

ROC Software System Validation Plan

Sonny LION



Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique

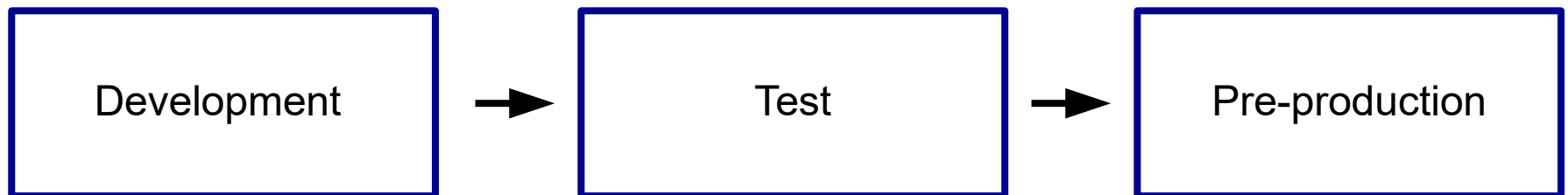


Software Testing Levels

- Unitary tests: individual units verification (design level)
- Integration tests: individual units combination verification (interfaces level)
- Validation tests: global functionalities verification (compliance with the RSSS)

Validation steps

- Validation of new functionality in the development environment
- Validation in the test environment
 - ➔ **Continuous integration** : non-regression and compatibility with interfaces
- Validation in the pre-production environment
 - ➔ compatibility with the **production** environment





1. Validation tools

Software validation tools

Language	Launcher	Framework	Miscellaneous
Python	tox	Pytest	hypothesis
JavaScript	Mocha	Chai	enzyme

- **Hypothesis:** library to parametrize tests and simply generate random data matching given specifications.
- **Enzyme:** JavaScript Testing utility for React that makes it easier to assert, manipulate, and traverse React Components.

System validation tools

- **Jenkins:** extensible automation / continuous integration server
- **Gitlab:** web platform used to monitorate versioning, issues and continuous integration
- **Locust:** python utility to do distributed load testing used to perform stress tests on the MUSIC Web tools

Gitlab server

→ maintained by the
Direction Informatique de
l'Observatoire (DIO)



06 Sep, 2017 4 commits



Add yolk to deps (yolk is used to list dependencies and metadata)

Lion Sonny committed 2 months ago



d6462f35



Browse Files



Add script to list python dependencies

Lion Sonny committed 2 months ago



e8f94c13



Browse Files



Add js_dependencies.txt to ignore list

Lion Sonny committed 2 months ago



ba09791c



Browse Files



Add script to compute js dependencies with names, versions and descriptions

Lion Sonny committed 2 months ago

8d2f25b6



Browse Files

31 Aug, 2017 42 commits



Merge branch 'feature/authentication' into 'develop' ...

Lion Sonny committed 2 months ago




dda0f4be



Browse Files

[TV] TM and TC log empty unit tests

#110 · opened a week ago by Lion Sonny  Sprint S5 - Crash test Critical

  0

updated a week ago


[TV] Science page empty unit tests

#106 · opened a week ago by Lion Sonny  Sprint S5 - Crash test Critical In Progress

  0

updated a week ago

[MUSIC] Design document

#105 · opened 3 weeks ago by Lion Sonny  Sprint S4 - Doc In Progress

Nov 8, 2017 4:39 pm

  2

updated 2 weeks ago

[TV] Split TM/TC page

#104 · opened a month ago by Lion Sonny Feature Medium

 0

updated a month ago


[FIGARO] Validate form submission

#102 · opened a month ago by Aboubacar Amadou Aichatou  Sprint S4 - In Progress

  2

updated a day ago

[MUSIC] SVP (release for the end-to-end version)

#101 · opened a month ago by Lion Sonny  Sprint S4 - Doc In Progress

  7

updated 2 weeks ago



Jenkins **3** search log In | sign up

Jenkins > music > [ENABLE AUTO REFRESH](#)

- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- [Workspace](#)
- [Build Now](#)
- [Delete Project](#)
- [Configure](#)
- [Coverage Report](#)
- [Embeddable Badges](#)
- [Test Results Analyzer](#)

Project music

- [Coverage Report](#)
- [Workspace](#)
- [Recent Changes](#)

Build History [trend](#)

find

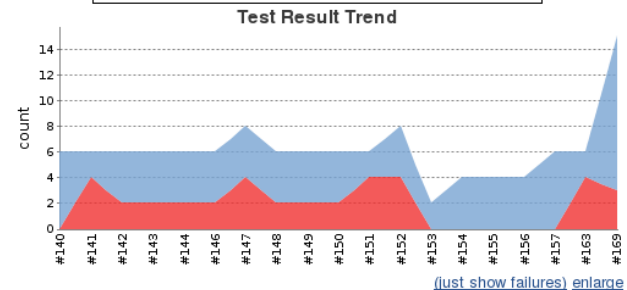
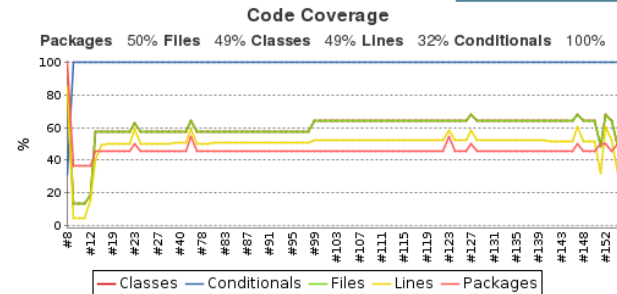
- #170** Nov 17, 2017 10:33 AM
Started by GitLab push by Aboubacar Amadou Aichatou
- #169** Nov 16, 2017 9:32 PM
Started by GitLab push by NGUYEN Quynh Nhu
- #168** Nov 16, 2017 10:42 AM
Started by GitLab push by Aboubacar Amadou Aichatou

Permalinks

- [Last build \(#170\), 1 day 14 hr ago](#)
- [Last stable build \(#157\), 18 days ago](#)
- [Last successful build \(#163\), 10 days ago](#)
- [Last failed build \(#170\), 1 day 14 hr ago](#)
- [Last unstable build \(#163\), 10 days ago](#)
- [Last unsuccessful build \(#170\), 1 day 14 hr ago](#)
- [Last completed build \(#170\), 1 day 14 hr ago](#)

[add description](#)

[Disable Project](#)



Jenkins Server (roc-dev)

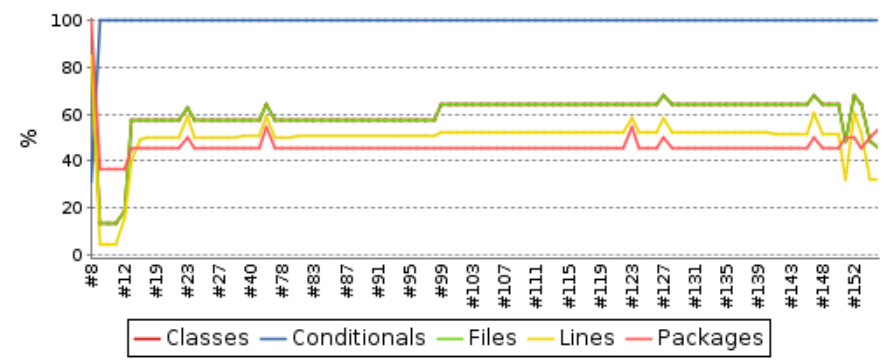
- Centralization: common interface for jobs, reports, etc.
- Automation: compilation, testing, deployment, etc.
- Test logs



