Test Approach

Gabriele Comoretto
SW Configuration and Release Engineer

DMLT Face-to-Face
November 6-8 2018

Presentation Summary

- Introduction
- Why?
- Approach Overview
- Test Objects in Jira
- Test Documents vs Jira Objects
- Test Documents Extraction
- Existing Test Documents
- Practical Example

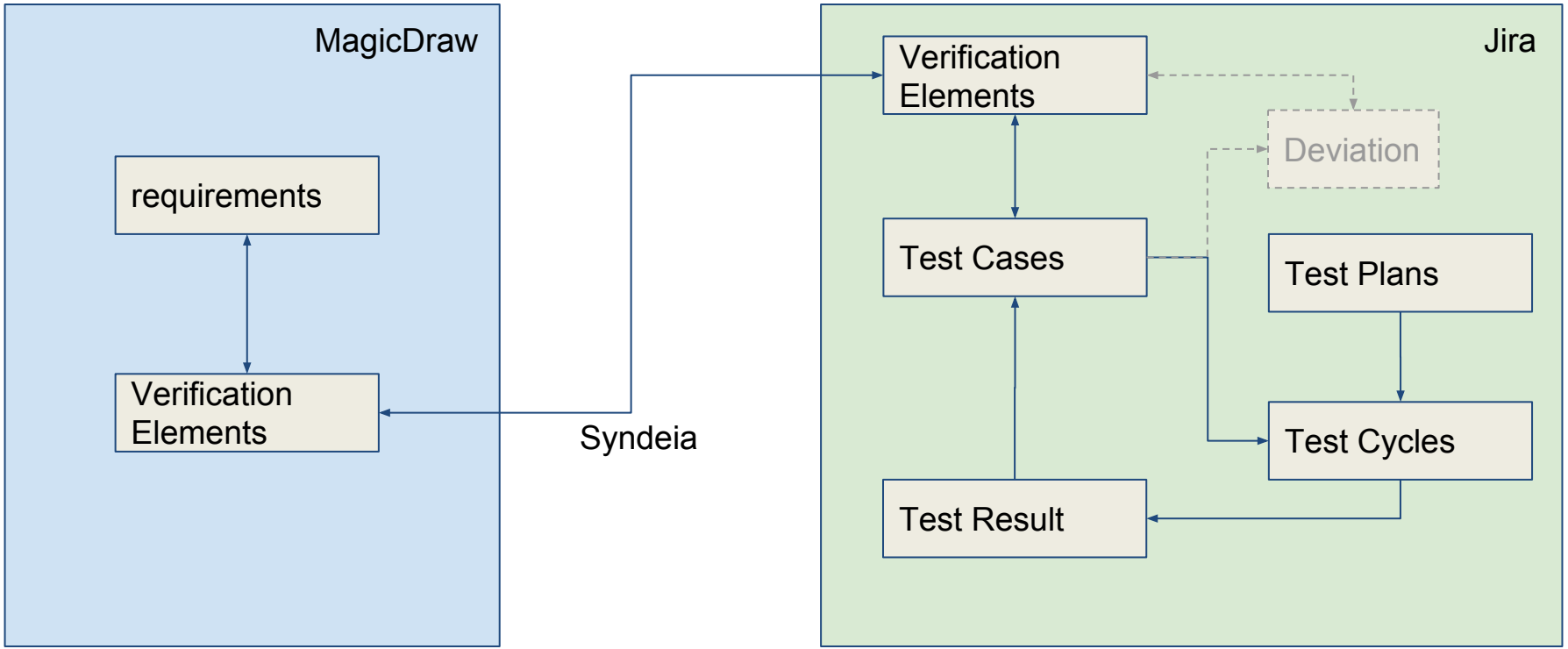
Introduction

- Adaptavist Test Manager (ATM) plugin for Jira
  - Information is organize in folder following the Product Tree structure
  - It is integrated with MagicDraw
- Test Campaign
  - It is a test activity that fulfil a test milestone (LDM-503) or verify a software release
- Documentation:
  - LDM-503 section 2 (just approved via RFC-540 last week)
    - It defines the DM test approach using ATM
  - DM Confluence page:
    - https://confluence.lsstcorp.org/display/DM/DM+Test+Approach
    - It should become a TN
  - SE confluence page:
    - https://confluence.lsstcorp.org/display/SYSENG/LSST+Verification+Architecture

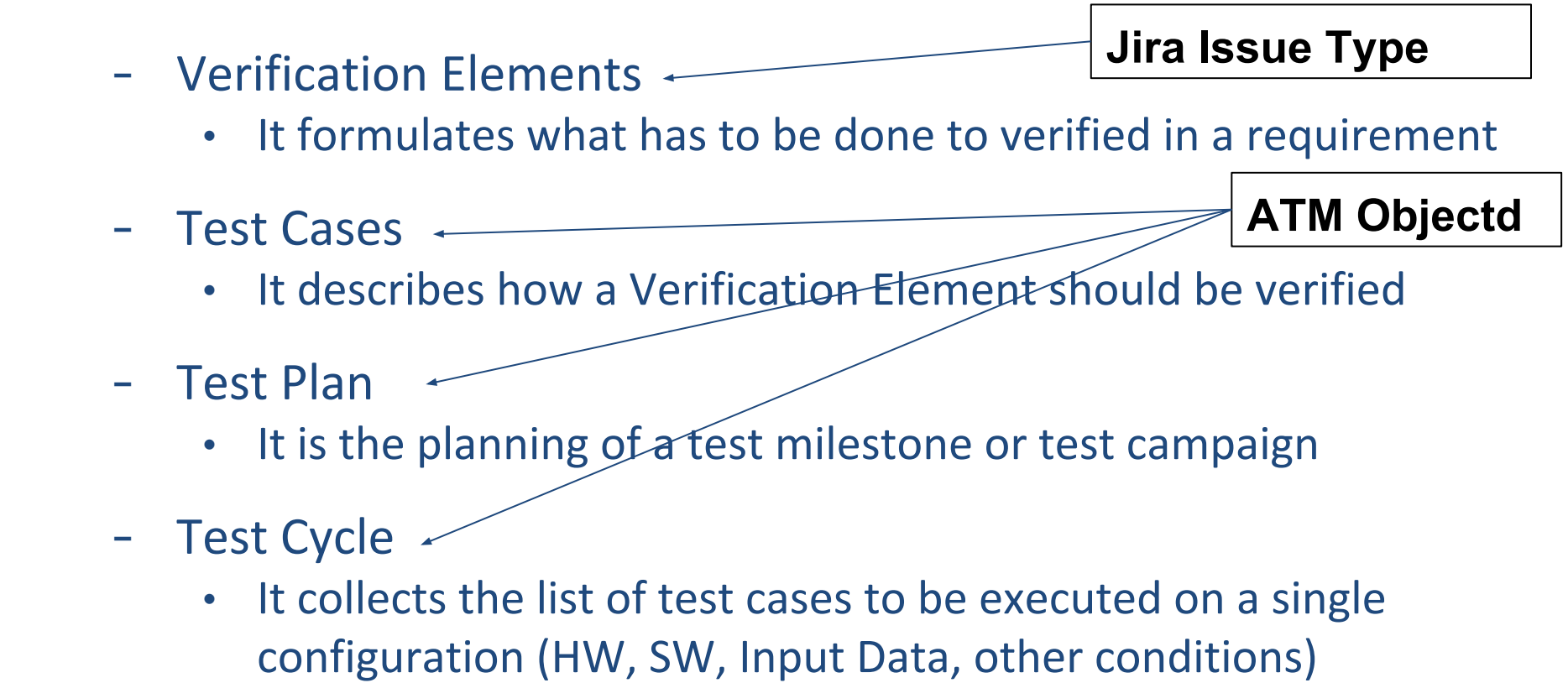
Why?

