EOSC SYNERGY

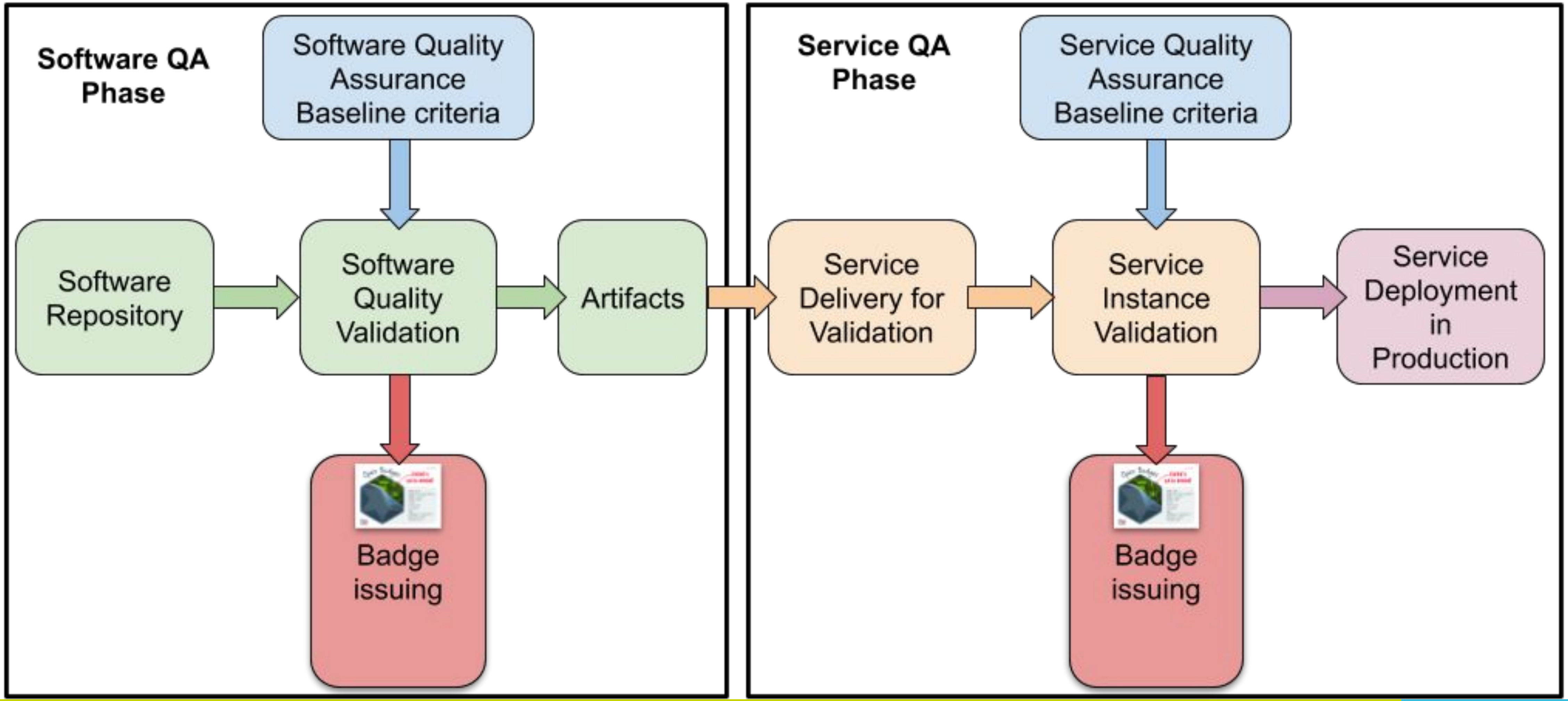
A quality based approach to software and services