Code Coverage

Cobertura Coverage Report

Trend



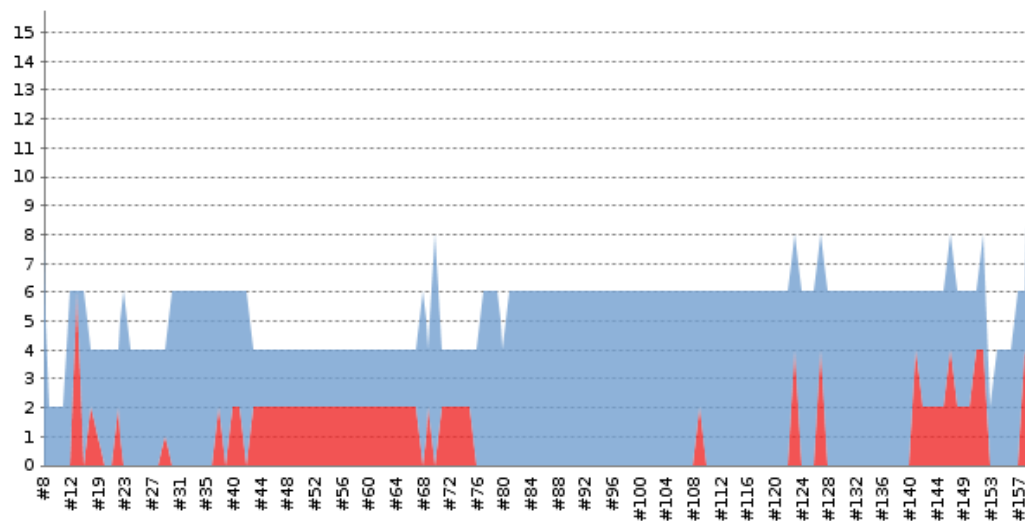
Project Coverage summary

Name	Packages	Files	Classes	Lines	Conditionals
Cobertura Coverage Report	54% 7/13	46% 22/48	46% 22/48	32% 203/633	100% 0/0

Coverage Breakdown by Package

Name	Files	Classes	Lines	Conditionals
.	0% 0/1	0% 0/1	0% 0/13	N/A
accounts	40% 2/5	40% 2/5	31% 51/167	N/A
accounts.migrations	0% 0/1	0% 0/1	0% 0/9	N/A

History for Test Results



show [duration](#)

Build	Description	Duration	Fail	Skip	Total
music #172		6,4 sec	4	0	6
music #169		15 sec	3	0	15
music #163		4,8 sec	4	0	6
music #157		10 sec	0	0	6
music #156		10 sec	0	0	4
music #155		9,4 sec	0	0	4
music #154		11 sec	0	0	4
music #153		20 ms	0	0	2

Jenkins - Test reports

Options

Number of builds: (all:)

Display run time for each test:

Chart type: Line Bar Pie

Chart Data Type:

Update

Download Test (CSV) Search:

Expand All

Collapse All

New Failures	Chart	See children	Build Number → Package-Class-Testmethod names ↓	172	171	170	169	168
	<input type="checkbox"/>	+	(root)	<u>PASSED</u>	<u>N/A</u>	<u>N/A</u>	<u>FAILED</u>	<u>N/A</u>
	<input type="checkbox"/>	+	backend.figaro.tests	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
	<input type="checkbox"/>	+	backend.tests.test_api	<u>FAILED</u>	<u>N/A</u>	<u>N/A</u>	<u>PASSED</u>	<u>N/A</u>
	<input type="checkbox"/>	+	backend.tests.test_models	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>
	<input type="checkbox"/>	+	backend.tv.tests	<u>FAILED</u>	<u>N/A</u>	<u>N/A</u>	<u>PASSED</u>	<u>N/A</u>
	<input type="checkbox"/>	+	should ..	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>PASSED</u>	<u>N/A</u>
	<input type="checkbox"/>	+	tests.test_serializers	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>	<u>N/A</u>