- LSST need to provide evidence that all requirements at all level have been implemented and validated (global VCD).
  - System Engineering will (this is not a DM task)
    - collect all test information from Jira (Test Management Plugin)
    - integrate it with the requirements information in MagicDraw and
    - provide a Verification Control Document that covers all LSST subsystems
  - DM is required to provide in Jira Test information
- To Produce a DM Verification Control Document (only from Jira)
- It is not just a “VCD exercise”!
  - Test Documents are important for:
    - Identify all components involved in a test activity ensuring we a testing the right thing
    - Document the results of the tests (give visibility)
    - Make the tests repeatable (exact conditions)

Approach Overview



Test Objects in Jira



- Verification Elements
  - It formulates what has to be done to verified in a requirement
- Test Cases
  - It describes how a Verification Element should be verified
- Test Plan
  - It is the planning of a test milestone or test campaign
- Test Cycle
  - It collects the list of test cases to be executed on a single configuration (HW, SW, Input Data, other conditions)

Test Document vs Jira Objects

- **Test Specification**
  - It is the document that collects all **Test Cases** related to a specific DM Component (see product tree and document tree)
  - Approval via RFC, each time that test cases are added or modified
  - Test Cases are related to Verification Elements that are related to the Requirements
- **Test Plan and Report**
  - It includes:
    - Test Campaign planning from the **Test Plan**
      - First issue of the document - approved by the product leader
    - Execution Report from the **Test Cycle(s)**
      - Second issue of the document - approved by “whoever requested the test”
  - Shall we have 2 different documents?

Test Document Extraction

- From Jira
  - A Jenkins job per document
    - extracts the relevant information from Jira
    - formats the LaTeX source
    - submits changes in the document repository (github master branch)
  - Travis will build and make available the pdf at lsst.io
  - The Jenkins job can be scheduled how often we wont, every day, every hour, every week, to be defined
- Test Specification: can be extracted without problems
- Test Plan and Report: extraction script still to be completed

Existing Test Documents

- VCD test information will be extracted from Jira:
  - All test activities documented in old documents need to be ingested in Jira
    - should be possible to be done manually (just few documents)
- Test Specifications
  - to be maintained and updated in Jira (except first 2 sections)
- Test Reports:
  - old DMTRs do not need to be regenerated
  - new Test Plan and Reports to be extracted from Jira

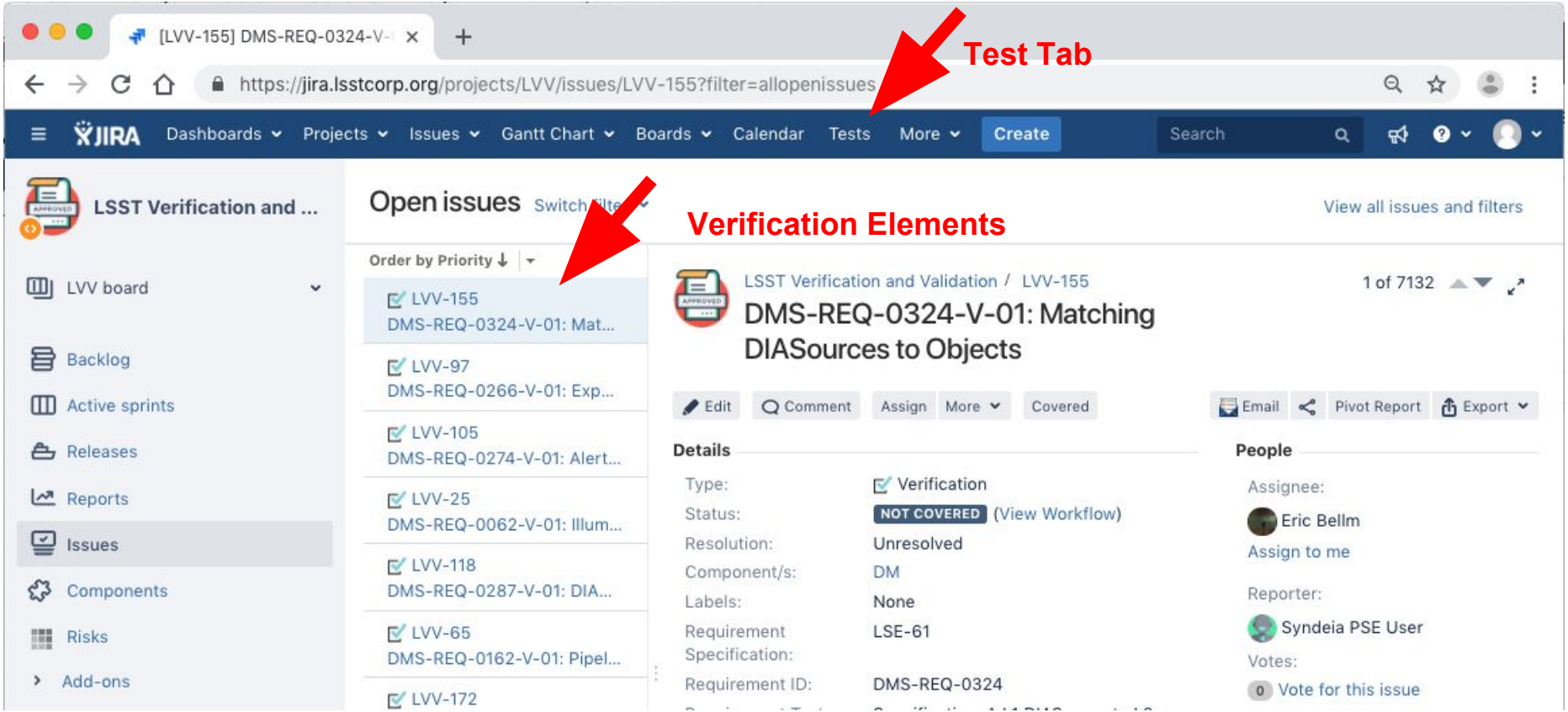
Questions?

