25	NCSA	DAQ validation
LDM-503-10b	2019-09-25	NCSA	Large Scale CCOB Data Access
LDM-503-11	2019-09-25	NCSA	Ops rehearsal for commissioning #2
LDM-503-11b	2019-09-25	NCSA	Pipelines Release Fall 2019
LDM-503-11a	2019-09-30	NCSA	ComCam Ops Readiness
LDM-503-12	2019-11-26	NCSA	Ops rehearsal for commissioning #3
LDM-503-12a	2020-07-27	NCSA	LSSTCam Ops Readiness
LDM-503-13a	2020-10-30	NCSA	Pipelines Release Fall 2020
LDM-503-13	2020-11-30	NCSA	Ops rehearsal for data release processing #1 (ComCam data)
LDM-503-14	2020-11-30	NCSA	DM Readiness for Science Verification
LDM-503-15a	2021-10-28	NCSA	Pipelines Release Fall 2021

[9] Level 2, in the parlance of LDM-294.

LDM-503-15	2021-11-29	NCSA	Ops rehearsal for data release processing #2
LDM-503-16	2022-02-28	NCSA	Ops rehearsal for data release processing #3
LDM-503-17a	2022-08-31	NCSA	Final Pipelines Delivery
LDM-503-17	2022-09-30	NCSA	Final operations rehearsal