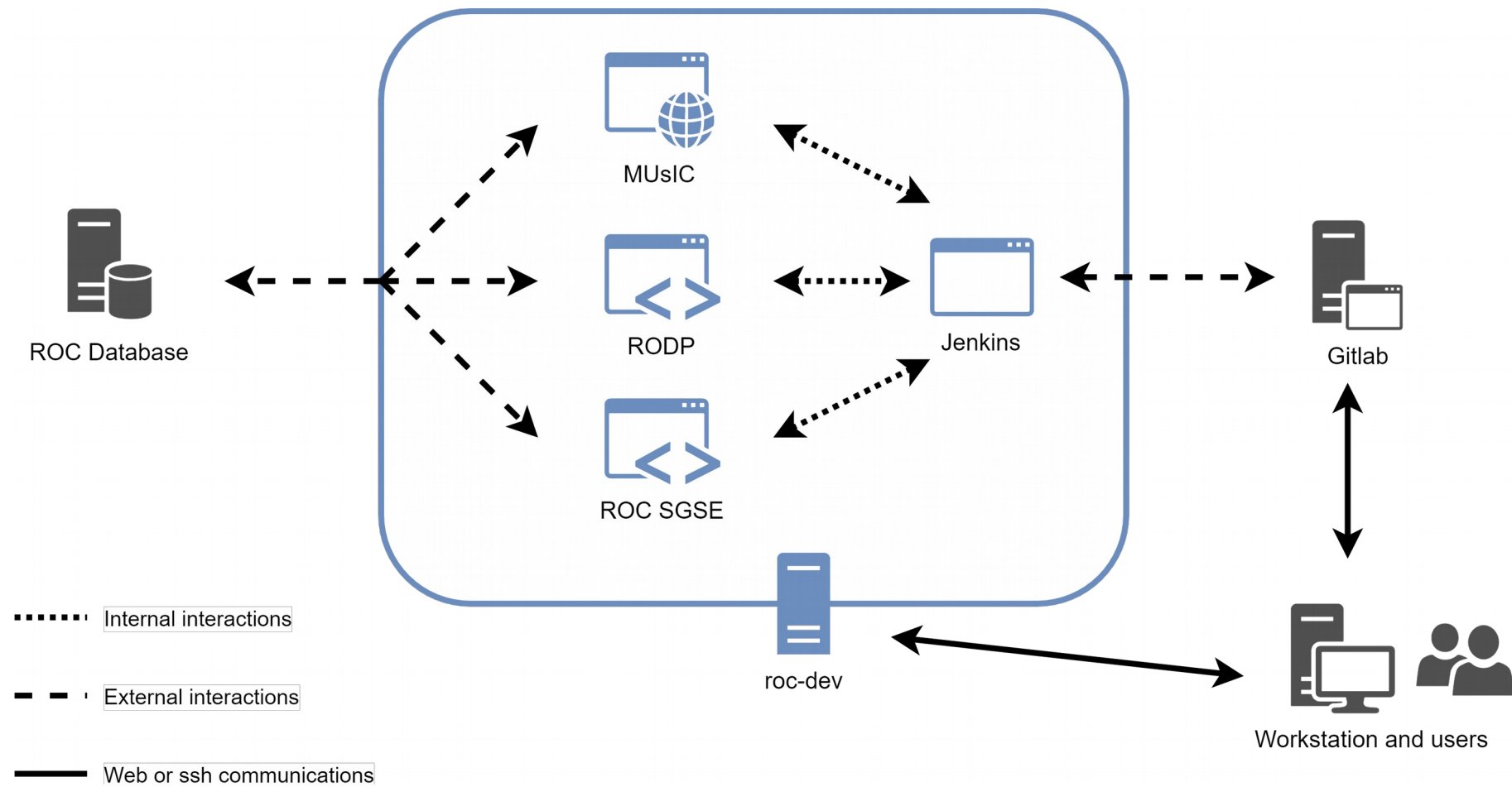


2. Validation resources and infrastructure

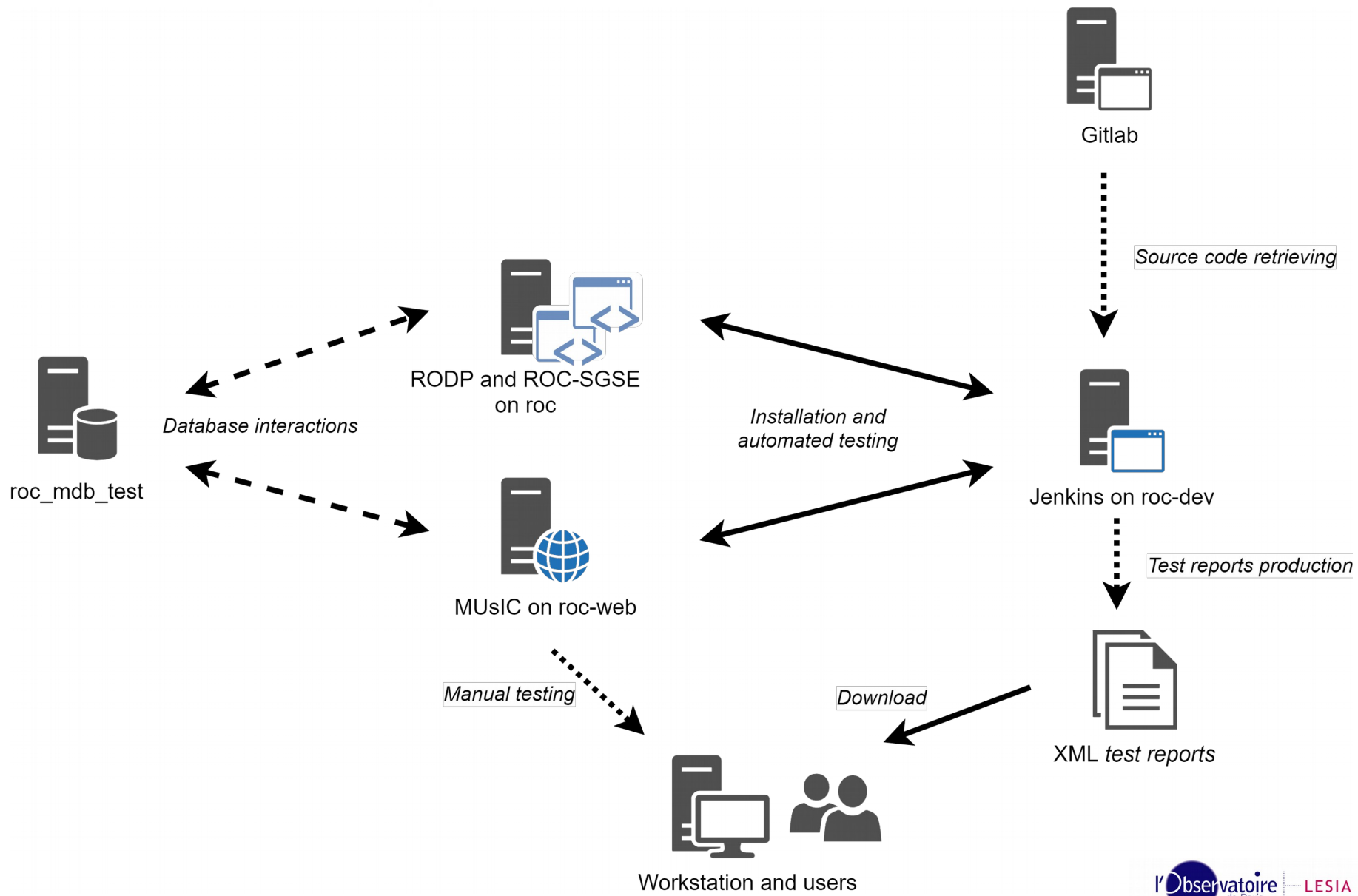
Test inputs

- Representative subset of data produced and processed during RPW ground calibration campaigns
- Data provided by ESA
- Data generated by the ROC (low-latency testcards)
- RPW Calibration Software (RCS) E2E data packages (teams in charge)
- Random data (hypothesis package)