Examples

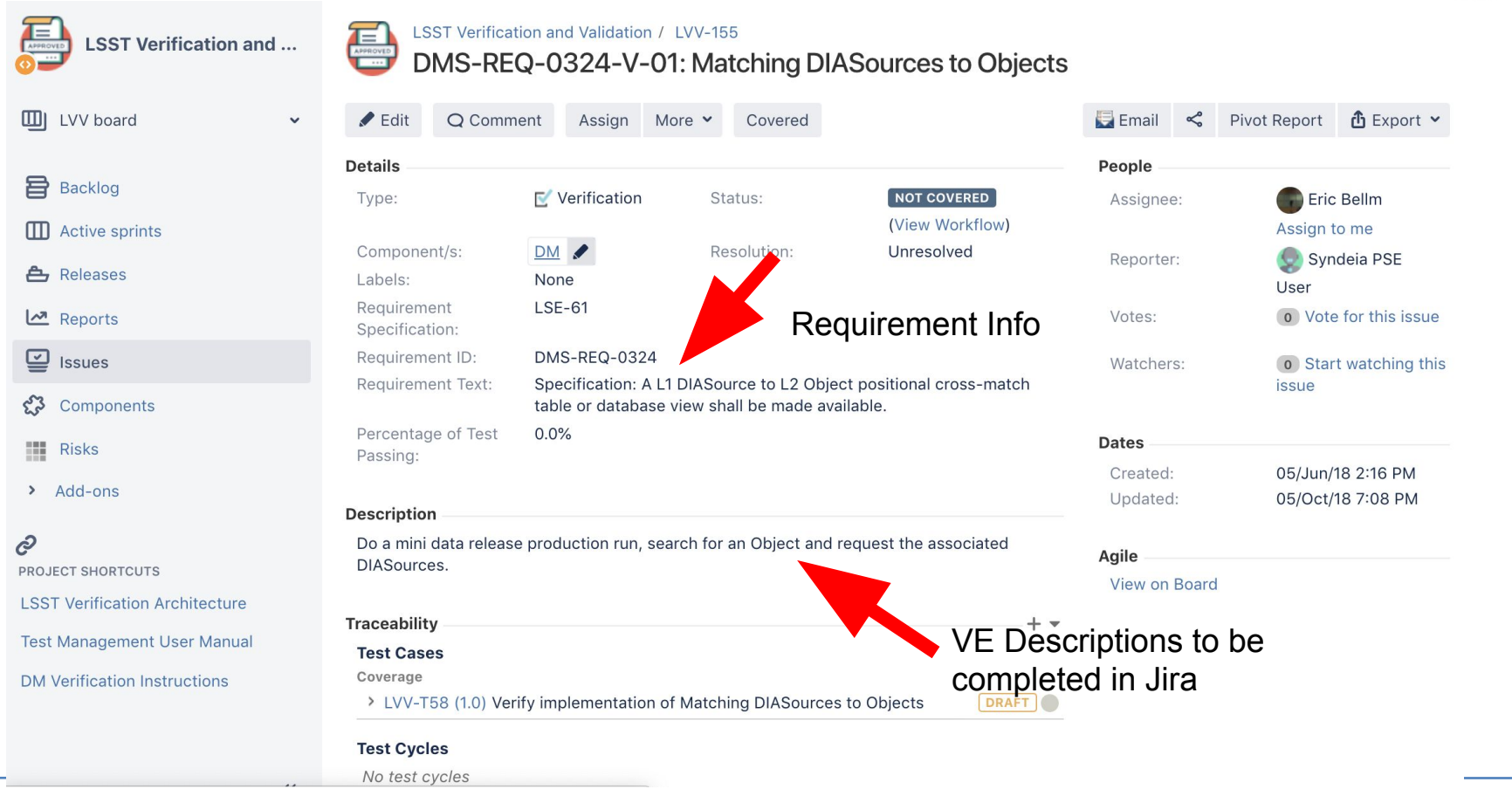
- LVV Jira project
- Verification Elements
- The Test Tab
- A Test Case
  - Test script and traceability

  5
- The Test Specification doc.
- Test Plan
- Test Cycle
- Test Plan and Report Step 1
- Test Player
- Finalizing the Test Plan