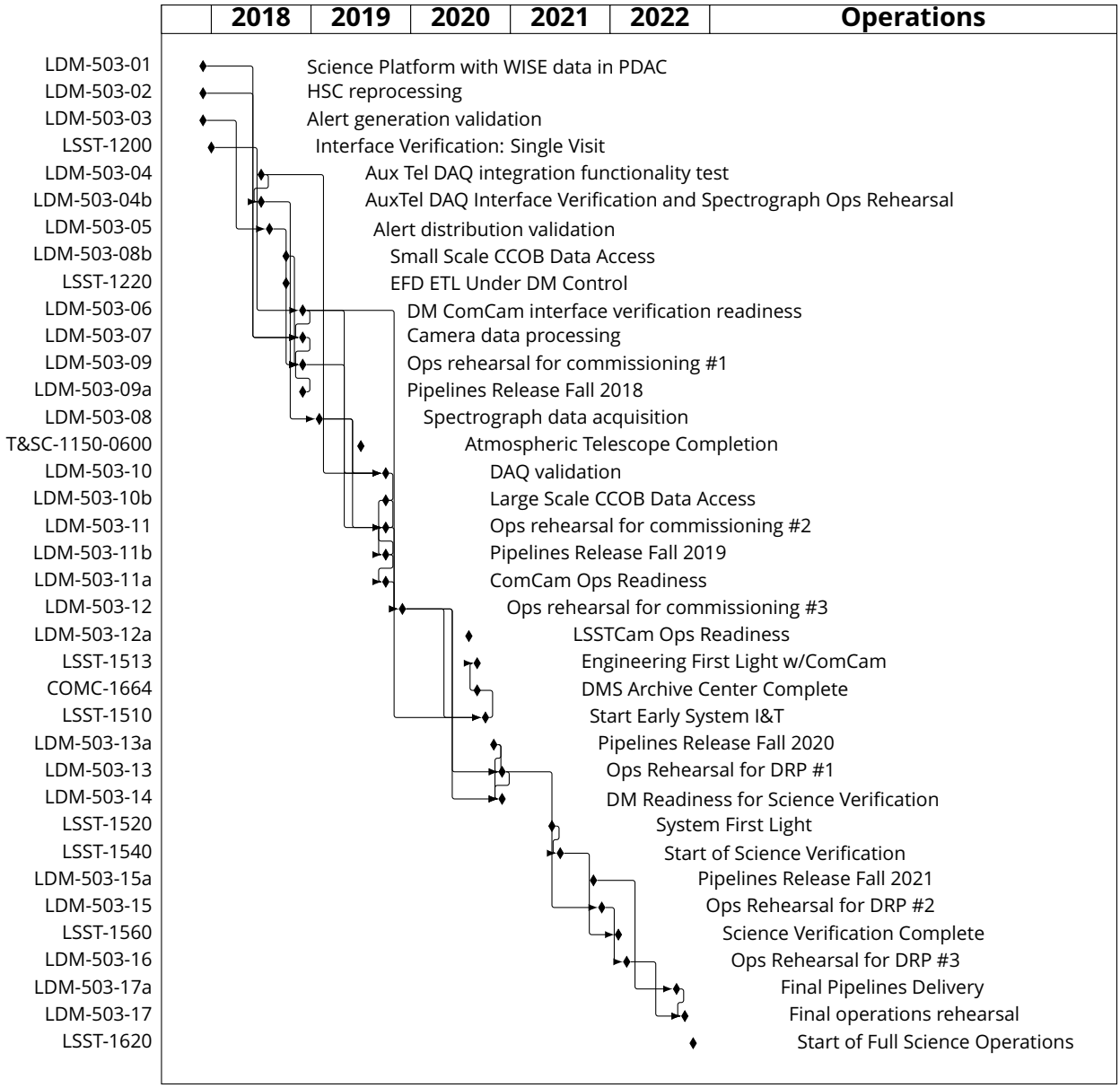

FIGURE 4: DM major milestones (LDM-503-x) in the LSST schedule.

6 Major DM Milestones

6.1 Nightly software integration tests and releases (LDM-503-NLY)

Nightly throughout construction, the DM continuous integration system (see `developer.lsst.io`) performs a complete rebuild of all major components of the Data Management codebase, executes their associated unit tests, and runs a series of automated small-scale integration tests. Failures are logged, and the DM team notified. The resulting build artefacts are packaged for release as a "nightly build".

6.2 Weekly software integration tests and releases (LDM-503-WLY)

Nightly throughout construction, the DM continuous integration system (see `developer.lsst.io`) performs a complete rebuild of all major components of the Data Management codebase, executes their associated unit tests, and runs a series of automated small-scale integration tests. Failures are logged, and the DM team notified. The resulting build artefacts are packaged for release as "weekly build", and automatically made available for developer use on shared hardware.

6.3 Science Platform with WISE data in PDAC (LDM-503-01)

6.3.1 Specification

This test will be executed following the procedure defined in LDM-540 v1.0 §3.1 (LSP-00).

6.3.2 Description

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

6.3.3 Comments

From DAX: need to be clear about which WISE datasets are to be loaded: the data wrangling effort required to download, inspect, convert, partition, and load each additional dataset is

cumulatively non-trivial for DAX

6.4 HSC reprocessing (LDM-503-02)

6.4.1 Specification

This test will be executed following the procedure defined in LDM-534 v3.1 §3.1 (DRP-00).

6.4.2 Description

Validate the data products with the LSST stack match or improve upon HSC products. Validate the ops platform in NCSA, including installing the stack, starting & stopping production. Generate a validation data set for weekly integration and other tests.

6.5 Alert generation validation (LDM-503-03)

6.5.1 Specification

This test will be executed following the procedure defined in LDM-533 v1.1 §3.1 (AG-00).

6.5.2 Description

Validate the alert generation stack performance on several DECam & HSC datasets.

6.5.3 Comments

Validate the alert generation stack performance on several DECam and HSC datasets. "Stack" is probably ill-defined here: is this simply testing science logic, or are we going after a wider integration exercise?

6.6 Aux Tel DAQ integration functionality test (LDM-503-04)

6.6.1 Specification

This test will be executed following the procedure defined in LDM-538 v1.0 §4.1 (RAS-00-00).

6.6.2 Description

Demonstrate the writing of a well-formed raw image with proper headers.

6.6.3 Comments

A minimal system that can be used to generate a raw image and then create a FITS file with headers.