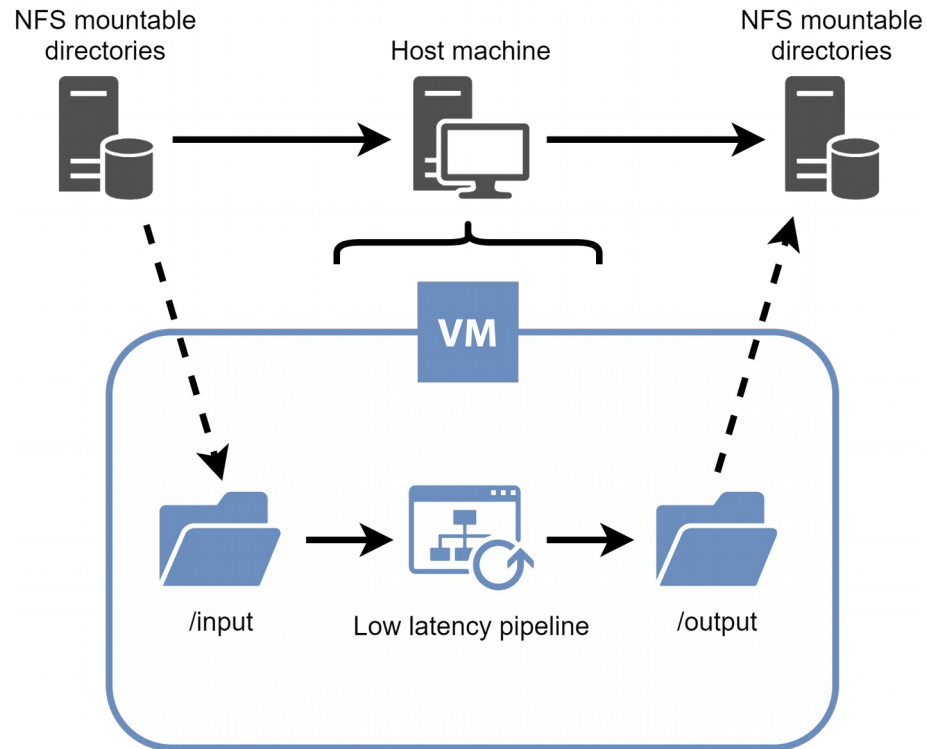
Validation resources - Test environment



Validation resources - Pre-production environment



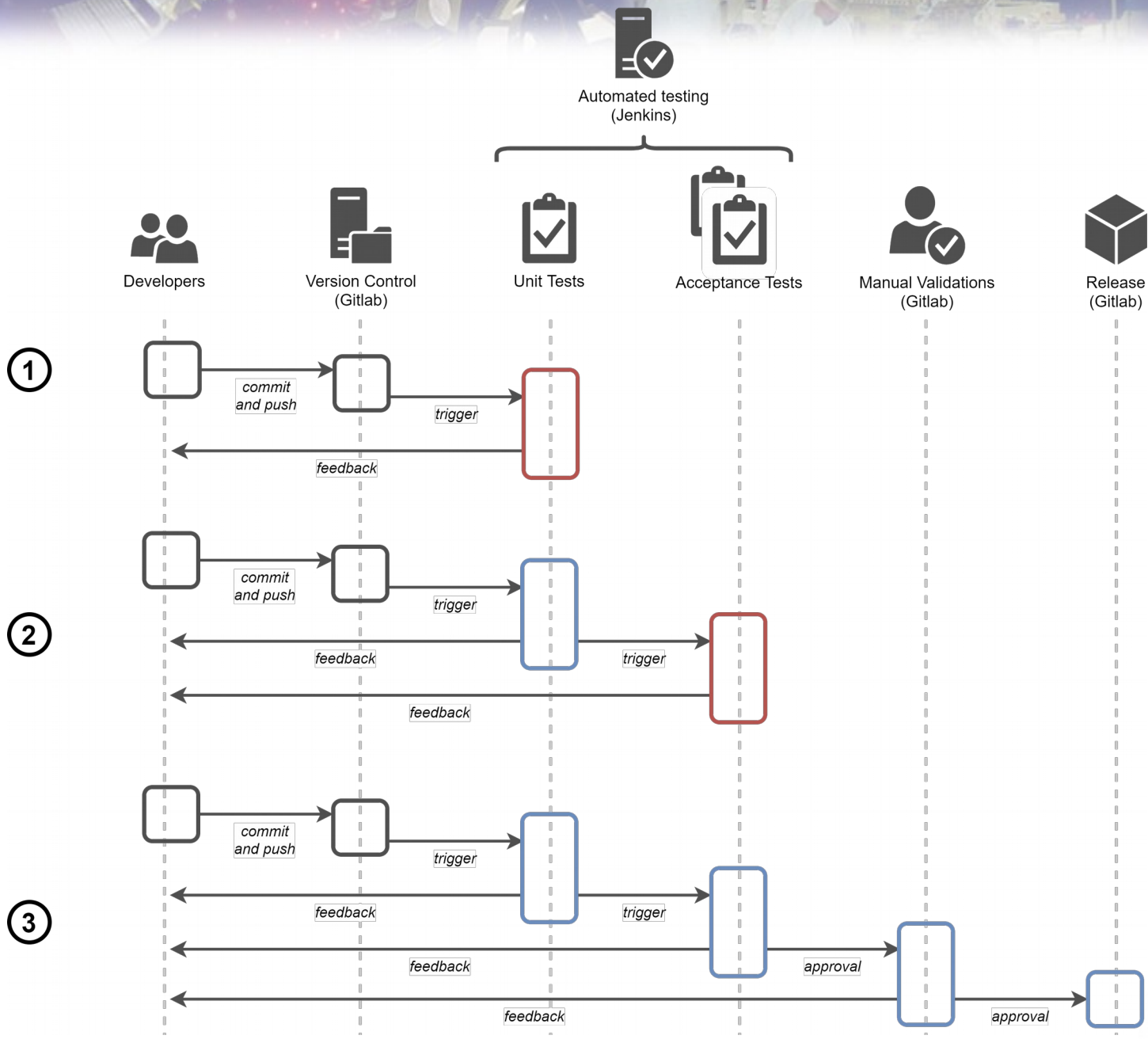
Validation resources - LLVM testing environment





3. Continuous integration

Continuous integration cycle





4. Validation Campaign

Pipeline components validation

- Basic testing: independent of the pipeline architecture
- Task testing: depend on the whole pipeline architecture (between unit and integration tests)
