From license compatibility checks to software quality assurance with SQA

eScience Center - GFZ
19.09.2022

Publishing software with good quality

- there can be many criteria for a software to consider it as having good quality:
  - no license or author issues
  - comply to the REUSE Specification
  - good code quality (linting, best practices, static code analysis …)
  - …
- why not detect flaws automatically by analyzing the software repository?

> *Goal: implement a framework that offers tools to check the quality of a software project*

→ therefore the SQA project was started

What it does

- command-line tool
- get information about a given software
- software should be contained within a Git repository
- list used programming languages, authors and some metadata
- list licenses (only Python and JavaScript for now)
- check compliance to REUSE Specification
- add missing license headers to files if desired
- check for passwords/secrets committed to the repository
- integration in CI pipelines possible (via Docker image)
- generates a quality report

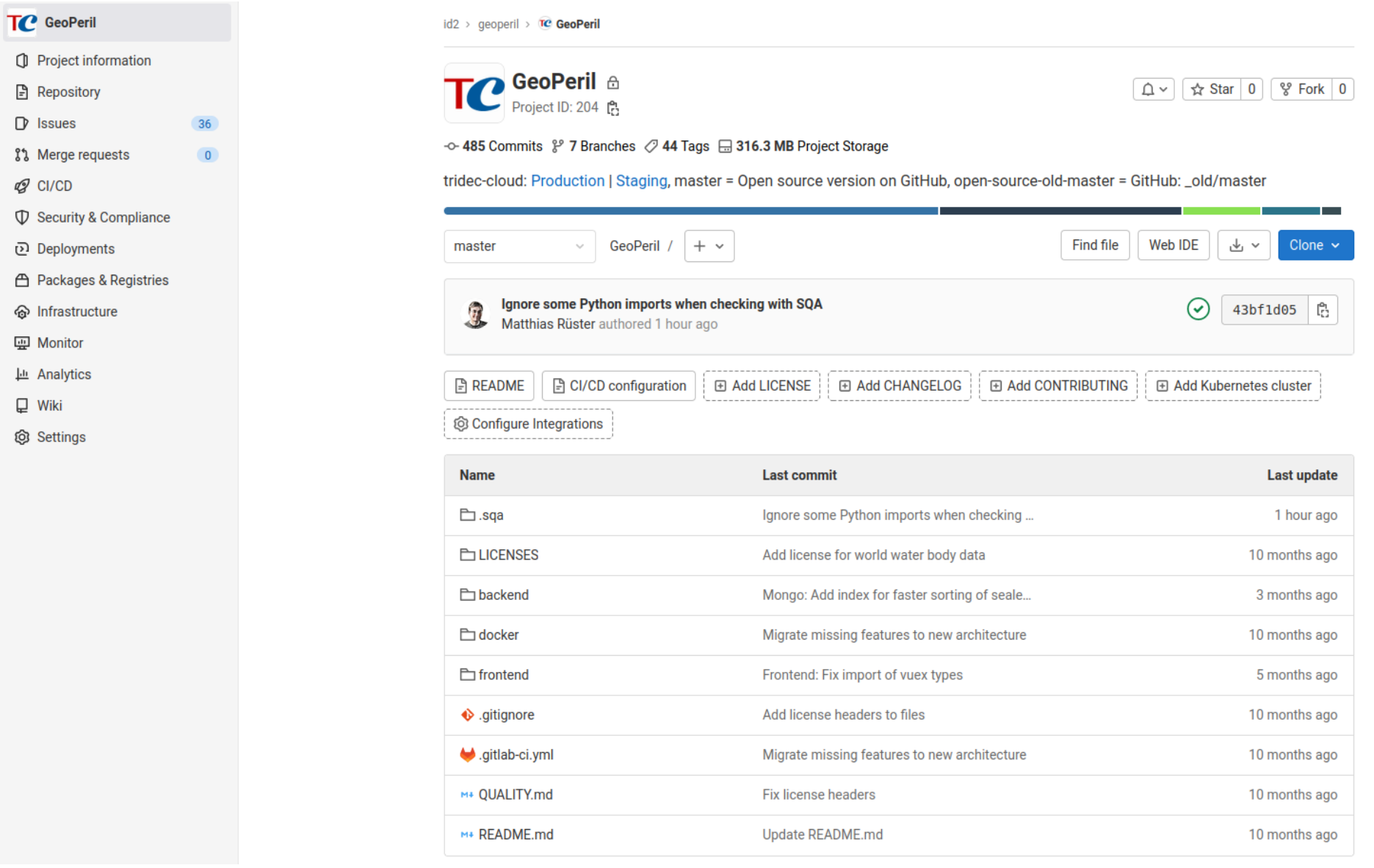
Reuse Specification

- SQA uses the reuse-tool to check the software
- SPDX license identifiers need to be used
- each file needs to have license information

```
# SPDX-FileCopyrightText: 2022 John Doe
#
# SPDX-License-Identifier: Apache-2.0
```

Example project

GIT REPOSITORY



Configuration for license check

```
description: 'A platform for the computation and web-mapping of hazard specific geospatial data, as well as for serving functi

ignore paths:
  - '.*\.git/.*'
  - '.*\.reuse/.*'

comment in file:
  - '.*\.sh$'
  - '.*\.py$'
  - '.*\.yml$'
  - '.*\.toml$'
  - '.*\.md$'
  - '.*\.css$'
  - '.*\.sh$'
  - '.*\.js$'
  - '.*\.ts$'
  - '.*\.xml$'
  - '.*\.csv$'
  - '.*\.gitignore$'
  - '.*\.gitmodules$'
  - '.*Dockerfile$'
  - '.*Makefile.*$'

licenses:
  default:
    copyright: 'Helmholtz Centre Potsdam - GFZ German Research Centre for Geosciences, Germany (https://www.gfz-potsdam.de)'
