

Introduction

Mario David (LIP) david@lip.pt

Importance of Software and Service Quality assessment in Research - I

- Quality assessment is an important trait for software and for services.
- It allows users and managers to have higher trust on the Software and Services during their use and operation:
 - They expect that the software and related services will work as supposed.
 - Give the expected results and meet their requirements.
- It also contributes to the maintainability, stability and sustainability of the software and services.

Importance of Software and Service Quality assessment in Research - II

- It contributes to facilitating the collaboration between software developers and promotes good practices for Software development.
- It promotes good practices for service development and operation.

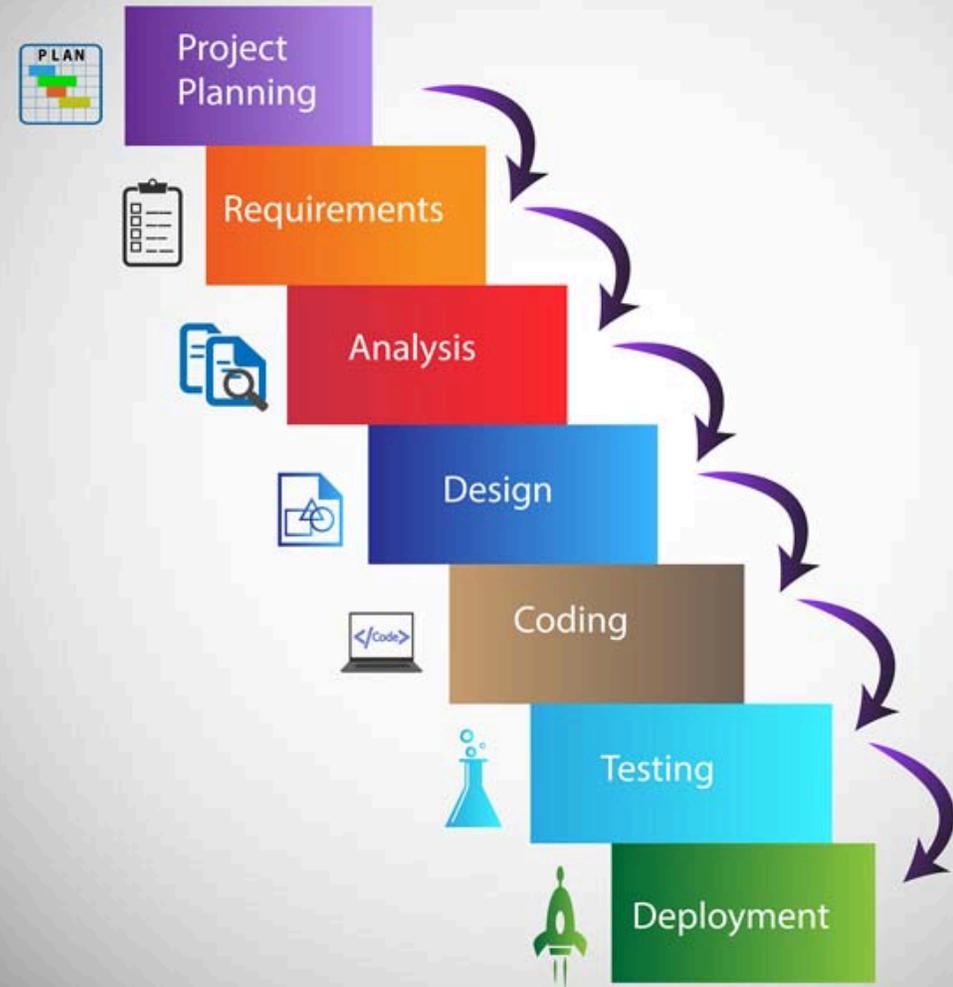


Software Development methodologies

SW development: In the past - I

As an example, one wide used SW development methodologies was the **Waterfall** model.