Example: LVV Jira project



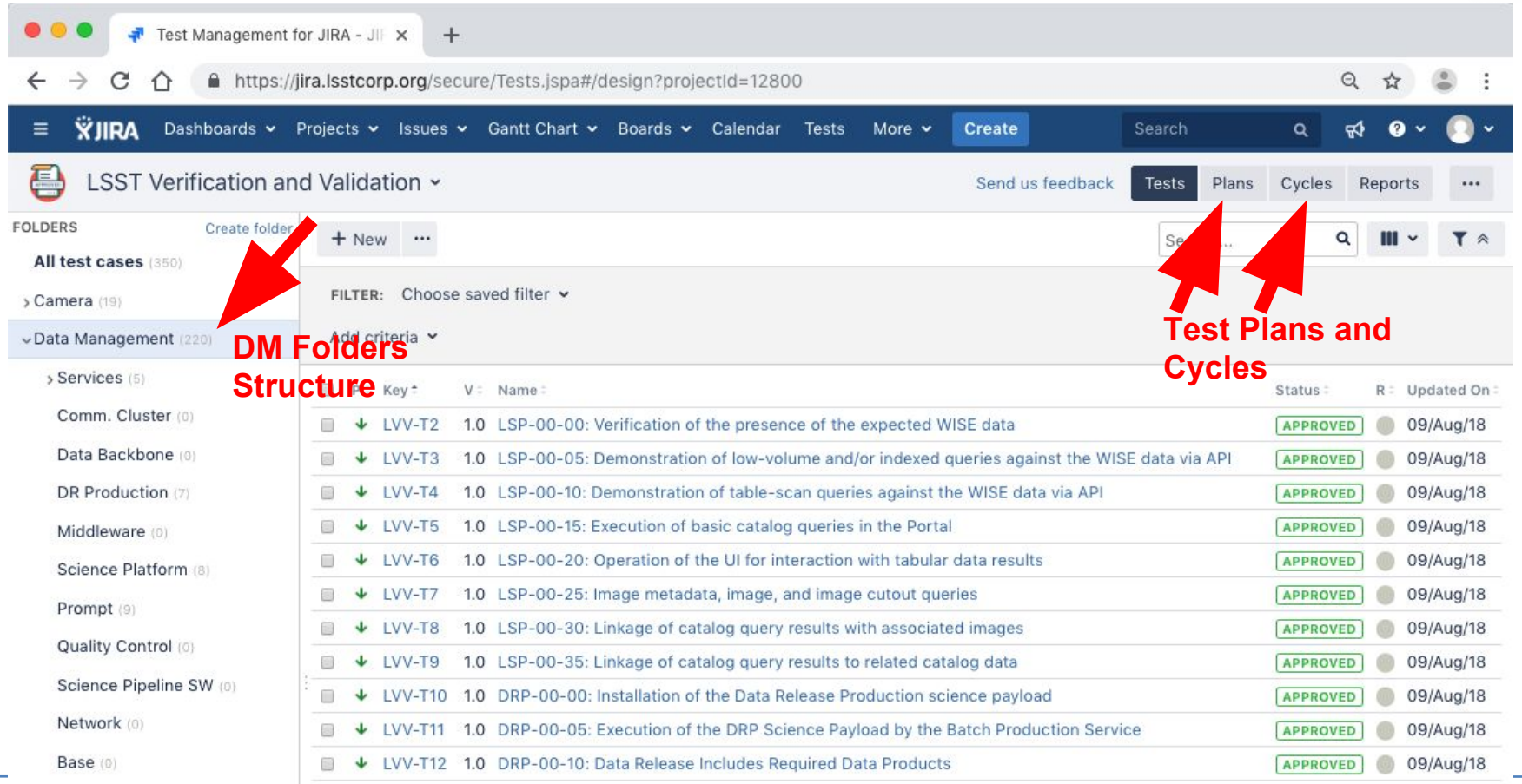
Example: Verification Element



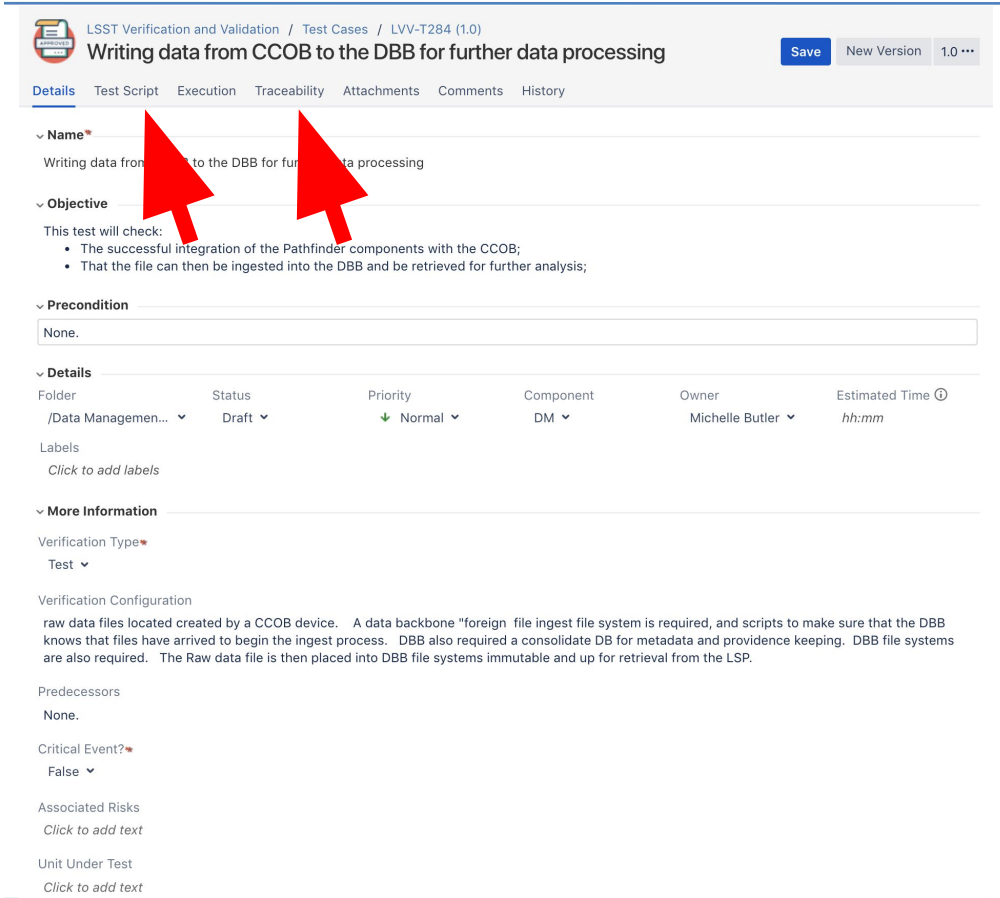
Verification Element Details

- Created empty in MagicDraw
  - 5 per requirement
  - Automatically related with the corresponding requirement
- The LVV-XXX jira issue are synchronized in Jira from MagicDraw using Syndeia.
- Verification elements details need to be completed in Jira
- SE will synchronize them back in MagicDraw

Example: The Test Tab



Example: a Test Case



LSST Verification and Validation / Test Cases / LVV-T284 (1.0)

Writing data from CCOB to the DBB for further data processing

Save | New Version | 1.0

Details | Test Script | Execution | Traceability | Attachments | Comments | History

Name

Writing data fro to the DBB for fu ta processing

Objective

This test will check:

- The successful integration of the Pathfinder components with the CCOB;
- That the file can then be ingested into the DBB and be retrieved for further analysis;

Precondition

None.

Details

Folder	Status	Priority	Component	Owner	Estimated Time
/Data Managemen...	Draft	Normal	DM	Michelle Butler	hh:mm

Labels

Click to add labels

More Information

Verification Type

Test

Verification Configuration

raw data files located created by a CCOB device. A data backbone "foreign file ingest file system is required, and scripts to make sure that the DBB knows that files have arrived to begin the ingest process. DBB also required a consolidate DB for metadata and providence keeping. DBB file systems are also required. The Raw data file is then placed into DBB file systems immutable and up for retrieval from the LSP.

Predecessors

None.

Critical Event?

False

Associated Risks

Click to add text

Unit Under Test

Click to add text

Unit Under Test

Click to add text

Required Software

- CCOB device and the software to produce a file to be transferred and kept
- DBB software to produce a retrieval file for further processing

Test Equipment

- CCOB
- Test machine for LSST Monitoring Service
- consolidate DB
- DBB ingest file system
- DBB output file system
- data transfer protocol to move data from CCOB file systems to DBB ingest file system

Test Personnel

Michelle Gower

Safety Hazards

Click to add text

Required PPE

Click to add text

Postcondition

- CCOB (raw image) files that follow specifications;
- DBB files that follow specifications;
- CCOB device directs a human to where a file is wanted to be stored in the DBB;
- Transfer the file to the DBB ingest area;