    spdx-identifier: 'Apache-2.0'
  Software:
    copyright: 'Helmholtz Centre Potsdam - GFZ German Research Centre for Geosciences, Germany (https://www.gfz-potsdam.de)'
    spdx-identifier: 'Apache-2.0'
    files:
      - '.*\.txt$'
      - '.*\.md$'
      - '.*\.yml$'
      - '.*\.gitignore'
      - '.*\.gitmodules'
      - '.*\.py$'
      - '.*\.sh$'
      - '.*Dockerfile$'
      - '.*\.css$'
      - '.*\.js$'
      - '.*\.ts$'
      - '.*\.vue$'
```

Configuration for Python dependencies

```yaml
exclude:
  - '.*\.gitlab-ci.yml$'
  - 'meta/.*'

manual dependency config:
  Python:
    - import name: 'base'
      pkg name: 'base'
      ignore: True
    - import name: 'basesrv'
      pkg name: 'basesrv'
      ignore: True
    - import name: 'msgsrv'
      pkg name: 'msgsrv'
      ignore: True
    - import name: 'processes'
      pkg name: 'processes'
      ignore: True
    - import name: 'surfer'
      pkg name: 'surfer'
      ignore: True
    - import name: 'data_products'
      pkg name: 'data_products'
      ignore: True
    - import name: 'auto_extent'
      pkg name: 'auto_extent'
      ignore: True
    - import name: 'build_cpt_file'
      pkg name: 'build_cpt_file'
      ignore: True
    - import name: 'build_legend'
      pkg name: 'build_legend'
      ignore: True
    - import name: 'tsunami_config'
      pkg name: 'tsunami_config'
      ignore: True
    - import name: 'bson'
      pkg name: 'pymongo'
    - import name: 'jsonschema'
      pkg name: 'jsonschema'
      licenses:
        - 'MIT'
    - import name: 'zipp'
      pkg name: 'zipp'
      licenses:
        - 'MIT'
```

Run SQA with command line

```
docker run -v /home/mruester/git/geoperil.git:/repo \
    -u $(id -u ${USER}):$(id -g ${USER}) \
    git.gfz-potsdam.de:5000/id2/software/services/fair/software-quality-assurance/software-quality-
assurance:latest \
    sqa run \
         --reuse_lint \
         --list_used_licenses \
         --list_used_languages \
         --list_authors \
         --check_credentials
```