Waterfall-Model



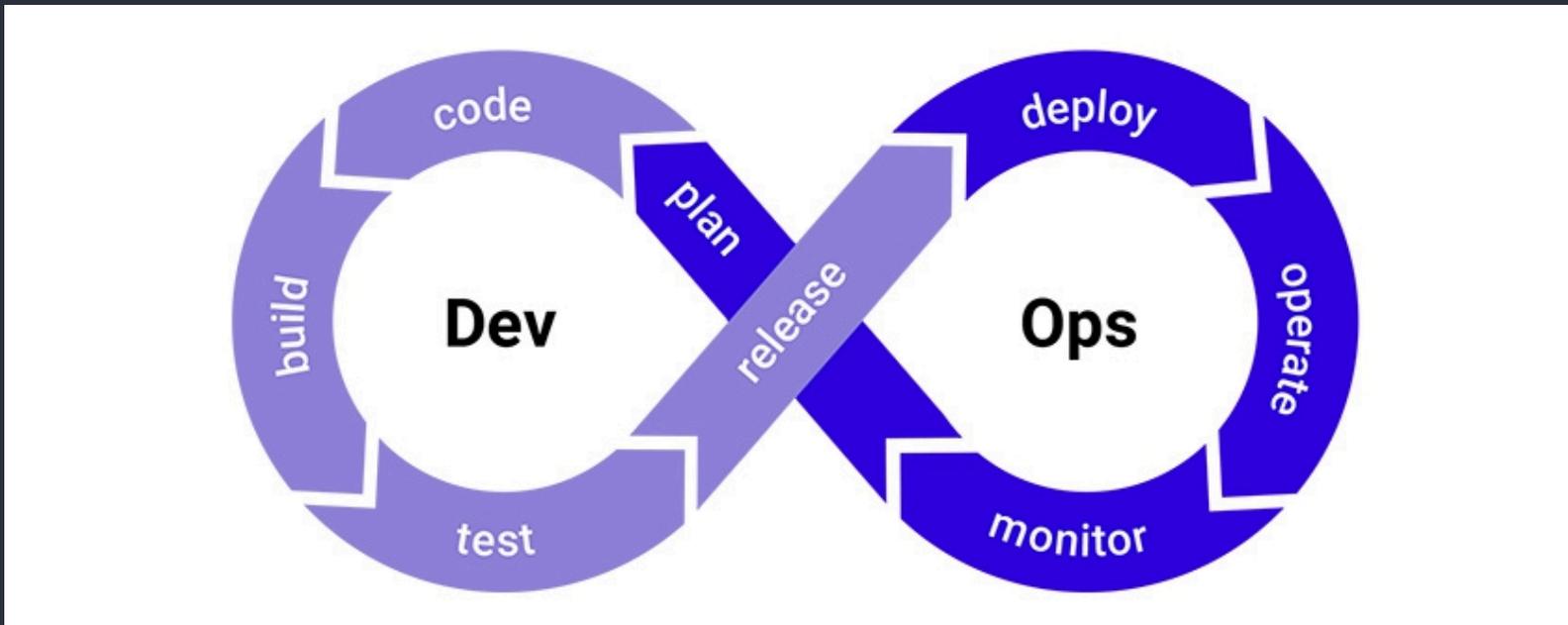
SW development: In the past - II

The project phases were characterized by:

- Long periods, both between phases and the overall process.
- Rigid and linear approach.
- Each phase has distinct goals.
- Once a phase is completed, there is no turning back.
- Does not allow room to accommodate inevitable changes.

DevOps methodology - I

- Is a set of **good practices**.
- Supported by **Continuous Integration/Continuous Delivery/Continuous Deployment - CI/CD(D)**.
- Enhances collaboration between the departments or groups.



DevOps methodology - II

It's characterized by the following high level phases:

- Development: *plan* → *design* and *code*.
- Quality assurance: *build, test, release* and *deployment* → CI/CD(D).
- Operations.

Short periods between phases → Fast + Automation.

What are Quality Models - I

- Software product quality models assess the quality properties of software products.
- A quality model is a set of Quality properties, or Quality Criteria that can be assessed for a given Software product.
- The most relevant quality model, is the standard defined in the ISO/IEC 25010:2011(en) Systems and software engineering, denoted: "Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models".

What are Quality Models - II

- In the framework of European projects, these are based on Maturity Levels of the Software or service. The EOSC-hub guide proposes characteristics to help assess the maturity of a service via the operational definition of the Technology Readiness Level (TRL) indicators, TRL, 7, 8 and 9:
- TRL 7 - Beta: "System prototype demonstration in operational environment".
- TRL 8 - Production: "System complete and qualified"
- TRL 9 - Production: "Actual system proven in operational environment"

What are Quality Models - III

CESSDA's Software Maturity Model (SMM):

- An approach for assessing the maturity of the components of the technical Research Infrastructure (RI)
- The SMM grade is based on the Reuse Readiness Levels (RRLs), as developed by NASA Earth Science Data Systems.
- Each criteria is graded with 5 levels.
- Documentation, Intellectual Property, Extensibility, Modularity, Packaging, Portability Standards Compliance, Support, Verification and Testing, Security, Internationalization and Localization, Authentication and Authorization.

