interTwin

D3.2 DTE First software release

Status: APPROVED BY THE EC

Dissemination Level: public

Funded by the European Union

Disclaimer: Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

Abstract

Key Words Software, Quality, Release, Assessment

The document describes the first interTwin DTE release along with all necessary steps towards it. It includes the Software release procedure, the Quality Criteria (mandatory and optional) required for each component, how each component is tracked and finally the release itself alongside some statistics about the number of components and their status.

Document Description			
DTE First Software Release			
Work Package number 3			
Document type	Deliverable		
Document status	APPROVED BY THE EC	**Version**	1
Dissemination Level	Public		
Copyright Status	cc BY This material by Parties of the interTwin Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead Partner	LIP		
Document link	https://documents.egi.eu/document/3931		
DOI	https://zenodo.org/records/14974042		
Author(s)	• Mario David (LIP) • Joao Pina (LIP) • Andrea Manzi (EGI Foundation)		
Reviewers	• Andrea Manzi (EGI Foundation) • Christoph Reimer (EODC)		
Moderated by:	• Sjomara Specht (EGI)		
Approved by	AMB		

Revision History

Version	Date	Description	Contributors
V0.1	06/02/2024	ToC	Mario David, Joao Pina (LIP)
V0.2	26/02/2024	Complete draft	Mario David, Joao Pina (LIP), Andrea Manzi (EGI Foundation)
v0.3	05/03/2024	Version reviewed by internal reviewers	Andrea Manzi (EGI Foundation) Christoph Reimer (EODC)
v0.4	05/03/2024	Version ready for QA	Mario David, Joao Pina (LIP)
V1.0	06/03/2024	**Final**	

Terminology / Acronyms

Term/Acronym	Definition
CI/CD	Continuous Integration/Continuous Delivery
DT	Digital Twin
DTE	Digital Twin Engine
QCA	Quality Criteria Attributes
SQAaaS	Software Quality Assurance as a Service
SW	Software

Terminology / Acronyms: **https://confluence.egi.eu/display/EGIG**

Table of Contents

1 Introduction 7

2 Software and Services Quality Assurance 8

2.1 Mandatory Criteria 8

2.2 Optional Criteria 10

3 interTwin Release Management 12

4 interTwin First DTE Release 16

5 Conclusions 18

6 References 19

Table of Figures

Figure 1 - Procedure for the SW release within interTwin. 13

Figure 2 - SQAaaS service to perform an assessment on the source code. 13

Figure 3 - SQAaaS assessment report for Deltares hydromt component. 14

Figure 4 -Jira ticket for the SW component openEO Python Process Implementation. The assessment reports (in pdf and json formats) are included in the ticket 14

Figure 5 - Mandatory QCA included in the tickets, that the SW and DTE developers should check to prepare their components for release. 15

Figure 6 - interTwin DTE architecture, showing on the right with blue boxes, the DTE categories. 16

Figure 7 - Web page describing the itwinai component 17

Executive summary

This deliverable describes the first interTwin DTE release. This is the work performed by Task 3.3 during the first project period. Namely, the documentation about the Software release procedure in a Confluence Wiki page. This procedure includes mandatory Quality Criteria that the project's DTE components have to satisfy to be part of the release as well as optional criteria towards earning a badge.

It describes in detail how each DTE component is tracked towards the release, through Jira tickets as well as how the Quality Criteria is assessed through the *Software Quality Assurance as a Service* with underlying CI/CD Jenkins pipelines.

The first interTwin DTE release is detailed with descriptions of the platform architecture and statistics about the number of components and their QC assessment.

1 Introduction

This Deliverable describes the work done in the first period regarding the setting of the Software (SW) release management procedure. This procedure has been described and implemented according to discussions with the DTE developer teams responsible for the interTwin platform and applications.

Quality is a very important trait in Software development and release; thus, it has been made part of this procedure. Each DTE component is required to pass the Software Quality Criteria set by Task 3.3 to be part of an interTwin release. The mandatory and optional Quality Criteria is detailed in this document.

The assessment is performed on the source code through a Version Control System repository with the "*Software Quality Assurance as a Service*" (SQAaaS) from the EOSC-Synergy project[1], executing CI/CD pipelines using a Jenkins service.

The procedure and guidelines are described in Confluence Wikis, and the DTE components for the interTwin release are tracked in Jira tickets under the control of Task 3.3.

The first interTwin DTE release is also detailed in this deliverable, it includes the platform architecture and the description and status of the components in the release.

This document is organized as follows:

- **Section 2** describes the software and services quality assurance, the mandatory and optional criteria.
- **Section 3** details the software and DTE release management procedure.
- **Sections 4** describes the 1st interTwin DTE release.
- **Section 5** draws some conclusions.

[1] https://sqaaas.eosc-synergy.eu/

2 Software and Services Quality Assurance

In the framework of Task 3.3, we have defined a set of Quality Criteria Attributes (QCA), that software components have to pass to be included in the interTwin release. This quality criteria set was discussed and agreed upon with the DTE developer teams. The document: "A set of Common Software Quality Assurance Baseline Criteria for Research Projects" [R1] has served as the basis for the QCAs in interTwin.

The set of QCAs is documented in a confluence page:

- **https://confluence.egi.eu/display/interTwin/Software+and+Services+Quality+Assurance+%28SQA%29+guidelines** (internal)

The QCAs are abstract in the sense that they are agnostic with respect to the technologies and services used to assess them.

There are two main sections regarding the SW release in InterTwin:

- Mandatory criteria for the SW release.
- Optional criteria: for DTE developer teams that want to obtain a badge according to the EOSC-Synergy standards.

