Software Quality Assurance (SQA) Report

22-26 Aug 2016

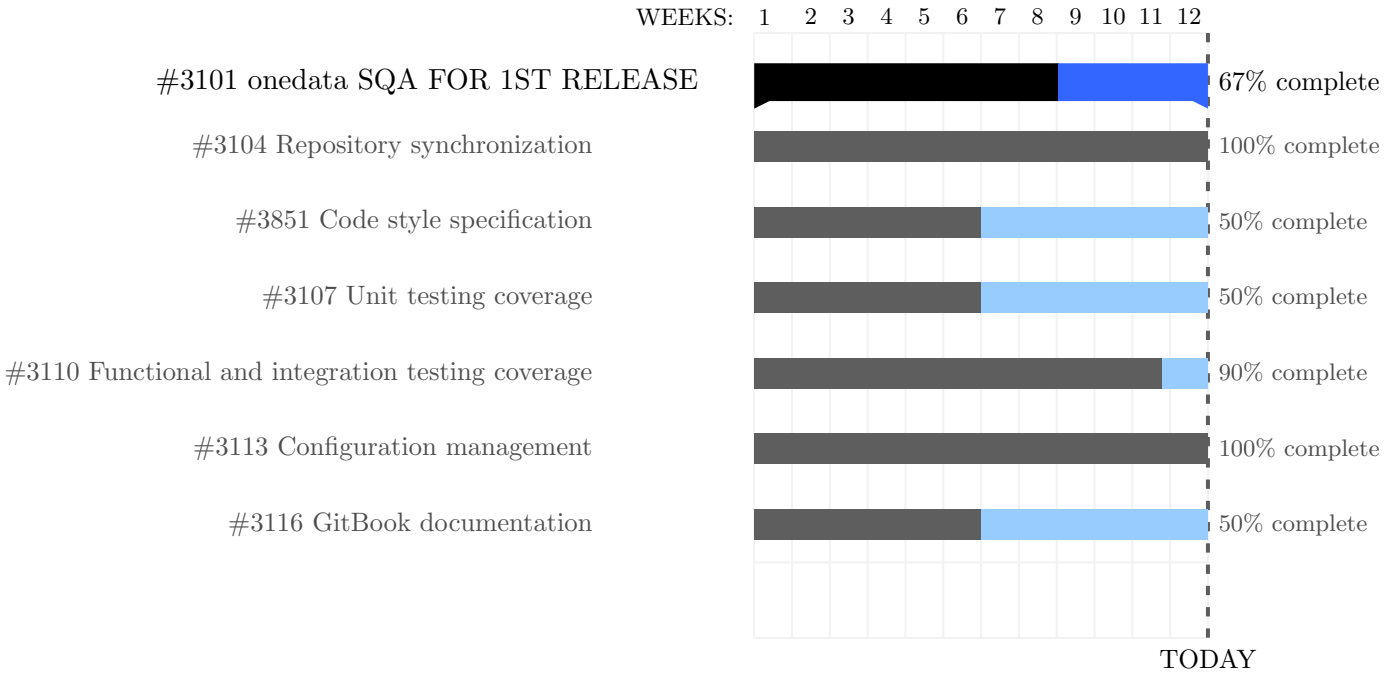
onedata

SQA Progress Status COMPLETE

67% done

GitHub repository	COMPLETE
Code style adherence	IN PROGRESS - WP3
Code coverage	IN PROGRESS - WP3
Functional/integration testing	COMPLETE
GitBook documentation	COMPLETE
Automated deployment	COMPLETE

Part I
Task Progress for the 1st Release



1 Repository synchronization

Products contributing to INDIGO-DataCloud project must have their code avaiable under GitHub's `indigo-dc` organization.

Repository exists under `indigo-dc` GitHub organization:

- https://github.com/indigo-dc/onedata
- https://github.com/indigo-dc/onezone
- https://github.com/indigo-dc/oneclient
- https://github.com/indigo-dc/luma

2 Code Style

Products contributing to INDIGO-DataCloud project are expected to be adhered to a community or de-facto standard code style definition. Exceptions can be made to the selected standard. Custom style guides are

accepted but nonetheless not recommended.

Code style definition	The Dialyzer, a DIscrepancy AnalYZer for ERlang programs
Community/de-facto standard	Yes
Exceptions	0
Richness	18 Errors 0 Warnings 18 link

3 Unit Testing

Code coverage will be tracked for the INDIGO-DataCloud related products and must not decrease during the project's duration. Recommended threshold is 70%.