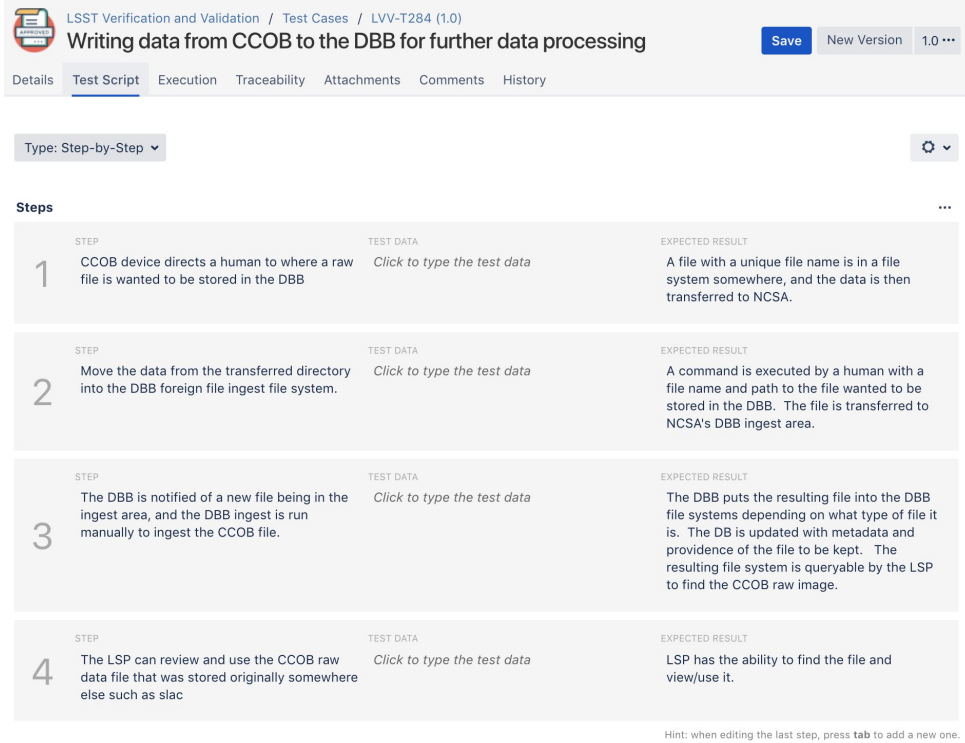
Test Case Details

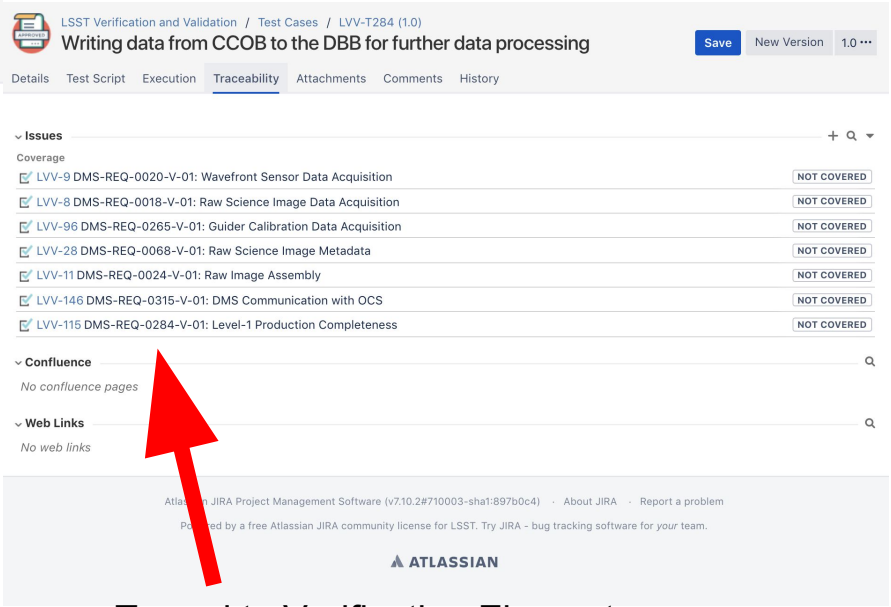
These fields are requested:

- **Name**: short string with the purpose of the test case (do no use the old naming at the beginning)
- **Objective**: Description of the scope of the test case
- **Precondition (Input Specification)**: all inputs required to run the test case
- **Details**: self explanatory, set the owner (who is writing and maintaining), the component is **DM**, the priority can be useful in the future
- **Verification Type** (mandatory): Test, Inspection, Demonstration, Analysis
- **Predecessor**: list here the tests cases that need to be completed before. Those test cases are not to be included in the test script
- **Critical Event** (mandatory): for DM is always FALSE
- **Required Software (Environment Needs - Software)**: the required software that are needed to be available and installed in order to run the test case. This is not the SW under test.
- **Test Equipment (Environment Needs - Hardware)**: the list of hardware (servers, CPU, memory, disk) required to run the test case.
- **Test Personel**: list here other people that is required to validate the test case results
- **Postconditions (Output Specifications)**: what output is expected from the test

Other fields available in jira can be completed and will be added to the Test Specification.

Example: Test Script and Traceability





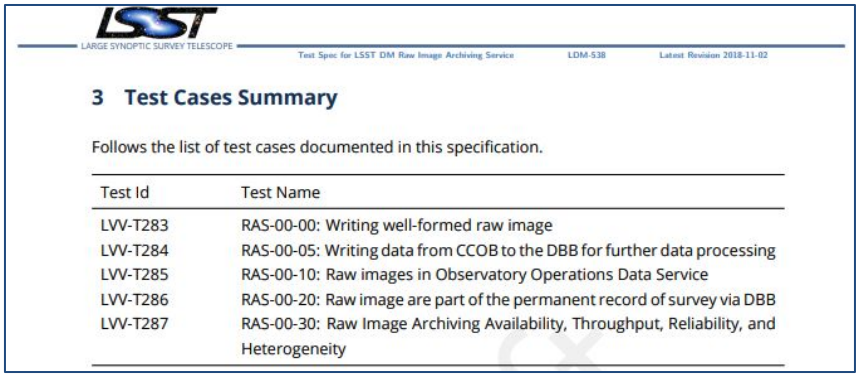
Traced to Verification Elements

Test Script

- The test script has to be generic
  - Not depending on a specific version of the software
  - Nor on a specific environment or configuration
- Input data can be specified for each step
- Expected Result
  - Do not leave this in blank: it is very important to say for each step what output we are expecting
- A step in the test script can be a different test case
  - Use only test cases written for this purpose
  - No multiple levels of inclusion: do not include a test cases that has another test case included in it s test script.

Example: Test Specification

- Document structure and sections 1 and 2
  - Written manually in the document github repository
- Sections 3 and 4
  - Generated from Jira
  - Any manual change will be overwritten from Jira



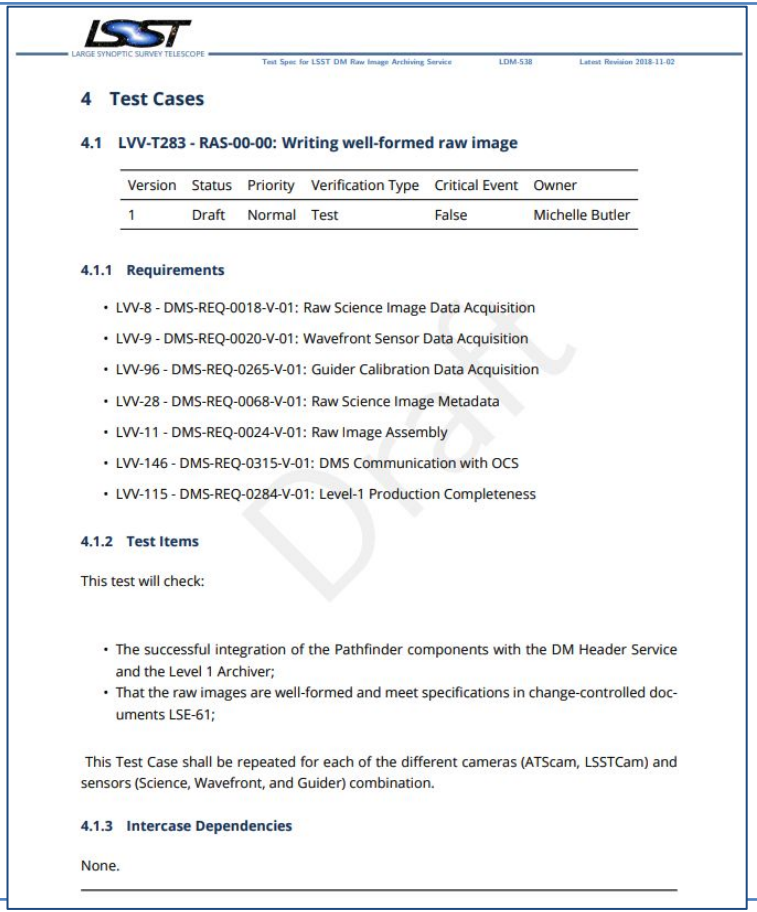
LARGE SYNOPTIC SURVEY TELESCOPE

Test Spec for LSST DM Raw Image Archiving Service LDM-538 Latest Revision 2018-11-02

3 Test Cases Summary

Follows the list of test cases documented in this specification.