Software Quality Report

> ***Note:*** *This report is automatically generated by Software Quality Assurance*

Table of Content

Meta Data

Software Name: GeoPeril

Software Location: git

Last Commit: 08df94eacb7406c66311c7b8da8090e1864540ee

Report Time: 30/05/2022 12:07:44 UTC

Report Version: 0.7.1

Completed Services

List Authors

> ***Description: Lists all people that committed to this config.***

- Hannes Fuchs
- Johannes Spazier
- Martin Hammitzsch
- Matthias Ruester
- Sebastian Jüngling
- Sven Reißland
- seth0r

List Used Languages

> ***Description: Detects used programming languages with linguist***

Language	Percentage
Python	54.36
Vue	26.48
Shell	8.41
TypeScript	6.46

QUALITY REPORT

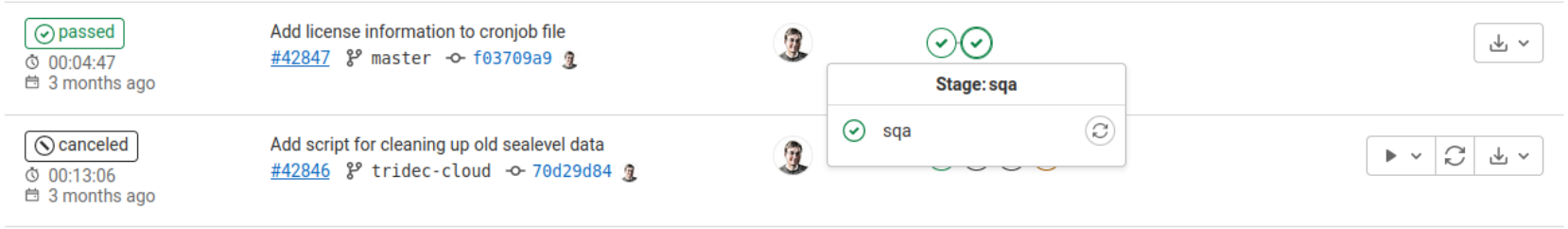
List Used Licenses

> ***Description: Generates a list of dependencies and their license.***

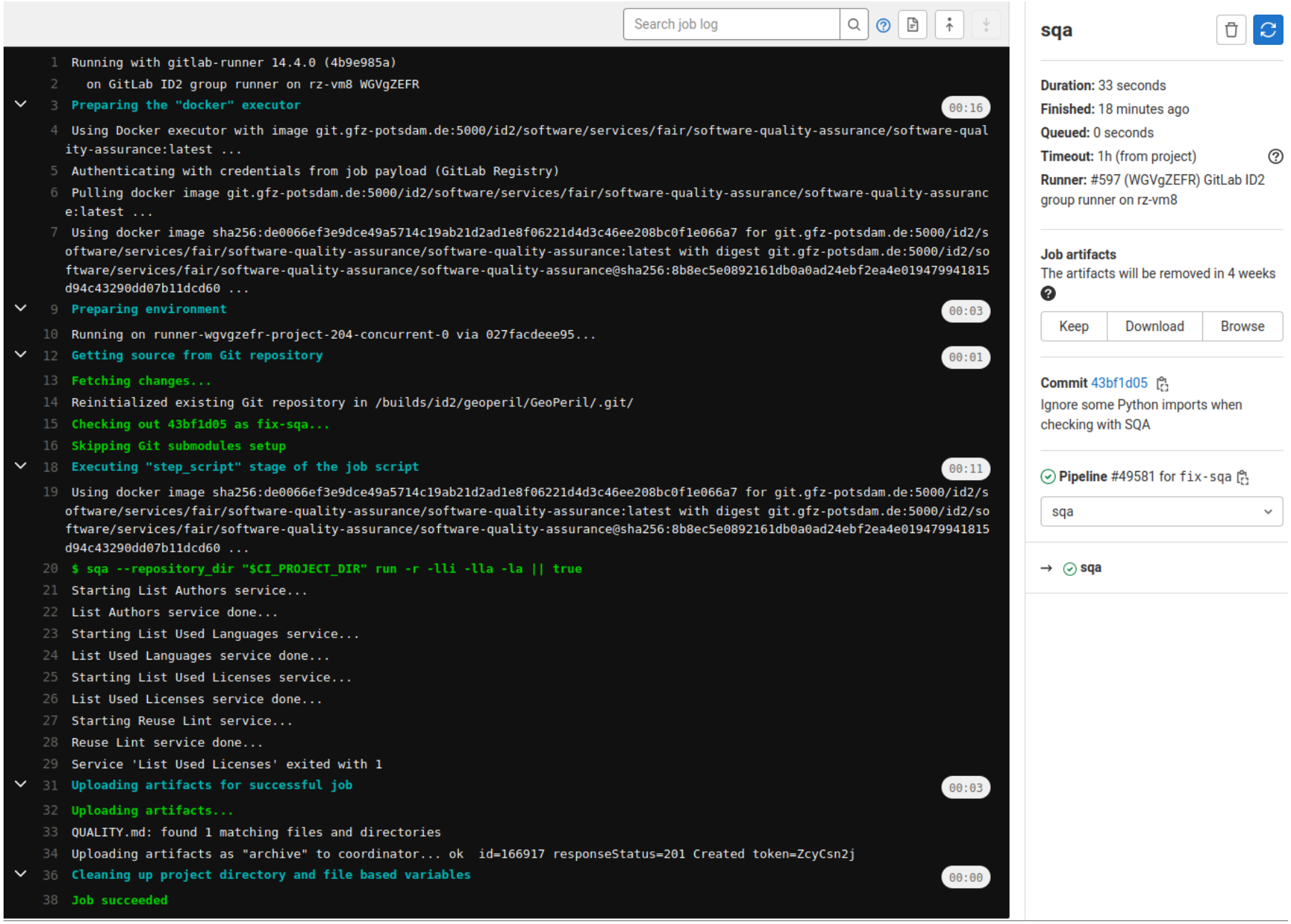
found licenses

license	libraries
Apache 2.0	Python / requests / 2.22.0 Python / requests / 2.24.0
Apache License, Version 2.0	Python / pymongo / 3.2
Apache Software License	Python / importlib-metadata / 2.0.0
BSD	Python / cherrypy / 3.8.2 Python / geopandas / 0.8.1 Python / itsdangerous / 1.1.0 Python / lxml / 4.6.5 Python / numpy / 1.21.5 Python / owslib / 0.20.0 Python / pandas / 1.1.2 Python / shapely / 1.7.1 Python / weasyprint / 51.0