EGI Conference 2020 - 3 Nov. 2020

Mario David and Pablo Orviz on behalf of EOSC-Synergy

EOSC-SYNERGY receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857647

Overview



Software QA Phase

Software Quality Assurance Baseline criteria

Software Repository

Software Quality Validation

Artifacts

Badge issuing

Service QA Phase

Service Quality Assurance Baseline criteria

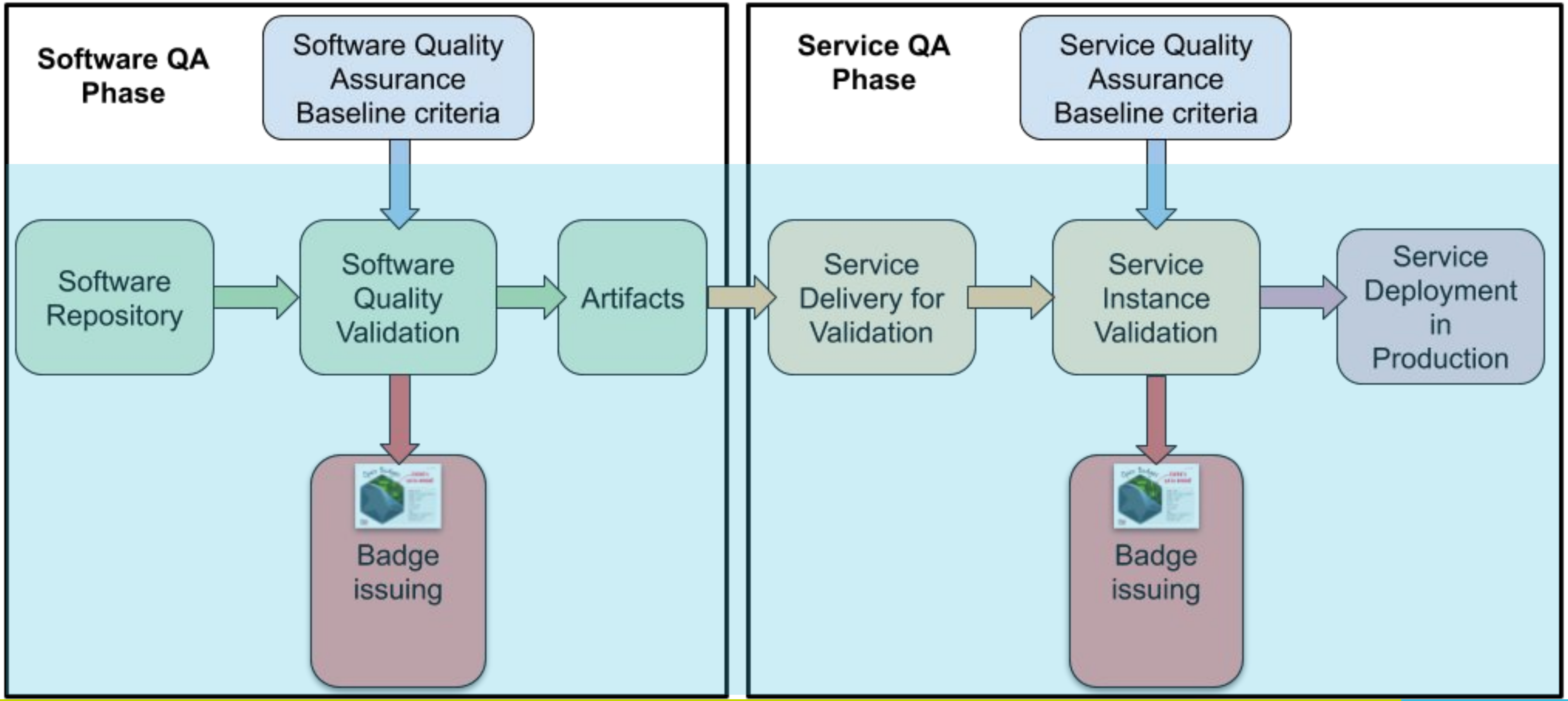
Service Delivery for Validation

Service Instance Validation

Service Deployment in Production

Badge issuing

Software and Service baseline criteria



Software QA Phase

Software Quality Assurance Baseline criteria

Software Repository

Software Quality Validation

Artifacts

Badge issuing

Service QA Phase

Service Quality Assurance Baseline criteria

Service Delivery for Validation

Service Instance Validation

Service Deployment in Production

Badge issuing

Software Quality Assurance

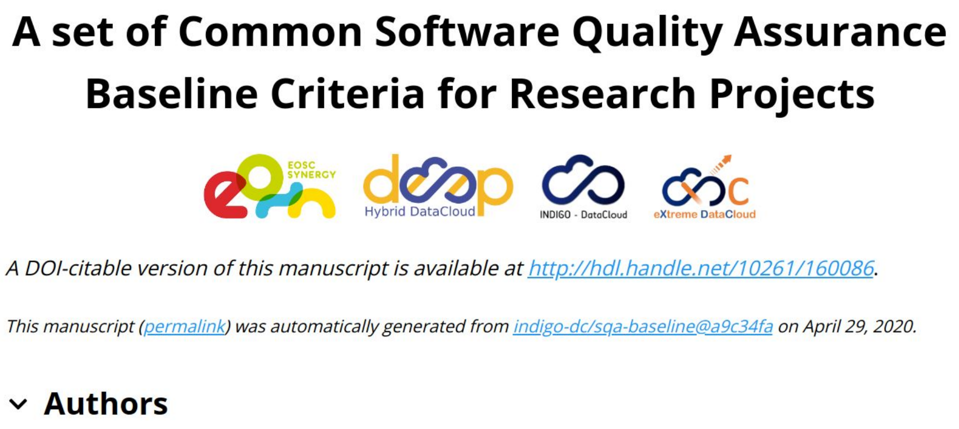
- SQA baseline doc is v3.2:
  - https://github.com/indigo-dc/sqa-baseline
  - https://indigo-dc.github.io/sqa-baseline/manuscript.pdf
    - doc in github
    - treated as code
    - discussions in "issues"
    - changes with PRs
    - autobuild:
      - when tag new release and pushed to "master"

- The criteria is designed towards automation ↪ translate this into an SQAaaS.
- Work ongoing for v3.3

Open

We accept contributions

http://hdl.handle.net/10261/160086



A set of Common Software Quality Assurance Baseline Criteria for Research Projects

A DOI-citable version of this manuscript is available at http://hdl.handle.net/10261/160086.

This manuscript (permalink) was automatically generated from indigo-dc/sqa-baseline@a9c34fa on April 29, 2020.

⌄ **Authors**

Software Quality Assurance

- Plan for v4.0:
  - New topics on SW release, maintenance and support
    - Appendices (if needed, KISS principle compliance) with examples of implementation from:
      - Indigo-Datacloud, DEEP-HybridDataCloud, eXtremeDataCloud (XDC) and **EOSC-Synergy**

We accept contributions

It's on the guidelines of:

"*EOSC Technical Specification Software Quality Assurance*"
↣ input to the **EOSC Architecture**

http://hdl.handle.net/10261/160086

A set of Common Software Quality Assurance Baseline Criteria for Research Projects

A DOI-citable version of this manuscript is available at http://hdl.handle.net/10261/160086.

This manuscript (permalink) was automatically generated from indigo-dc/sqa-baseline@a9c34fa on April 29, 2020.

⌄ **Authors**

Contextualization of a Service

A **Service**, as conceived in EOSC-Synergy, represent the following:

- **Web service**:
  - A web service is an application or data source that is accessible via a standard web protocol (HTTP or HTTPS).
  - Web services are designed to communicate with other programs, rather than directly with users.
  - Most web services provide an API, or a set of functions and commands, that can be used to access the data.
- **Web application**:
  - A web application or "web app" is a software program that runs on a web server.
  - Web apps must be accessed through a web browser.
- **Platform or Service Composition**:
  - Aggregation of multiple small services into larger services.
  - An integrated set of Web services, Web applications and software components.

Examples are: **Web Portals**, **Scientific Portals** and **Gateways**, **Data Service** and **Repositories,** the **Thematic Services**.

Service Quality Assurance

- Follow DevOps pragmatic approach:
    - Build from the outcomes and experience of SW Quality Assurance baseline
- **Services**: put in place processes and tools to:
    - Define and automatically validate **Software and Services Quality and Maturity**
    - Valid for **Thematic & Generic Services**
        - **Apply to EOSC-Synergy**
- **Data**: automated **verification of FAIR** data principles:
    - Collaboration with FAIRsFAIR results and outcomes
    - Through metadata analysis and,
    - Leveraging actionable features on data repositories

Service Quality Assurance

Open

We accept contributions

- ServiceQA baseline doc is v1.0:
  - Managed in github as the SQA baseline document:
  - https://github.com/EOSC-synergy/service-qa-baseline
  - https://eosc-synergy.github.io/service-qa-baseline/manuscript.pdf

  - The criteria is designed towards automation ↪ translate this into an SQAaaS.