Test Id	Test Name
LVV-T283	RAS-00-00: Writing well-formed raw image
LVV-T284	RAS-00-05: Writing data from CCOB to the DBB for further data processing
LVV-T285	RAS-00-10: Raw images in Observatory Operations Data Service
LVV-T286	RAS-00-20: Raw image are part of the permanent record of survey via DBB
LVV-T287	RAS-00-30: Raw Image Archiving Availability, Throughput, Reliability, and Heterogeneity



LARGE SYNOPTIC SURVEY TELESCOPE

Test Spec for LSST DM Raw Image Archiving Service LDM-538 Latest Revision 2018-11-02

4 Test Cases

4.1 LVV-T283 - RAS-00-00: Writing well-formed raw image

Version	Status	Priority	Verification Type	Critical Event	Owner
1	Draft	Normal	Test	False	Michelle Butler

4.1.1 Requirements

- LVV-8 - DMS-REQ-0018-V-01: Raw Science Image Data Acquisition
- LVV-9 - DMS-REQ-0020-V-01: Wavefront Sensor Data Acquisition
- LVV-96 - DMS-REQ-0265-V-01: Guider Calibration Data Acquisition
- LVV-28 - DMS-REQ-0068-V-01: Raw Science Image Metadata
- LVV-11 - DMS-REQ-0024-V-01: Raw Image Assembly
- LVV-146 - DMS-REQ-0315-V-01: DMS Communication with OCS
- LVV-115 - DMS-REQ-0284-V-01: Level-1 Production Completeness

4.1.2 Test Items

This test will check:

- The successful integration of the Pathfinder components with the DM Header Service and the Level 1 Archiver;
- That the raw images are well-formed and meet specifications in change-controlled documents LSE-61;

This Test Case shall be repeated for each of the different cameras (ATScam, LSSTCam) and sensors (Science, Wavefront, and Guider) combination.

4.1.3 Intercase Dependencies

None.

Example: Test Specification - 2

4.1.4 Environment Needs

4.1.4.1 Software

- Level 1 software and services needed to create raw image
- LSST Monitoring Service and plugins specific to monitoring Level 1 Test Stand and services

4.1.4.2 Hardware

- Level 1 test stand
- Test machine for LSST Monitoring Service

4.1.5 Input Specification

None.

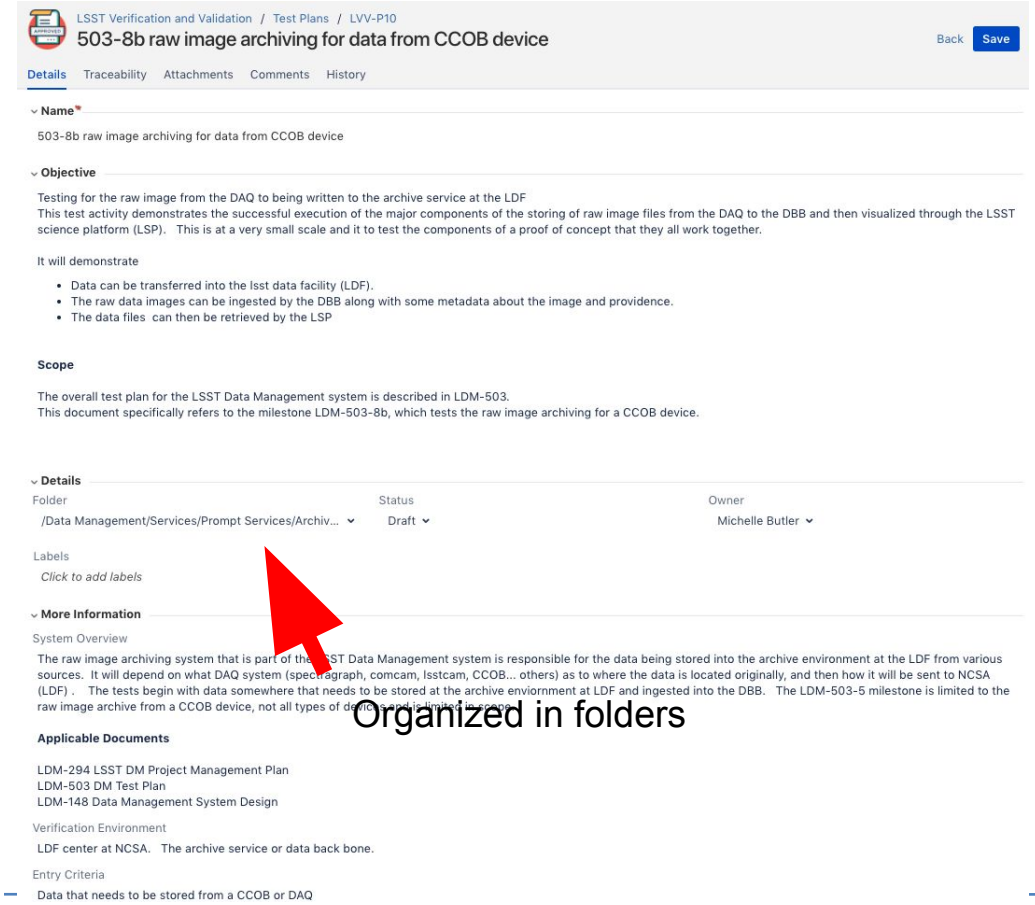
4.1.6 Output Specification

Raw image(s) that follow specifications defined in change-controlled document LSE-61.

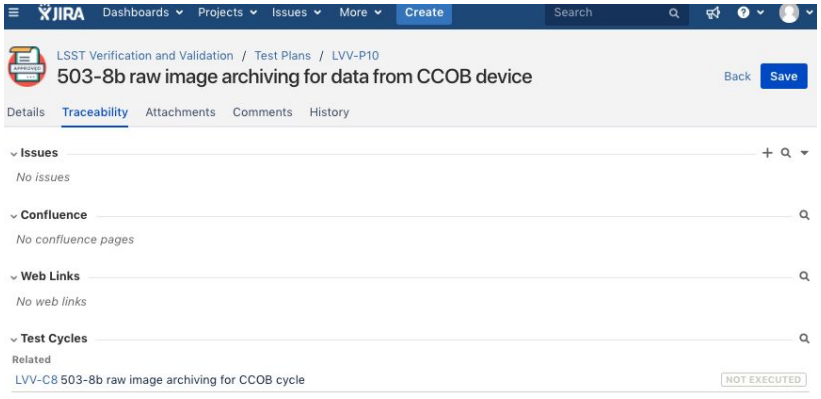
4.1.7 Test Procedure

Step	Description, Input Data and Expected Result	
1	Description	Configure system to pull appropriate data from the DAQ emulator
	Test Data	No data.
	Expected Result	A functional DAQ for images to be received from.
2	Description	Acquire raw data from DAQ readout and DMHS
	Test Data	No data.
	Expected Result	a raw image and a header from the DMHS
3	Description	Fetch data and reassemble correctly, regardless of CCD/Sensor manufacturer type (two different types will be used)
	Test Data	No data.
	Expected Result	Build the data into a fits file
4	Description	CheckCompletenessandcorrectnessoftherawimagesincludingformat,metadata,and image data; • Check proper fetch and reassembly of image data from camera DAQ (correct format and data); • Check proper merge of header service data with image data; • Check correct insertion of exposure specific data needed in the data file that is not supplied by header service; • Check minimum required metadata (from requirements document LSE-61) exists in raw image header;
	Test Data	No data.
	Expected Result	a well formed FITS file with a proper header that has been verified to be correct.
5	Description	Check that the checksum of the file matches the previously calculated value that will be passed on to downstream services
	Test Data	No data.
	Expected Result	a MD5sum number generated from the step 4 file.
6	Description	Check confirmation that the data files arrive at their destination intact
	Test Data	No data.
	Expected Result	a transfer of the file to the correct location for further retrieval from other services.
7	Description	Check that LSST Monitoring Service showed the appropriate information successfully
	Test Data	No data.
	Expected Result	all systems remained green through out the test, and showed all systems up and available.