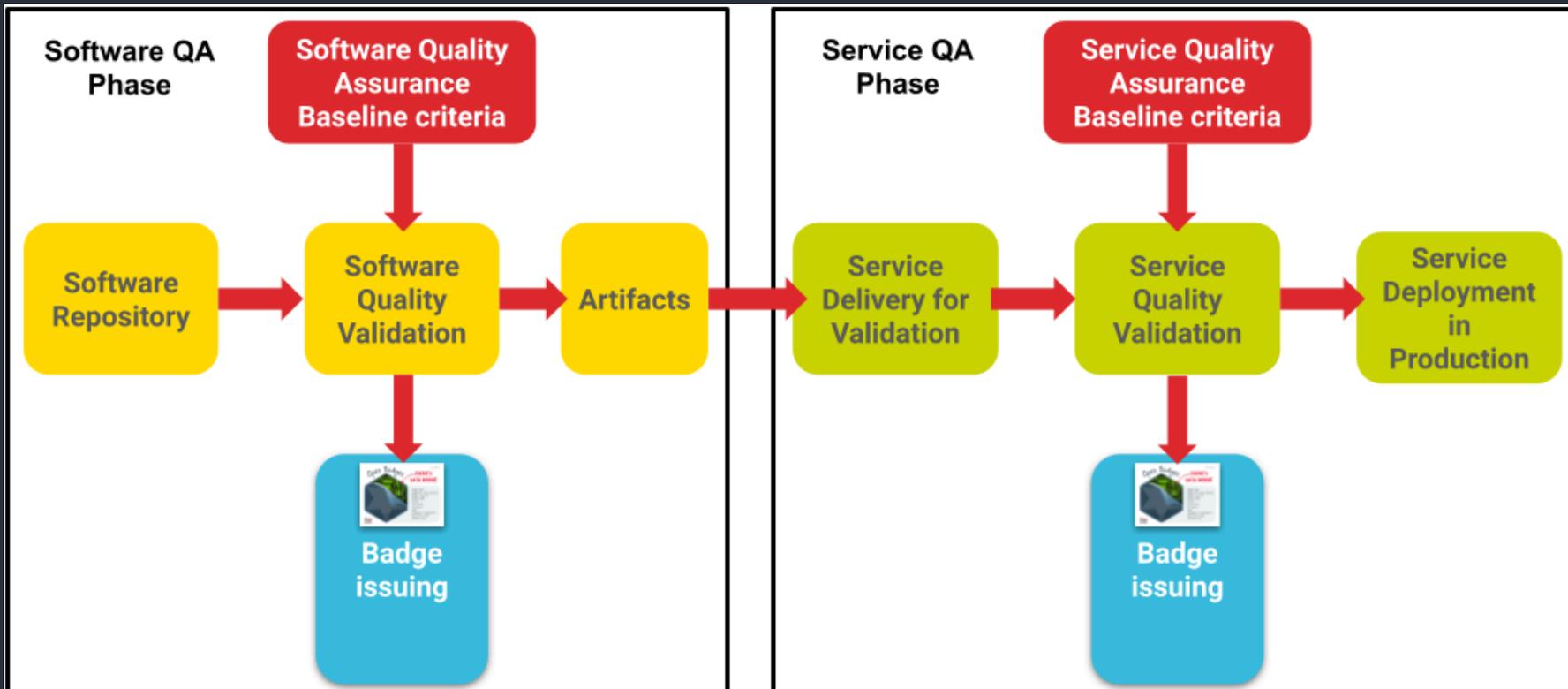
What are Quality Models - IV

DevOps approach links development and operations for software components:

- Use of a Continuous Integration and Continuous Delivery pipelines - CI/CD.
- The CAMS model stands for Culture, Automation, Measurement and Sharing, which are named as the four-fundamental dimensions to enable DevOps.
- It does not have a single standard, but takes the best practices from several standards.