- Command testing: depend on the whole pipeline architecture (**validation tests**)

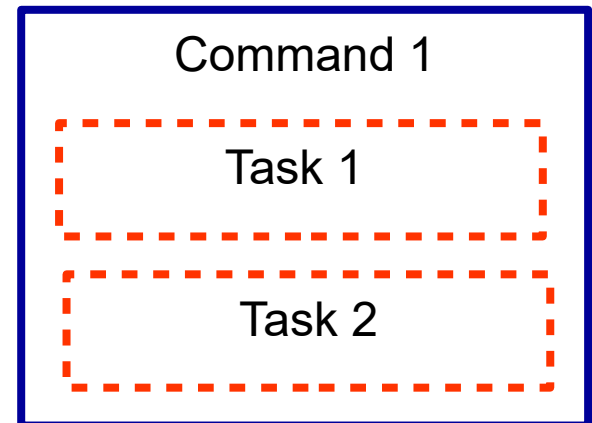
Note : RCS tests are included in the command tests

Performance testing

- Evaluated by measuring the time taken to execute each command

Robustness testing

- part of each test phase
- ensuring that the pipeline responds as expected to erroneous inputs



Backend testing

- Unit testing: views, serializers and models
- Validation tests : calling all API end points, simulate complete user queries, use erroneous requests and test the permission system