http://hdl.handle.net/10261/214441

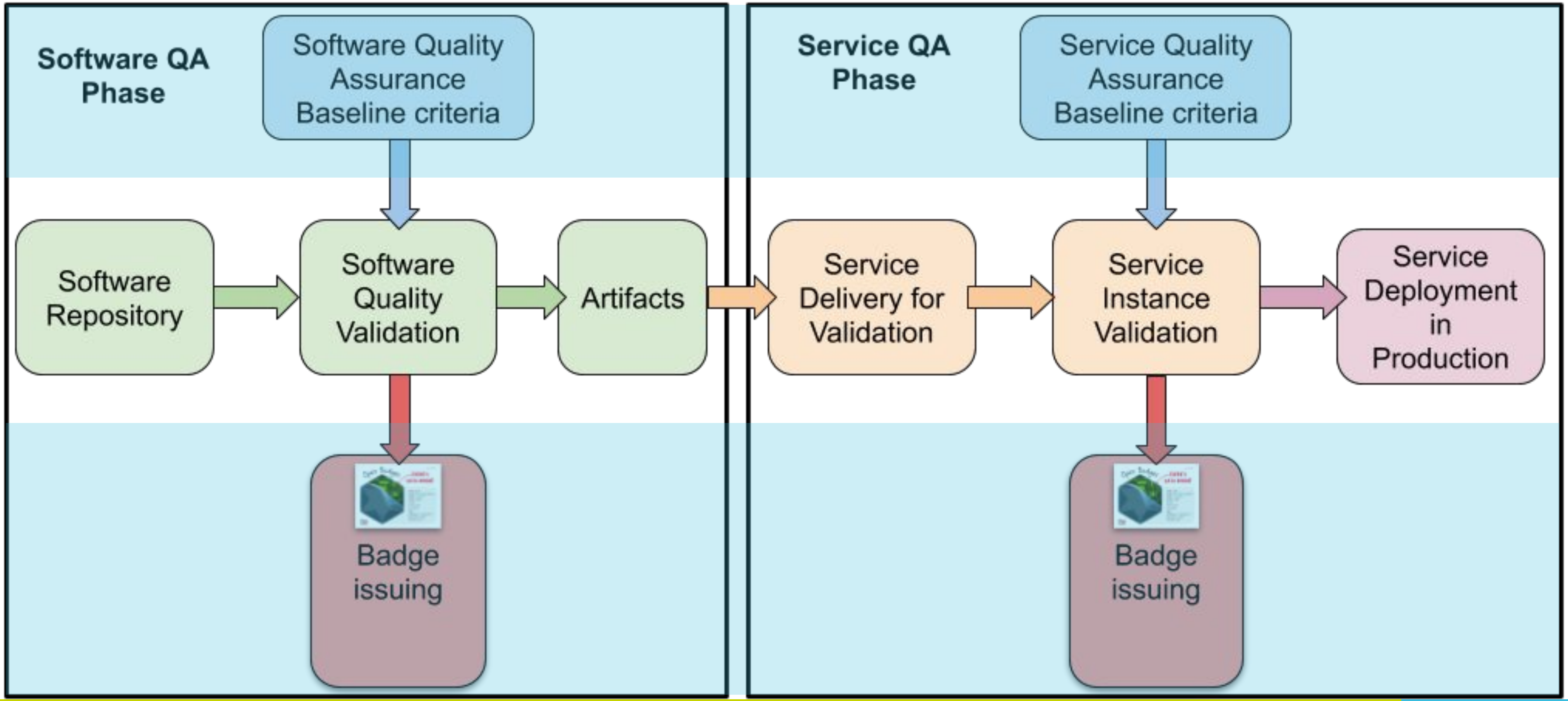
A set of Common Service Quality Assurance Baseline Criteria for Research Projects

A DOI-citable version of this manuscript is available at http://hdl.handle.net/.

This manuscript was automatically generated on 29-04-2020.

Authors

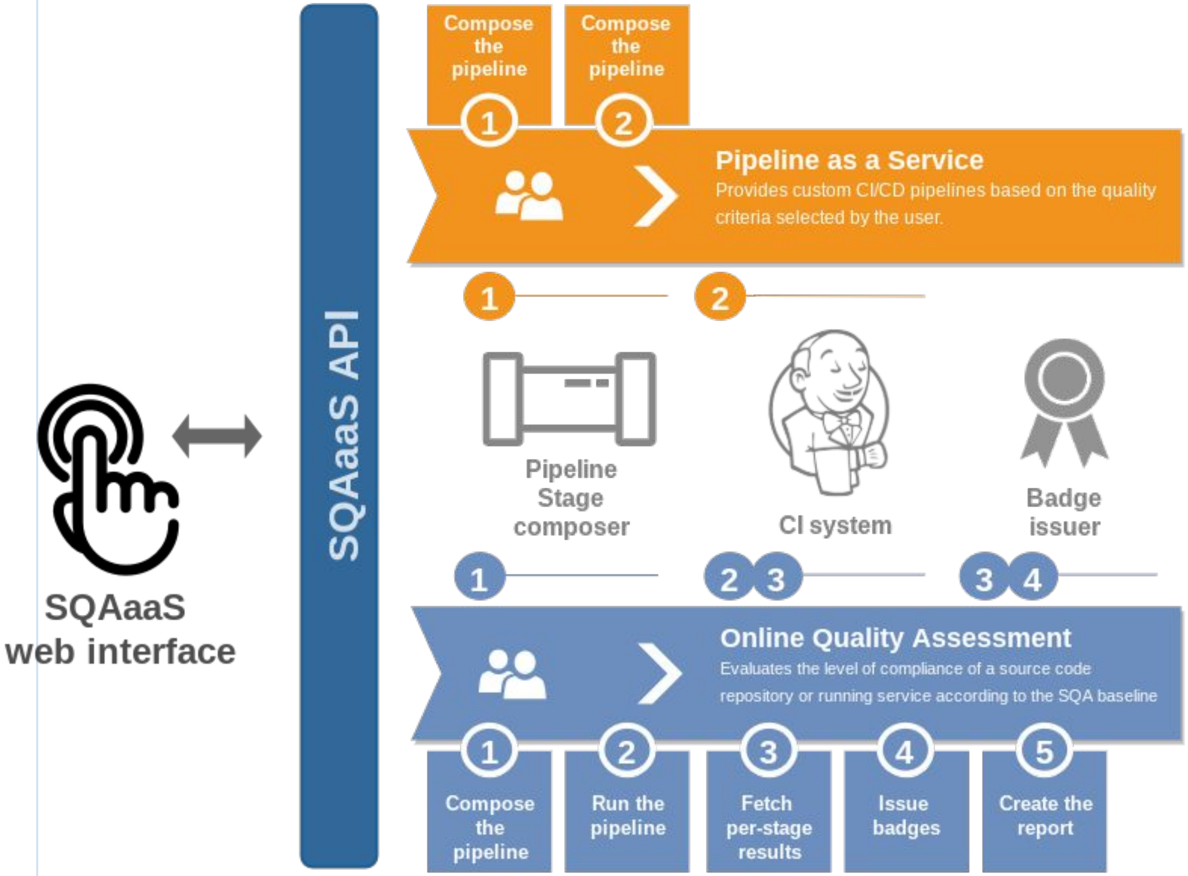
SQAaaS - The implementation


Software QA Phase
Software Quality Assurance Baseline criteria
Software Repository
Software Quality Validation
Artifacts
Badge issuing
Service QA Phase
Service Quality Assurance Baseline criteria
Service Delivery for Validation
Service Instance Validation
Service Deployment in Production
Badge issuing