Vision and high level architecture - I

The vertical view axis represent the high level architecture of the Quality Assurance developed in EOSC-Synergy.



Vision and high level architecture - II

1. **Top layer (red):** Quality Assurance baseline documents: Detail a set abstract quality metrics.
2. **Middle layer (yellow/green):** SQA as a Service (SQAaaS): Implement Quality assessment based on the baselines documents.
3. **Bottom layer (blue):** Issue badges: as a proof that a given Software or Service has passed the Quality Assurance criteria.

The horizontal axis, represents the practical implementation:

- The left hand side: the box represent the **Software** QA phase
- The right hand side box representing the **Service** QA phase.

Continuous Integration CI

- CI → Continuous Integration:
 - Coding.
 - Building: includes Automation → produces artifacts.
 - Testing: includes Automation and SW Quality Assurance (of produced artifacts).
- Code Review: manual step, comments/approval/voting by partners/colleagues/contributors.

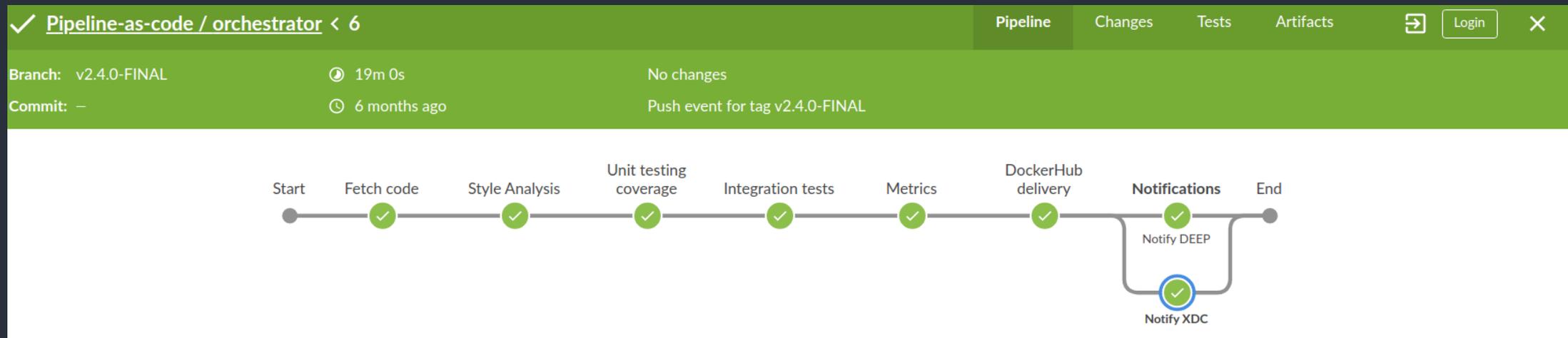
Continuous Delivery/Continuous Deployment CD(D)

- CD → Continuous Delivery → Deployment:
 - Delivery: Artifacts are released - ready for usage (by users or system administrators) in production.
 - Deployment: refers mainly to services → Installation, configuration, service (re)start.