3.1 Observations

- Command/s to check unit testing coverage have been provided, but not currently working. Need support from product team.

4 Functional/Integration testing

Functional testing must cover at least the basic functionalities that the product was requested to fulfill within the INDIGO-DataCloud project scope. Integration testing must cover the interactions with other components. Both types of testing will be automated whenever feasible by integrating them in the project's continuous integration service.

No automated execution, reports are being provided.

4.1 Reports

Functional report/s available:

- owncloud report

4.2 Observations

- Command/s to check functional testing coverage have been provided, but not currently working. Need support from product team.

5 GitBook documentation

Product-related documentation must be uploaded to GitBook's `indigo-dc` central repository. Types of documentation includes a) Developer b) Deployment and Administration c) Command-line Interface (CLI) and Application Program Interface (API) d) User Documentation. All these types may not be applied for every product. Those products that offer functionalities out of the scope of INDIGO-DataCloud project needs may not provide all the spectrum, but links to the official documentation.

Documentation available under `indigo-dc` GitBook organization:

https://indigo-dc.gitbooks.io/onedata-documentation/content/

5.1 Types of documentation currently provided

Readme | User documentation | Administrator documentation

5.2 Observations

- Points to improve in the documentation:
  - Could not find the way to install the component. For example trying to install `onezone` I followed: `Onedata for administrators > Onezone > Onezone Setup > download section`, linked to https://onedata.org/download#/home where I could not find any hint for installing the packages. Clicking in `get started` takes me back to GitBook.

6 Configuration Management

Those products released by INDIGO-DataCloud project that need to be deployed by the end user must rely on a maintained open-source configuration management tool to provide an automated means to install and configure the product. The recommended tool is `Ansible`.

6.1 Observations

- Quoting team comments:
  - When one installs oneprovider/onezone they really just install instance of onepanel. Oneprovider/zone are cluster (multinode) solutions. After installing oneprovider/zone on a number of nodes. User logs into onepanel on any of them, onepanel then detects other instances of onepanel running on other nodes. User uses onepanel to chose which services will be installed on which nodes.
  - Onepanel itself has no configuration variables when installed, hence the packages of oneprovider/onezone (and oneclient) has not configuration options on its own.
  - Onedata currently is being mainly tested and used using docker containers. We provide packages deb/rmps but we prefer for now that users focus on docker releases. Introducing another configuration layer on top of onepanel/docker containers would make the process of installing onedata even harder to grasp.
  - It is possible that in near future it will be possible to integrate our custom solution with ex. Ansible. For that reason, that task is being suspended for now.

Part II

How to read this document

1 Summary (front) page

Both the overall product's SQA adherence and per-task status codes are explained below:

COMPLETE	Task has been successfully completed and fulfills the project's SQA requirements, listed in Deliverable D3.1 and Extensions to Software Quality Assurance documents.
NOT COMPLETE	Task has not been completed, yet some missing required bits have not been provided.
IN PROGRESS	Task has not been completed, but can proceed as it is.
WP3 PENDING	Task has some pending work from WP3 side, meaning that the product team already submitted the required data but it has not been yet consumed by WP3.

2 Task Progress

2.1 Code style

Code style definition	Name and link of the standard to which the product is adhered.
Community/de-facto standard	Whether the adopted standard is community-wide accepted.
Exceptions	Number of exceptions from the standard definition.
Richness	Number of rules defined in the adopted standard. Additionally (whenever available) the number of errors, number of warnings documented in the standard will be displayed as well as the link to the latest definition.

2.2 Unit testing

This section will display the a) trend graph with the evolution of the code coverage over time and b) the Cobertura report, with the coverage results of different methods. Both are taken from the project's Jenkins continuous integration service.
Note: resultant coverage value is the lowest of the ones for the different methods: packages, files, classes, lines, conditionals.

2.3 Functional/Integration testing

2.4 GitBook documentation

Whenever the documentation of the product is available at the project's GitBook repository, both the a) **link** to the documentation index and b) **type of documentation** provided will be displayed in the report.

2.5 Configuration Management

Whenever the product has an recipe to be deployed automatically the following information will be available:

Tool	Configuration management tool used.
Manifest link	URL pointing to the manifest/s.
Deployment level	Whether `installation`, `configuration` or both.
Build status	Current build status for the project's supported OS distributions.