Frontend testing

- Unit/integration tests for components, actions, sagas and reducers

Stress/performance testing

- Generate predefined requests via Locust to simulate the activity of a given number of users

Robustness testing

- Part of each test phase
 - ensuring that MUSIC responds as expected to erroneous inputs

Beta testing

- Frontend validation
- Frontend performance
- Ergonomic aspects
- Security aspects

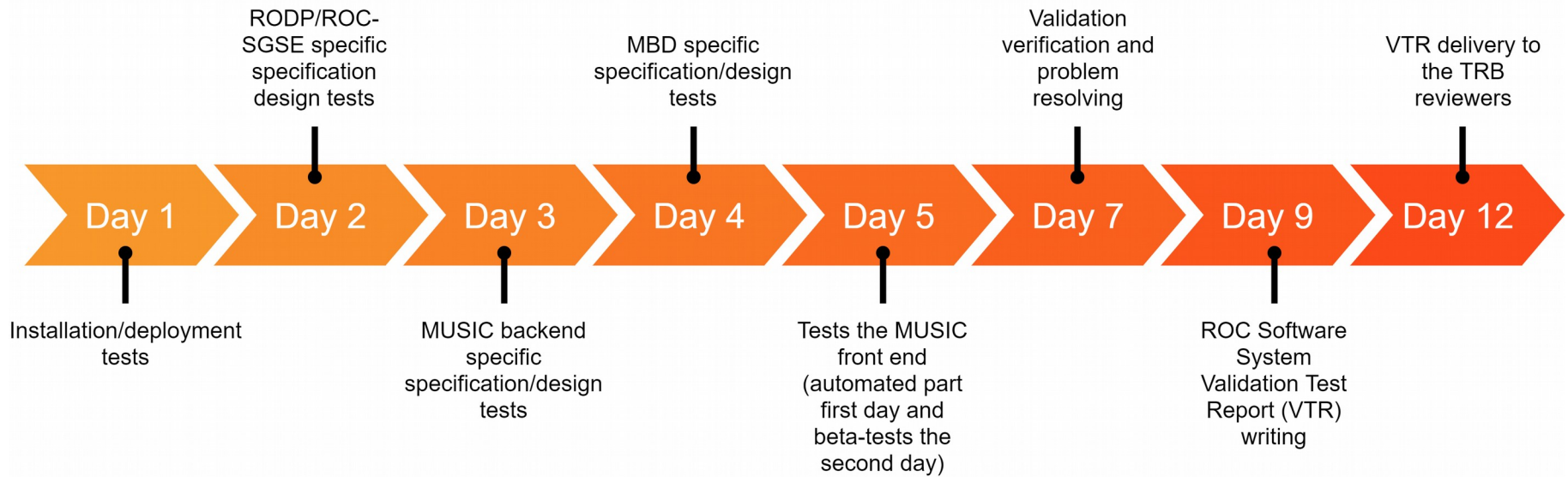
Test Reports

- Pytest and Mocha publish **JUnit XML test reports** (collected by Jenkins)
 - include **metrics** (test and comment coverage, coding guidelines compliance)
- Locust results are saved in a **CSV file**.
 - include statistics on response times and errors
- Beta test reports
 - series of test cases with evaluation grids and feedback sections (cf. the **Beta Testing Report Template**)

All these reports will be compiled in a single document, the ROC Software System **Validation Test Report**



Validation campaign overall schedule



Validation approach - Risks

Point of failure	Consequences	Impact	Probability
Network connection issues	Installation and servers communication issues	Major	Unlikely
Server down	Unable to run the RSS/Jenkins/Gitlab (depending on the server)	Major	Unlikely
Power failure	All the servers are off	Major	Very unlikely
Staff absence	The validation tasks will be done by another operator	Minor	Possible
Computer breakdown	The validation campaign will be done using another computer	Minor	Unlikely
Database corruption/error	Integration and acceptance tests can not be performed	Major	Very unlikely