SQAaaS implementation: JePL

SQAaaS architecture, internal components

- ***JePL - Jenkins Pipeline Library*:**
  - Leverages the Pipeline as Code capabilities from Jenkins framework.
  - Provides a library to be used by the Jenkinsfiles
- ***Jenkins CI system:***
  - Carries out the execution of the pipelines.
- ***Badge Issuing system:***
  - Framework that provides the issuing of digital badges according to the results from the pipeline execution.
- ***SQAaaS API:***
  - Manages the incoming requests and triggers the appropriate workflow.



SQAaaS web interface
SQAaaS API
Compose the pipeline 1
Compose the pipeline 2
Pipeline as a Service
Provides custom CI/CD pipelines based on the quality criteria selected by the user.
1
2
Pipeline Stage composer
CI system
Badge issuer
1
2 3
3 4
Online Quality Assessment
Evaluates the level of compliance of a source code repository or running service according to the SQA baseline
1 Compose the pipeline
2 Run the pipeline
3 Fetch per-stage results
4 Issue badges
5 Create the report

Yesterday's demo: JePL (Jenkins Pipeline Library)

SQAaaS implementation: JePL

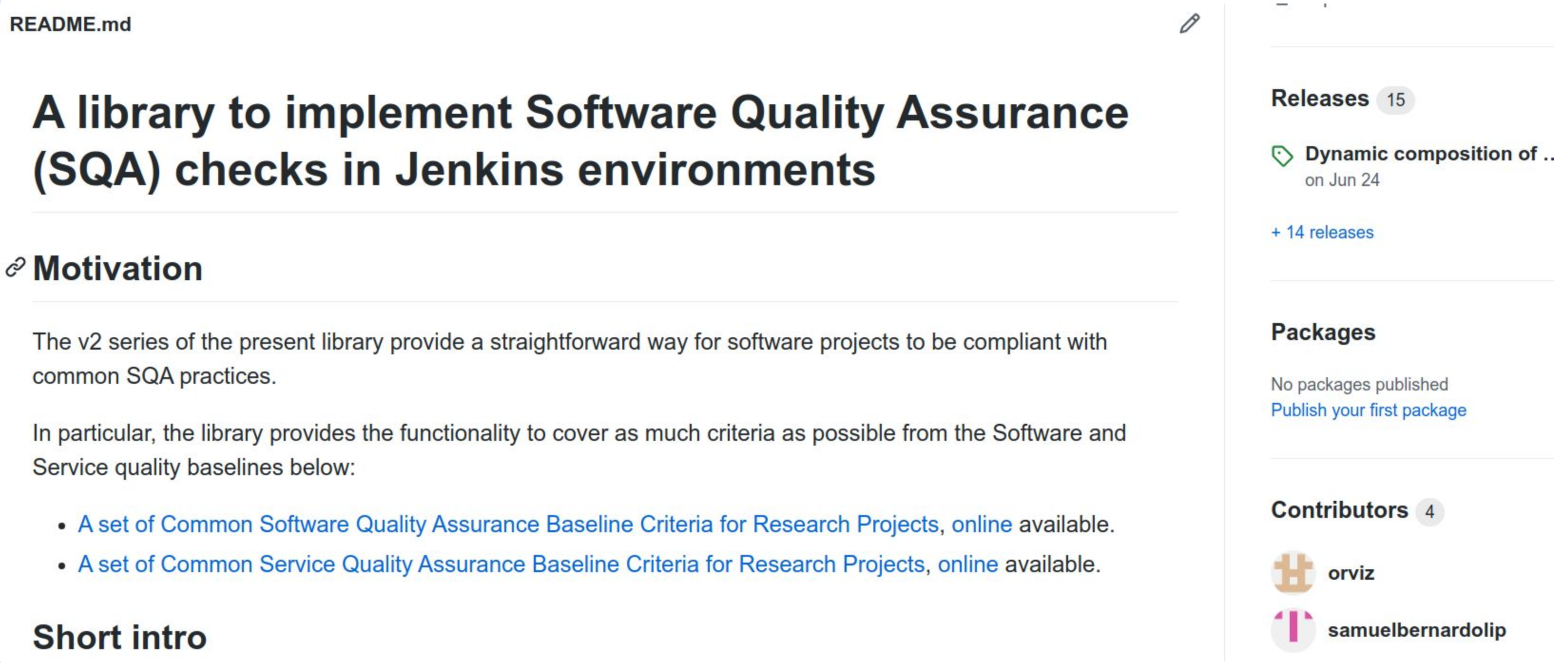
Software quality: Jenkins pipelines

- Verify criteria
- Produce artefacts
- Issue badges

Service quality: extending Jenkins pipelines

- Step beyond software quality
- Automated deployment
- Issue badges

https://github.com/indigo-dc/jenkins-pipeline-library



README.md

A library to implement Software Quality Assurance (SQA) checks in Jenkins environments

Motivation

The v2 series of the present library provide a straightforward way for software projects to be compliant with common SQA practices.

In particular, the library provides the functionality to cover as much criteria as possible from the Software and Service quality baselines below:

- A set of Common Software Quality Assurance Baseline Criteria for Research Projects, online available.
- A set of Common Service Quality Assurance Baseline Criteria for Research Projects, online available.