6.7 Aux Tel DAQ interface Integration Verification and Spectrograph Operations Rehearsal (LDM-503-04b)

6.7.1 Specification

This test will be executed following the procedure defined in LDM-538 v1.0 §4.4 (RAS-00-20).

6.7.2 Description

Demonstrate the integration of writing the well formed FITS file with the permanent record of the survey. The archive or permanent record of the survey includes site file systems with multiple copies, immutable data and entries in the consolidated database for later retrieval and search needs.

6.7.3 Comments

A minimal system that can archive simulated images from the AuxTel DAQ and demonstrate that they can be retrieved.

6.8 Alert distribution validation (LDM-503-05)

6.8.1 Specification

This test will be executed following the procedure defined in LDM-533 draft version of 2018-07-16[10] §§4.7–4.9 (test cases LVV-T216, LVV-T217, LVV-T218).

6.8.2 Description

Validate alert distribution system and mini-broker fed by live or simulated live data.

6.9 Small Scale CCOB Data Access (LDM-503-08b)

6.9.1 Specification

This test will be executed following the procedure defined in LDM-538 v1.0 §4.2 (RAS-00-05).

6.9.2 Description

Demonstrate the integration of the CCOB with the permanent record of the survey. The archive or permanent record of the survey includes site file systems with multiple copies, immutable data and entries in the consolidated database for later retrieval and search needs.

6.9.3 Comments

A minimal system that can archive images and metadata from the CCOB and demonstrate that they can be retrieved.

6.10 DM ComCam interface verification readiness (LDM-503-06)

6.10.1 Specification

The execution procedure for this test is currently unspecified.

[10] Git revision `4291d80`.

6.10.2 Description

Test the operation of the DM system prior to ComCam availability in Tucson.

6.10.3 Comments

"The DM system" should use some further definition: what do we want to test here? Data flow from ComCam to the Data Backbone, or science processing of ComCam data? Note the LSE-79 requirements for DM services in support of ComCam in table 8. They're required by Nov 2019/Feb 2020; it may be more appropriate to test some of them at a later date?

6.11 Camera data processing (LDM-503-07)

6.11.1 Specification

This test will be executed following the procedure defined in Will be covered by LDM-534; detailed test specification still under development..

6.11.2 Description

This test will demonstrate that camera test stand data can be made available and processed through the "Butler" data access abstraction

6.12 Ops rehearsal for commissioning #1 (LDM-503-09)

6.12.1 Specification

The execution procedure for this test is currently unspecified.

6.12.2 Description

Test how the system will run during commissioning.

6.12.3 Comments

Focus on ISR: we should test whatever we have available. See LSE-79 for a list of requirements.

6.13 Pipelines Release Fall 2018 (LDM-503-09a)

6.13.1 Specification

This test will be executed following the procedure defined in DMTR-111.

6.13.2 Description

Major Science Pipelines release in support of activities in late 2018 and early 2019. The contents of the release are described in LDM-564. The release will be accompanied by a characterization report.

6.14 Spectrograph data acquisition (LDM-503-08)

6.14.1 Specification

The execution procedure for this test is currently unspecified.

6.14.2 Description

Demonstrate that we can acquire (and process, depending on capabilities at time of test execution) data from the Spectrograph.

6.15 DAQ validation (LDM-503-10)

6.15.1 Specification

The execution procedure for this test is currently unspecified.

6.15.2 Description

There is a project milestone that DAQ/DM/Networks are available late 2019. We will run a full scale data acquisition test in to show this is ready.

6.16 Large Scale CCOB Data Access (LDM-503-10b)

6.16.1 Specification

The execution procedure for this test is currently unspecified.

6.16.2 Description

Demonstrate the ability to read data from the CCOB with 21 rafts, store at the Data Facility and make it available on some cluster for analysis.

6.16.3 Comments

This was added as part of DM-13073 to show we can process optical bench data specifically CCOB 21 raft setup. This extends LDM-503-8b.