Example: Test Plan

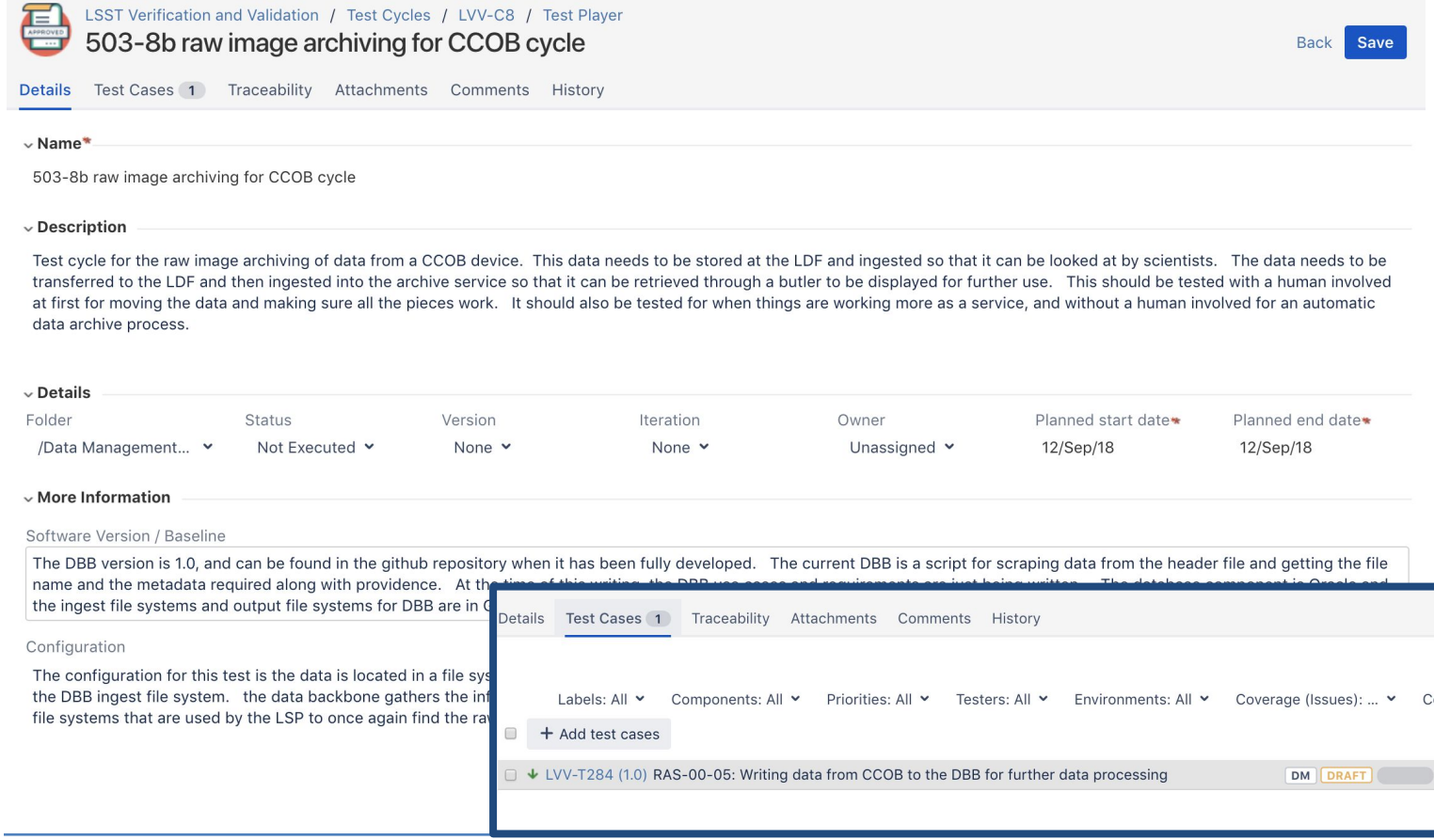


Organized in folders



Related Test Cycles

Example: Test Cycle



Test Plan and Report: step 1

From the Test Plan jira object:

- **Name**: identification of the test activity. Better if we include the work “Plan” in it.
- **Objective**: describe the objective and scope of the test plan (test campaign). The heading “Scope” has to be in bold.
- **Details**: self explanatory. Identify always the owner.
- **Verification Environment (Test Configuration - Hardware)**: describe the environment to be used for the test campaign
- **System Overview**: description of the system under test and the list of applicable documents. The heading “Applicable Documents” has to be in bold

From the Test Cycle jira object:

- **Name:** name of the test cycle: important in case a test plan has multiple cycles
- **Description:** short description of the test cycle, important in case a test plan has multiple cycles
- **Software Version / Baseline:** the software under test and its version identification
- **Configuration:** list the configuration used for the this cycle
- **Test Cases:** in the Test Cases Tab need to list in the right order the test cases to be executed.
  - A test case can be included only one time in a test cycle

Example: Test Plan and Report - step 1

LDM-503-8b (Raw Image Archiving) Test Plan and Report

1 Introduction

1.1 Objectives

Testing for the raw image from the DAQ to being written to the archive service at the LDF This test activity demonstrates the successful execution of the major components of the storing of raw image files from the DAQ to the DBB and then visualized through the LSST science platform (LSP). This is at a very small scale and it to test the components of a proof of concept that they all work together.

It will demonstrate

- Data can be transferred into the lsst data facility (LDF).
- The raw data images can be ingested by the DBB along with some metadata about the image and providence.
- The data files can then be retrieved by the LSP

1.2 Scope

The overall test plan for the LSST Data Management system is described in LDM-503. This document specifically refers to the milestone LDM-503-8b, which tests the raw image archiving for a CCOB device.

1.3 System Overview

The raw image archiving system that is part of the LSST Data Management system is responsible for the data being stored into the archive environment at the LDF from various sources. It will depend on what DAQ system (spectragraph, comcam, lsstcam, CCOB... others) as to where the data is located originally, and then how it will be sent to NCSA (LDF) . The tests begin with data somewhere that needs to be stored at the archive enviornment at LDF and

DRAFT 1 DRAFT

Once the above sections are completed, this document can be reviewd in order to e
the test activity can start.

Section 5 is filled after the test activity is completed. Its includes an overview of the
5.1 while 5.2 provides more detailed results from each individual test case.