Short intro

Releases 15

Dynamic composition of ..
on Jun 24

+ 14 releases

Packages

No packages published
Publish your first package

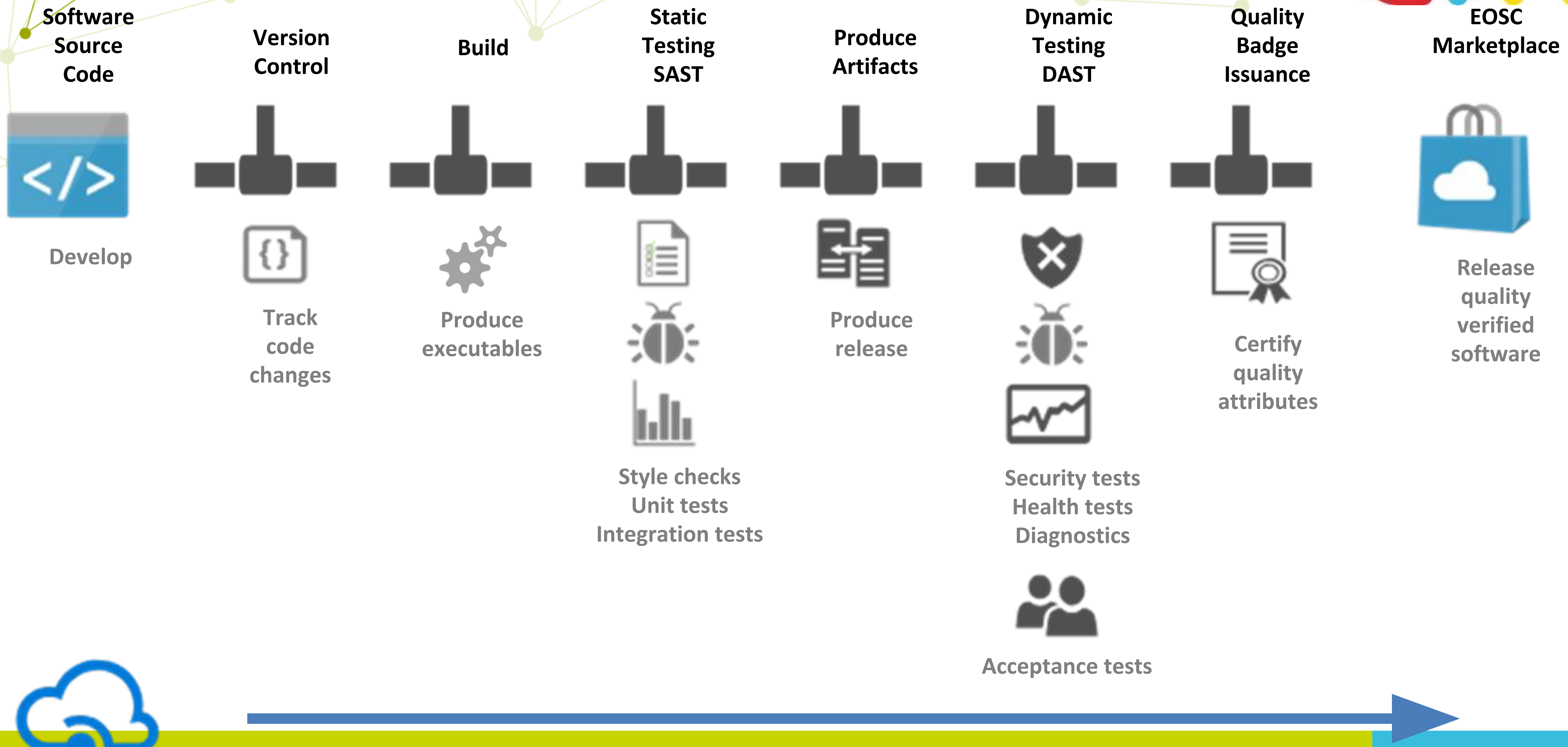
Contributors 4

orviz

samuelbernardolip

Yesterday's demo: JePL (Jenkins Pipeline Library)