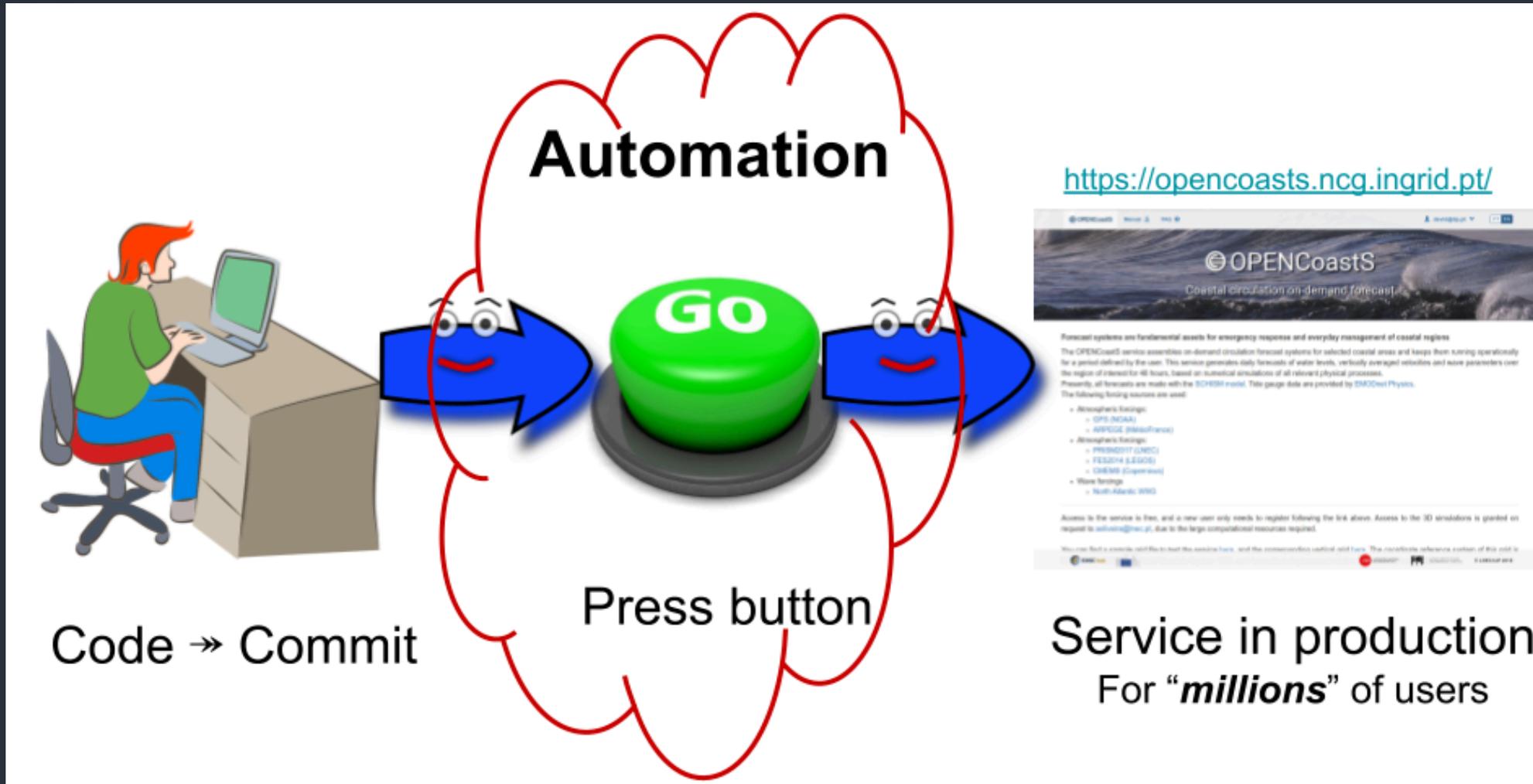


CI/CD(D)

A real example: Jenkins CI/CD pipeline.



DevOps and automation - I



DevOps and automation - II

Scary but in the real world/practice not *all* steps or phases are automatic or automated. For example:

- "Code Review" is a manual step.
- In many cases - the process is stopped in the "Delivery" step.
- In many cases - the automated deployment is preformed in a "Staging" or "Pre-production" or "Preview" infrastructure, (not on the production service/infrastructure).

References - I

- Mario David presentation 2019
https://docs.google.com/presentation/d/1SZbqpU3LLKY6ye-zYNgxjLSNDX5CpldxJ_Y0lo6vn0/edit#slide=id.g3d2cbff6ef_1_64
- EOSC-Synergy Deliverable D3.1 <https://digital.csic.es/handle/10261/219306>
- Galli, Tamas, Francisco Chiclana, and Francois Siewe. "Software Product Quality Models, Developments, Trends and Evaluation." SN Computer Science, 1:154, (2020). <https://doi.org/10.1007/s42979-020-00140-z>
- ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models: <https://www.iso.org/standard/35733.html>

References - II

- Humble, J. and Farley, D. (2011), Continuous delivery: Reliable software releases through build, test, and deployment automation, A Martin Fowler signature book, Addison-Wesley, Upper Saddle River.
- König, Leon, and Andreas Steffens. "Towards a quality model for devops." Continuous Software Engineering & Full-scale Software Engineering (2018): 37.
- EOSC-hub Service Maturity Classification: <https://eosc-portal.eu/providers-documentation/eosc-provider-portal-resource-maturity-classification>
- John Shepherdson, CESSDA Software Maturity Levels (2019), DOI: 10.5281/zenodo.2591055