6.17 Ops rehearsal for commissioning #2 (LDM-503-11)

6.17.1 Specification

The execution procedure for this test is currently unspecified.

6.17.2 Description

More complete commissioning rehearsal: how do the scientists look at data? How do they provide feedback to the telescope? How do we create/update calibrations?

6.18 Pipelines Release Fall 2019 (LDM-503-11b)

6.18.1 Specification

The execution procedure for this test is currently unspecified.

6.18.2 Description

Science Pipelines release in support of operations rehearsal LDM-503-11. Release contents described in LDM-564. Will be accompanied by a characterization report.

6.19 ComCam Ops Readiness (LDM-503-11a)

6.19.1 Specification

The execution procedure for this test is currently unspecified.

6.19.2 Description

ComCam will be in use in Nov. The DM system must be ready to deal with it.

6.19.3 Comments

"The DM system" should use some further definition: what do we want to test here? Data flow from ComCam to the Data Backbone, or science processing of ComCam data? Note the LSE-79 requirements for DM services in support of ComCam in table 8. They're required by Nov 2019/Feb 2020; it may be more appropriate to test some of them at a later date?

6.20 Ops rehearsal for commissioning #3 (LDM-503-12)

6.20.1 Specification

The execution procedure for this test is currently unspecified.

6.20.2 Description

Dress rehearsal: commissioning starts in April so by this stage we should be ready to do everything needed.

6.21 LSSTCam Ops Readiness (LDM-503-12a)

6.21.1 Specification

The execution procedure for this test is currently unspecified.

6.21.2 Description

6.22 Pipelines Release Fall 2020 (LDM-503-13a)

6.22.1 Specification

The execution procedure for this test is currently unspecified.

6.22.2 Description

Science Pipelines release in support of operations rehearsal LDM-503-13. Release contents described in LDM-564. Will be accompanied by a characterization report.

6.23 Ops rehearsal for data release processing #1 (ComCam data) (LDM-503-13)

6.23.1 Specification

The execution procedure for this test is currently unspecified.

6.23.2 Description

ComCam data will now be available. Demonstrate its use in producing a data release.

6.23.3 Comments

Note that LSE-79 requires a suite of DM services in support of the full camera in May 2020. It seems inappropriate to test them as part of a commissioning ops rehearsal, but they are well before this data. Do we need another test milestone?

6.24 DM Readiness for Science Verification (LDM-503-14)

6.24.1 Specification

The execution procedure for this test is currently unspecified.

6.24.2 Description

Science Verification starts in April. Demonstrate that all required DM software is available.

6.24.3 Comments

SV will include calculating all KPMs to demonstrate that we are reaching the science requirements. That obviously means we'll need code which is both capable of reaching those requirements, and calculating the KPMs.

6.25 Pipelines Release Fall 2021 (LDM-503-15a)

6.25.1 Specification

The execution procedure for this test is currently unspecified.

6.25.2 Description

Science Pipelines release in support of operations rehearsal LDM-503-15. Release contents described in LDM-564. Will be accompanied by a characterization report.

6.26 Ops rehearsal for data release processing #2 (LDM-503-15)

6.26.1 Specification

The execution procedure for this test is currently unspecified.

6.26.2 Description

Science Verification data will now be available. Demonstrate its use in producing a data release.

6.27 Ops rehearsal for data release processing #3 (LDM-503-16)

6.27.1 Specification

The execution procedure for this test is currently unspecified.

6.27.2 Description

Test readiness for operations.

6.28 Final Pipelines Delivery (LDM-503-17a)

6.28.1 Specification

The execution procedure for this test is currently unspecified.

6.28.2 Description

Final Science Pipelines release from the DM Construction Project. Release contents described in LDM-564. Will be accompanied by a characterization report.

6.29 Final operations rehearsal (LDM-503-17)

6.29.1 Specification

The execution procedure for this test is currently unspecified.

6.29.2 Description

Confirm readiness for operations.