SQAaaS implementation: JePL



Software Source Code

Develop

Version Control

Track code changes

Build

Produce executables

Static Testing SAST

Style checks
Unit tests
Integration tests

Produce Artifacts

Produce release

Dynamic Testing DAST

Security tests
Health tests
Diagnostics

Acceptance tests

Quality Badge Issuance

Certify quality attributes

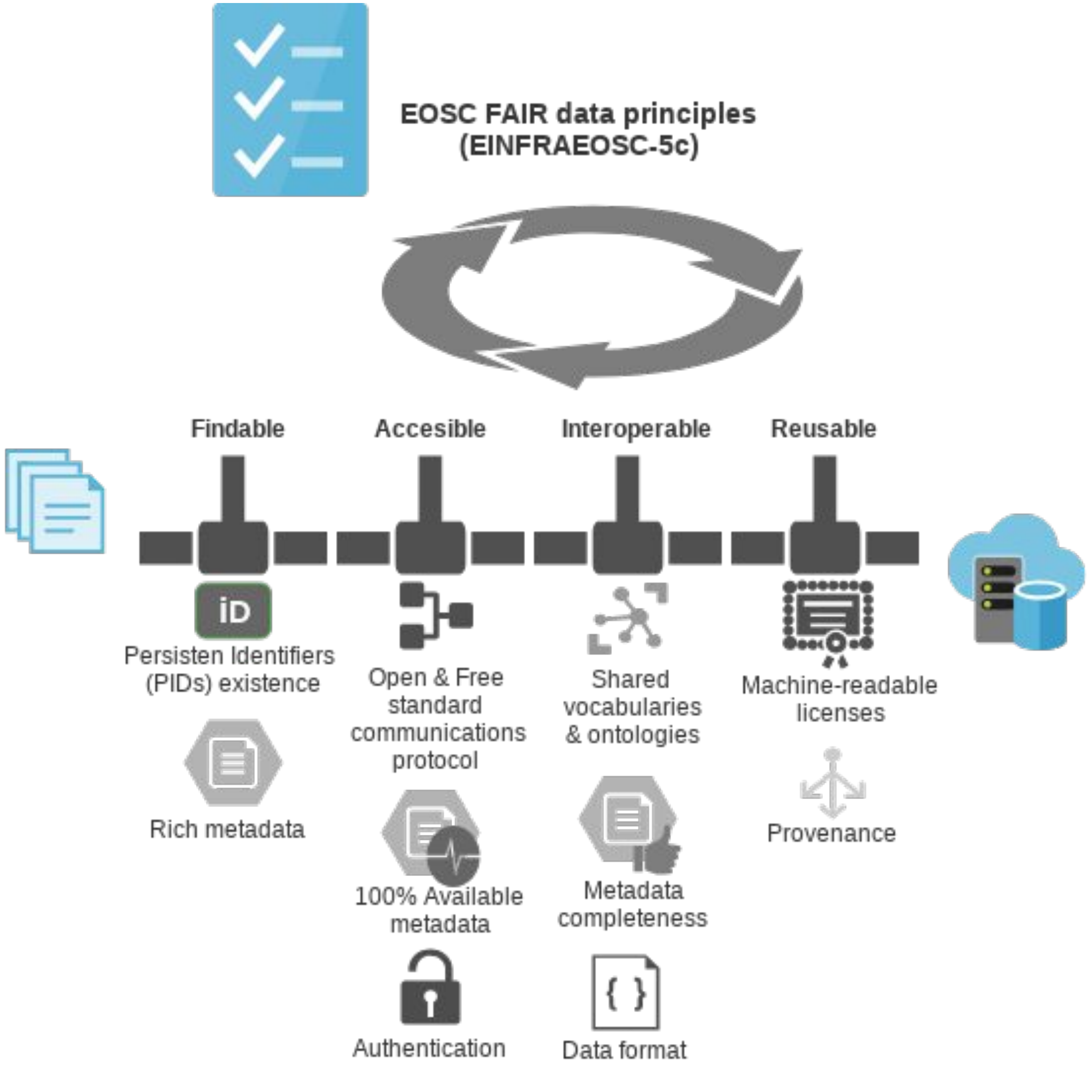
EOSC Marketplace

Release quality verified software

SQAaaS FAIRness criteria implementation - JePL

Data repository quality:

- Automated check of FAIR compliance (coming from FAIRsFAIR project)
- Issue badges



EOSC FAIR data principles (EINFRAEOSC-5c)