2 Test Configuration

The configuration for this test is the data is located in a file system somewhere. It is t
to NCSA into a the LSST GPFS file system. Files that are needed in the DBB are mo
DBB ingest file system. the data backbone gathers the info from the headers and
it inserts rows into the DBB consolidated DB environment and moves the file to th
systems that are used by the LSP to once again find the raw image file to be viewed
for further analysis.

2.1 Hardware

LDF center at NCSA. The archive service or data back bone.

2.2 Software

The DBB version is 1.0, and can be found in the github repository when it has
developed. The current DBB is a script for scraping data from the header file and
file name and the metadata required along with providence. At the time of this
DBB use cases and requirements are just being written. The database componen
and the ingest file systems and output file systems for DBB are in GPFS at the LDF

2.3 Entry Criteria

Data that needs to be stored from a CCOB or DAQ.

DRAFT 3

2.4 Exit Criteria

Data is transferred in proper form and can be viewed on LSP after b
and ingested into DBB.

3 Personnel

Following personnel is involved in the test activity:

- Test Plan (LVV-P10) Owner: Michelle Butler
- Test Cycle (LVV-C8) Owner: Unassigned
- LVV-T284 Tester: Michelle Butler
- Additional Test Personel involved: None

4 Planned Test Activities

4.1 Test Cycle LVV-C8

Test cycle for the raw image archiving of data from a CCOB device. This data needs to be stored at the LDF and ingested so that it can be looked at by scientists. The data needs to be transferred to the LDF and then ingested into the archive service so that it can be retrieved through a butler to be displayed for further use. This should be tested with a human involved at first for moving the data and making sure all the pieces work. It should also be tested for when things are working more as a service, and without a human involved for an automatic data archive process.

4.1.1 LVV-T284: Writing data from CCOB to the DBB for further data processing

This test will check: The successful integration of the Pathfinder components with the CCOB; That the file can then be ingested into the DBB and be retrieved for further analysis;

Step	Description
1	CCOB device directs a human to where a file is wanted to be stored in the DBB
2	Move the data from the transferred directory into the DBB foreign file ingest file system.
3	The DBB is notified of a new file being in the ingest area, and the DBB ingest is run manually to ingest the CCOB file.
4	The LSP can review and use the CCOB raw data file that was stored originally somewhere else such as slac

5 DRAFT

5 Test Results

5.1 Overview of the Test Results

5.1.1 Summary Table

TEST CASE ID	PASS/FAIL	COMMENTS
LVV-T284	Not Run	

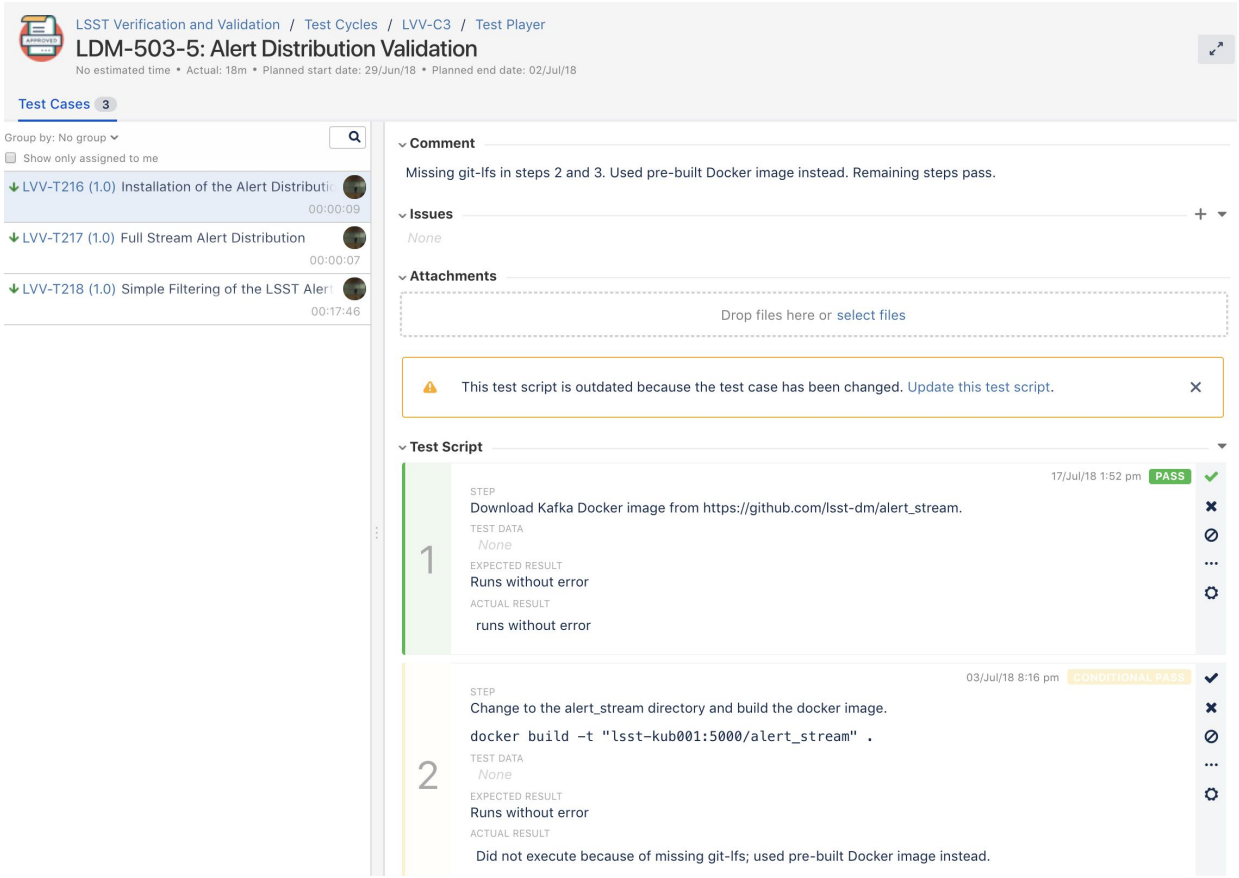
5.1.2 Overall Assessment

5.1.3 Recommended Improvements

5.2 Detailed Test Results

Empty, to be completed after the test is executed

Example: Test Player



For each test case, the tester has to go through all the steps:

- execute them
- “post” the results in the **Actual Results** field
- Set the step status as:
  - Pass
  - Failed
  - Conditional Pass

For each test case has to write a comment that then is reported in the summary table of DMTR section 5.

Report issues related with the test

Finalizing the Test Plan

LSST Verification and Validation / Test Plans / LVV-P1

LDM-503-5: Alert Distribution Validation

Save

Details | Traceability | Attachments | Comments | History

Exit Criteria

Click to add text

PMCS Activity

Click to add text

☐ Observing Required?

Verification Artifacts

Click to add text

Overall Assessment

The test campaign is considered in overall: PASS.

Missing elements on the Kubernetes Commons prevented us from building the Docker images in place and using a complete night of unique alerts. However, using pre-built Docker images with a smaller, repeated sampl[illegible]f alerts enabled the test to proceed as expected.

Recommended Improvements

Re-execution with the missing components on the Kubernetes Commons.

Examination of the deserialized alerts in the logs is cumbersome and does not allow real verification of the alert content. Improved tools for counting alerts, comparing their contents to the sent alerts, and timing throughput would all improve the utility of this test.

Questions?