BSD-2-Clause	Node / ol / 5.3.3
BSD-3-Clause	Python / click / 7.1.2 Python / flask / 1.1.2 Python / jinja2 / 2.11.2 Python / markupsafe / 1.1.1 Python / werkzeug / 1.0.1
BSD-like	Python / idna / 2.10
Dual License	Python / python-dateutil / 2.8.1
LGPL	Python / chardet / 3.0.4
LGPLv3	Python / flufl.enum / 4.1.1
MIT	Python / attrs / 20.2.0 Python / jsonschema / user config Python / pyproj / 2.6.1.post1 Python / pyrsistent / 0.17.3 Python / pytz / 2020.1 Python / pywps / 4.5.1 Python / pyyaml / 5.3.1 Python / six / 1.15.0 Python / sqlalchemy / 1.3.20 Python / urllib3 / 1.25.11 Python / zipp / user config
MIT License	Python / pathlib / 1.0.1
MPL-2.0	Python / certifi / 2020.6.20

Integration in GitLab CI



Integration in GitLab CI



Future plans

- support more programming languages (for licenses, linting …)
- offer a self-service for scientist to check their repository via web browser
- check for duplicate code
- check if tests exist
- static code analysis (vulnerabilities, best practices …)

Interested?

- SQA repository: https://git.gfz-potsdam.de/id2/software/services/fair/software-quality-assurance
- contact us: software-services@gfz-potsdam.de
- helpful links:
  - REUSE specification: https://reuse.software
  - SPDX license list: https://spdx.org/licenses