Findable

Accesible

Interoperable

Reusable

Persisten Identifiers (PIDs) existence

Rich metadata

Open & Free standard communications protocol

100% Available metadata

Authentication

Shared vocabularies & ontologies

Metadata completeness

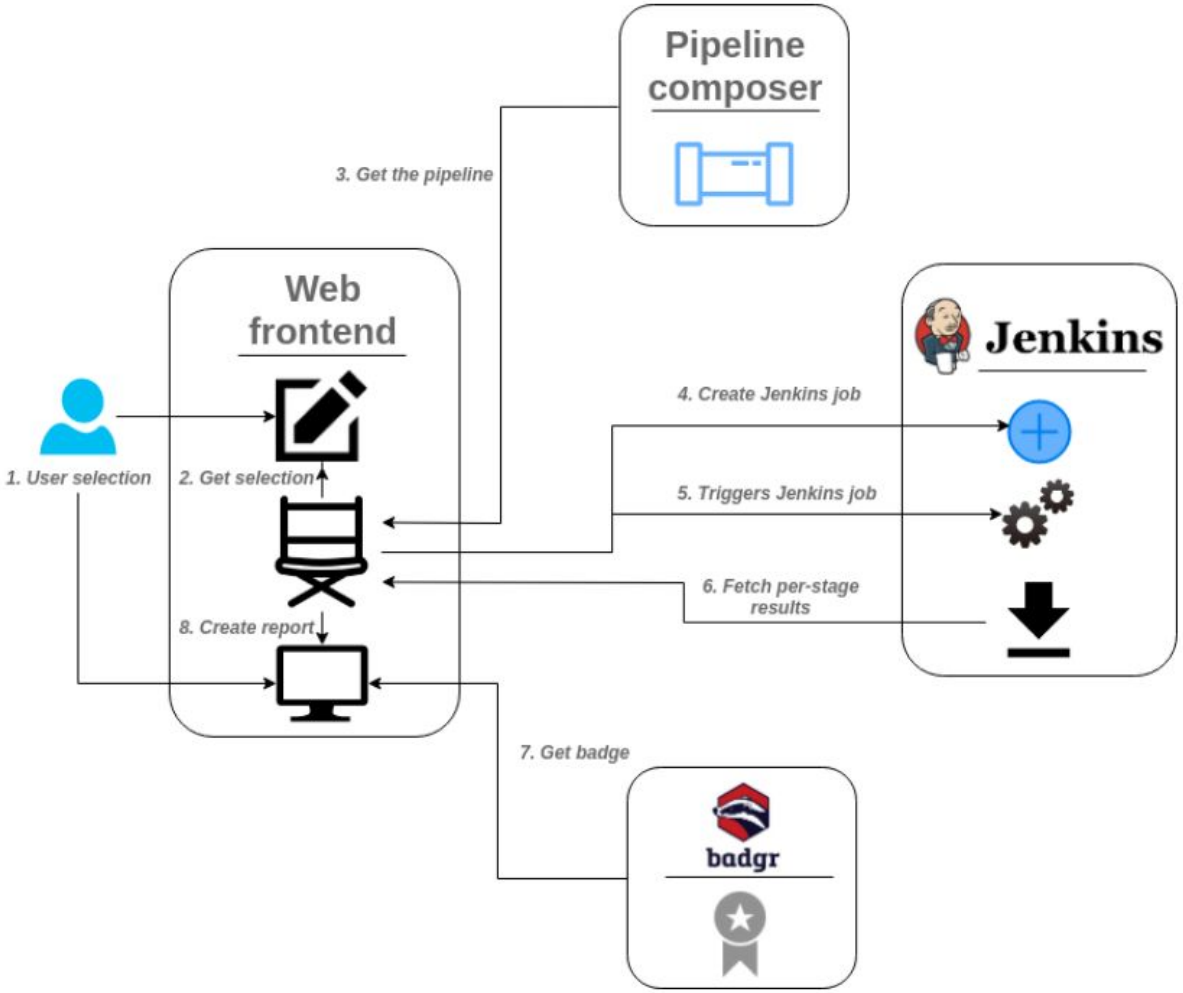
Data format

Machine-readable licenses

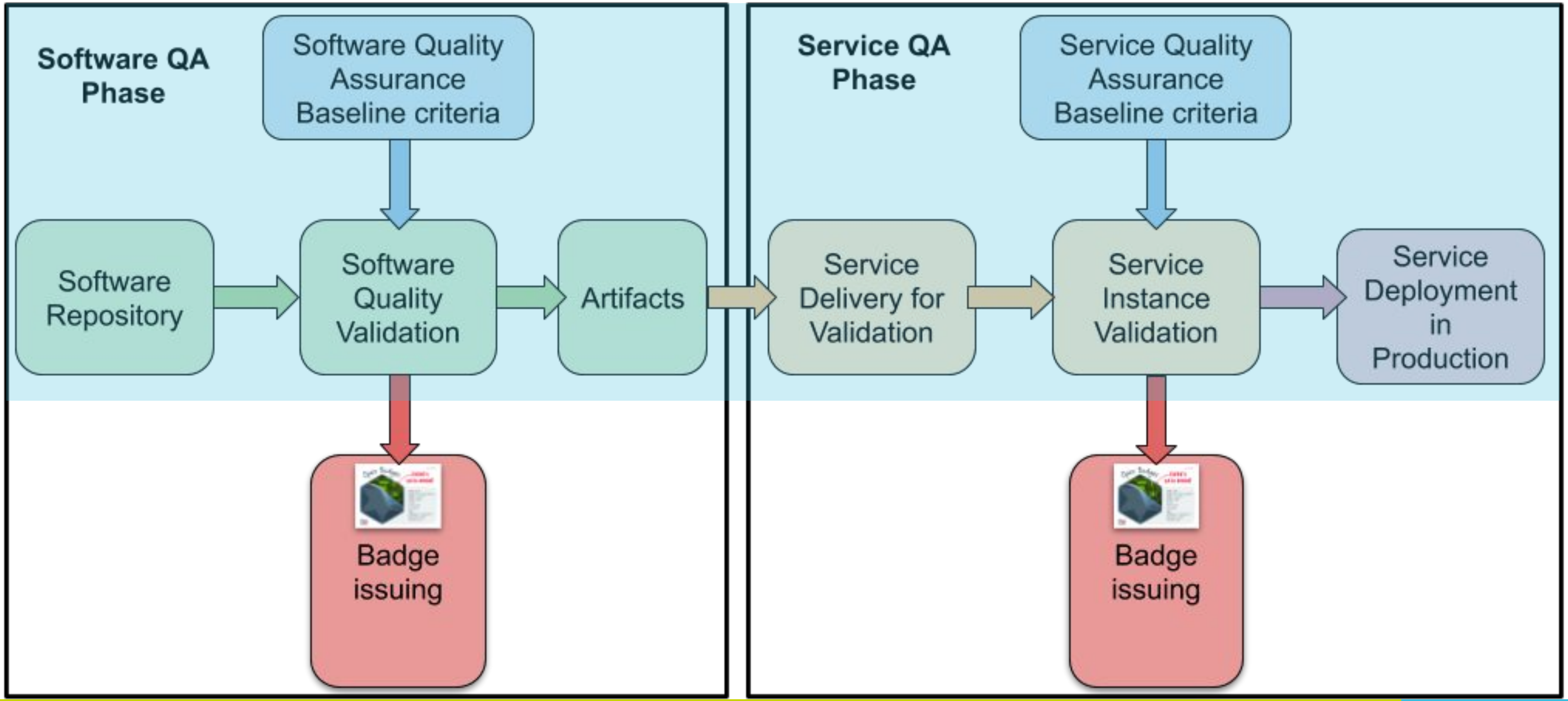
Provenance

SQAaaS - Technical implementation

- **On-demand quality assessment for:** Service software repositories and Service Instance
- **Making use of:** Quality criteria and Verification mechanisms
- **Requires:** User interface and further automation



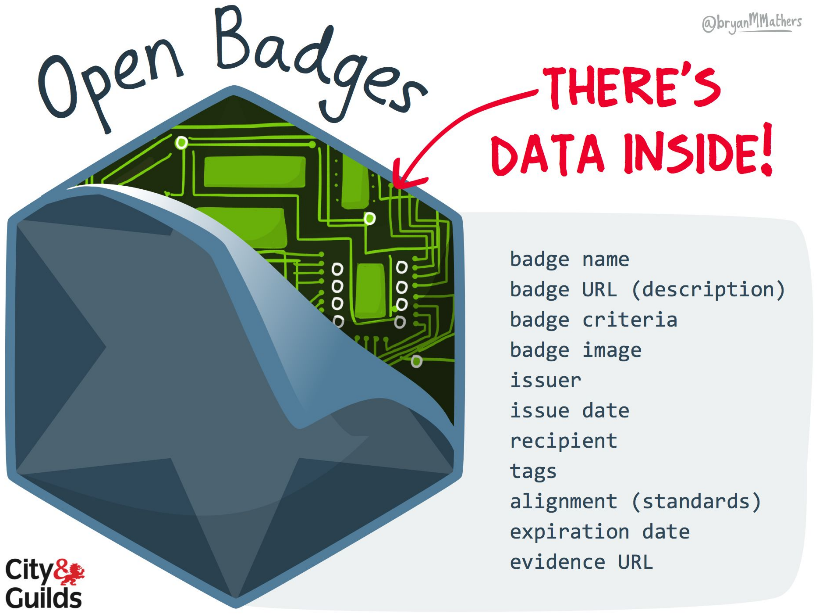
Badgr - Issuing badges


Software QA Phase
Software Quality Assurance Baseline criteria
Software Repository
Software Quality Validation
Artifacts
Badge issuing
Service QA Phase
Service Quality Assurance Baseline criteria
Service Delivery for Validation
Service Instance Validation
Service Deployment in Production
Badge issuing

Badges

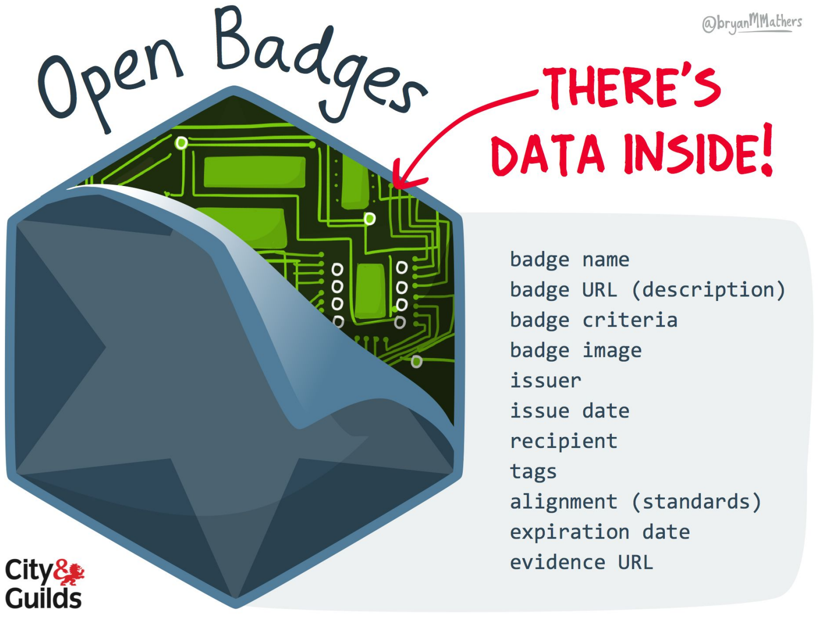
Digital badges represent virtual certificates that can easily be shared, watched and verified online:

- Recognition for software and services, compliant with the quality levels defined in the EOSC-Synergy project.
- Use OpenBadges specification:
  - Initial technology scouting whitepaper:
  - "State of the Art Regarding Digital Badge Issuing Technology"

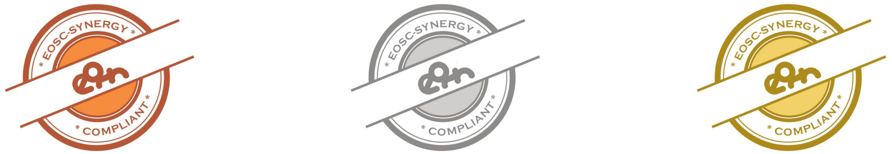


Badges

- The badges contain detailed metadata about achievements.
  - In the current case: passing the quality criteria
- Badgr service operational:
  - https://badges.eosc-synergy.eu/

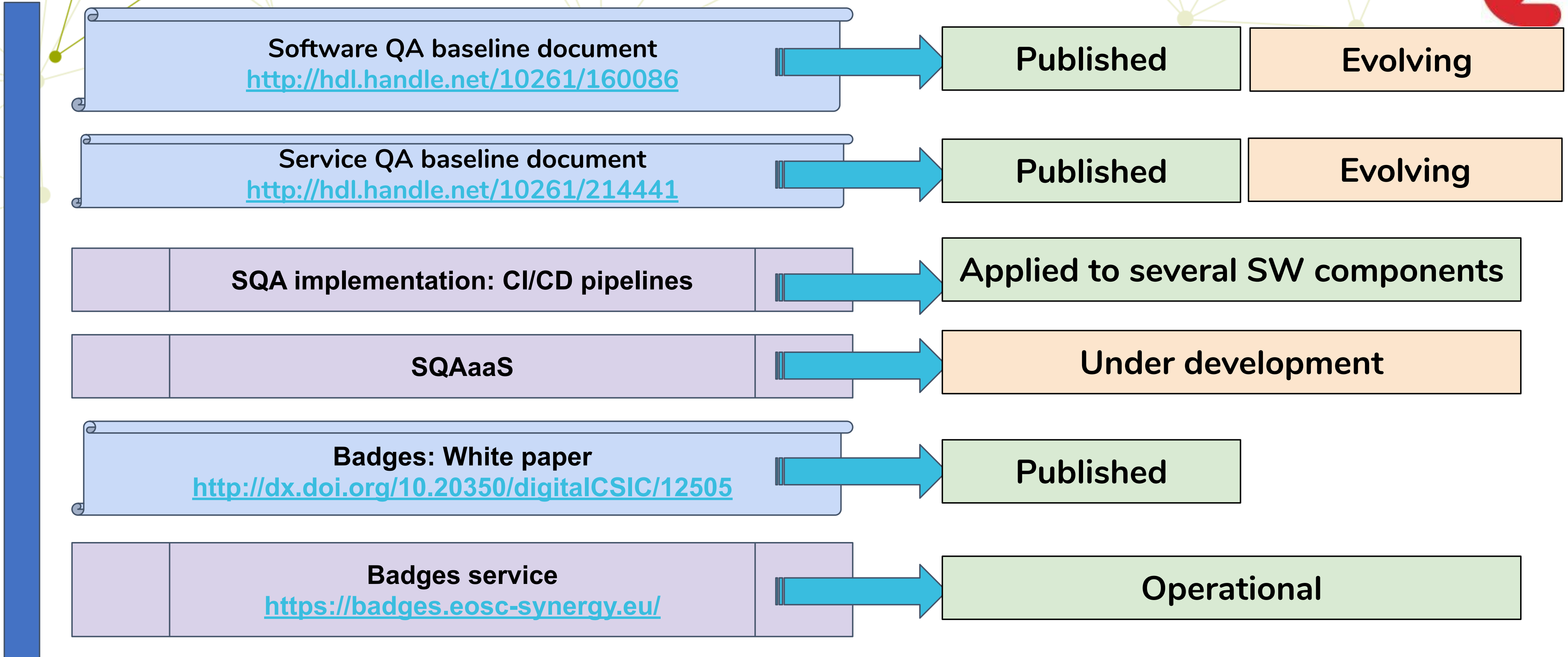


badge name
badge URL (description)
badge criteria
badge image
issuer
issue date
recipient
tags
alignment (standards)
expiration date
evidence URL



Final design with the three-level classifications (bronze, silver and gold)

Final remarks



Software QA baseline document
http://hdl.handle.net/10261/160086
→ Published | Evolving

Service QA baseline document
http://hdl.handle.net/10261/214441
→ Published | Evolving

SQA implementation: CI/CD pipelines
→ Applied to several SW components

SQAaaS
→ Under development

Badges: White paper
http://dx.doi.org/10.20350/digitalCSIC/12505
→ Published

Badges service
https://badges.eosc-synergy.eu/
→ Operational

Earth Observation
Monitoring of coastal changes and inundations, the processing of satellite image data and the estimation of forest mass, addressing different types of targets.

Environment
The monitoring and protection of ozone, the forecast of sand and dust storms, the simulation of water network distribution and untargeted mass-spectrometry analysis for toxics.

Biomedicine
Benchmarking platform for Genomic data processing tools and the processing of Cyron-electron microscopy imaging videos.

Astrophysics
The Latin-American Giant Observatory (a non-centralized and distributed collaborative network of more than 25 institutions of 9 Latin-American countries).

10 Thematic Services
1PB Scientific Data
>40K CPU hours per week

@eosc-synergy
www.eosc-synergy.eu

EOSC-synergy receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857647.

Thematic Service under test → WORSICA

Endpoints

- Jenkins deployed in INCD and available at
  - https://jenkins.eosc-synergy.eu
- GitHub and Dockerhub organizations:
  - https://github.com/EOSC-synergy
  - https://hub.docker.com/orgs/eoscsynergy
- Thematic pilot service
  - https://github.com/orgs/WorSiCa/dashboard
- OpenBadges services
  - https://badges.eosc-synergy.eu/
  - https://badges.eosc-synergy.eu:8443/

Thank you

For further information:

communications@eosc-synergy.eu

www.eosc-synergy.eu