2.1 Mandatory Criteria

The following criteria are mandatory for any software component release. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [R2].

1. Code Accessibility [QC.Acc]

    - [QC.Acc01] Following the open-source model, the source code being produced MUST be open and publicly available to promote the adoption and augment the visibility of the software developments.
    - [QC.Acc02] Source code MUST use a Version Control System (VCS).

2. Code Workflow [QC.Wor]

    - [QC.Wor01] The main branch in the source code repository MUST maintain a working state version of the software component.

3. Code Management [QC.Man]

    - [QC.Man01] An issue tracking system MUST be in place; it facilitates structured software development. Leveraging issues to track down both new enhancements and defects (bugs, documentation typos). This criteria is adapted from the original baseline.

4. Semantic Versioning [QC.Ver]

- [QC.Ver01] Semantic Versioning[2] specification SHOULD be used for tagging the production releases. (This criteria is adapted from the original baseline.) This standard proposes a `<MAJOR>.<MINOR>.<PATCH>` schema for tagging releases, where:
    - A MAJOR increment denotes incompatible changes breaking backwards compatibility (obsoletes components).
    - A MINOR increment denotes backwards-compatible features/functionalities.
    - A PATCH increment denotes backwards-compatible bug fixes.

5. Licensing [QC.Lic]

- [QC.Lic01] As open-source software, source code MUST adhere to an open-source license to be freely used, modified and distributed by others. Non-licensed software is exclusive copyright by default. License MUST be compliant with the Open Source Definition[3].
- [QC.Lic01.1] Licenses MUST be physically present (e.g. as a LICENSE file) in the root of all the source code repositories related to the software component.

6. Documentation [QC.Doc]

- [QC.Doc02] Documentation is RECOMMENDED to use plain text format using a markup language, such as Markdown or reStructuredText. (This criteria is adapted from the original baseline.)
- [QC.Doc03] Documentation MUST be online and available in a documentation repository.
- [QC.Doc04] Documentation MUST be updated on new software versions involving any substantial or minimal change in the behaviour of the application.
- [QC.Doc05] Documentation MUST be updated whenever reported as inaccurate or unclear.
- [QC.Doc06] Documentation MUST be produced according to the target audience, varying according to the software component specification. The identified types of documentation and their RECOMMENDED content are:
    - [QC.Doc06.1] README file MUST be present.
    - [QC.Doc06.5] Developer, when applicable.

[2] https://semver.org

[3] https://opensource.org/osd

  - [QC.Doc06.6] Deployment and Administration, when applicable.
  - [QC.Doc06.7] User, when applicable.

2.2 Optional Criteria

The optional QCA are for the = DTE developer teams that aim at a badge awarded by EOSC-Synergy through the SQAaaS: [**https://sqaaas.eosc-synergy.eu/**].

In addition to the mandatory set of QCAs, the following QCAs should be met in order to obtain a badge. There are three badges: **Gold**, **Silver,** and **Bronze**.

For the Bronze badge mandatory QCAs must be meet, plus:

Documentation [QC.Doc]:

- [QC.Doc01] Documentation MUST be treated as code,
- [QC.Doc01.1] Version controlled, it MAY reside in the same repository where the source code lies.
- [QC.Doc06.2] CONTRIBUTING file MUST be present in order to communicate how external parties can contribute to the code.
- [QC.Doc06.3] A code of conduct (usually defined in a CODE_OF_CONDUCT file) MUST be present to establish the positive social attitudes expected within the community of code contributors.

For the Silver badge, the Bronze badge QCAs must be met, plus:

Metadata [QC.Met]

Metadata for the software component provides a way to achieve its full identification, thus making software citation viable [R3]. It allows the assignment of a Digital Object Identifier (DOI) and is key towards preservation, discovery, reuse, and attribution of the software component.

- [QC.Met01] A metadata file SHOULD exist alongside the code, under its VCS. The metadata file SHOULD be updated when needed, as is the case of a new version.

For the Gold badge mandatory the Silver badge QCAs must be met, plus:

Style [QC.Sty]

Code style requirements pursue the correct maintenance of the source code by the common agreement of a series of style conventions. These vary based on the programming language being used.

- [QC.Sty01] Each individual software product MUST comply with community-driven or de-facto code style standards for the programming languages being used.

Unit [QC.Uni]

Unit testing evaluates all the possible flows in the internal design of the code, so that its behavior becomes apparent. It is a key type of testing for early detection of failures in the development cycle.

- [QC.Uni01] Minimum acceptable code coverage threshold SHOULD be 70%.

Security [QC.Sec]

The security assessment is essential for any production software. An effective implementation of the security requirements applies to every stage in the Software Development Life Cycle (SDLC), especially effective at the source code level.

- [QC.Sec02] Source code MUST use automated linter tools to perform static application security testing (SAST)[4], that flag common suspicious constructs that may cause a bug or lead to a security risk (e.g. inconsistent data structure sizes or unused resources).

[4] https://owasp.org/www-community/Source_Code_Analysis_Tools

3 interTwin Release Management

The software release procedure is described in the following confluence page:

- **https://confluence.egi.eu/display/interTwin/PROC10+Software+Release** (internal)

The procedure describes the goals, entities involved and necessary steps towards a software release, as shown in **Figure 1**. As it can be seen, the 1st step is the Quality Assurance assessment of all software components that are candidates to be part of the interTwin release. The assessment is performed using the EOSC-Synergy's SQAaaS service: **https://sqaaas.eosc-synergy.eu/** as shown in **Figure 2**.

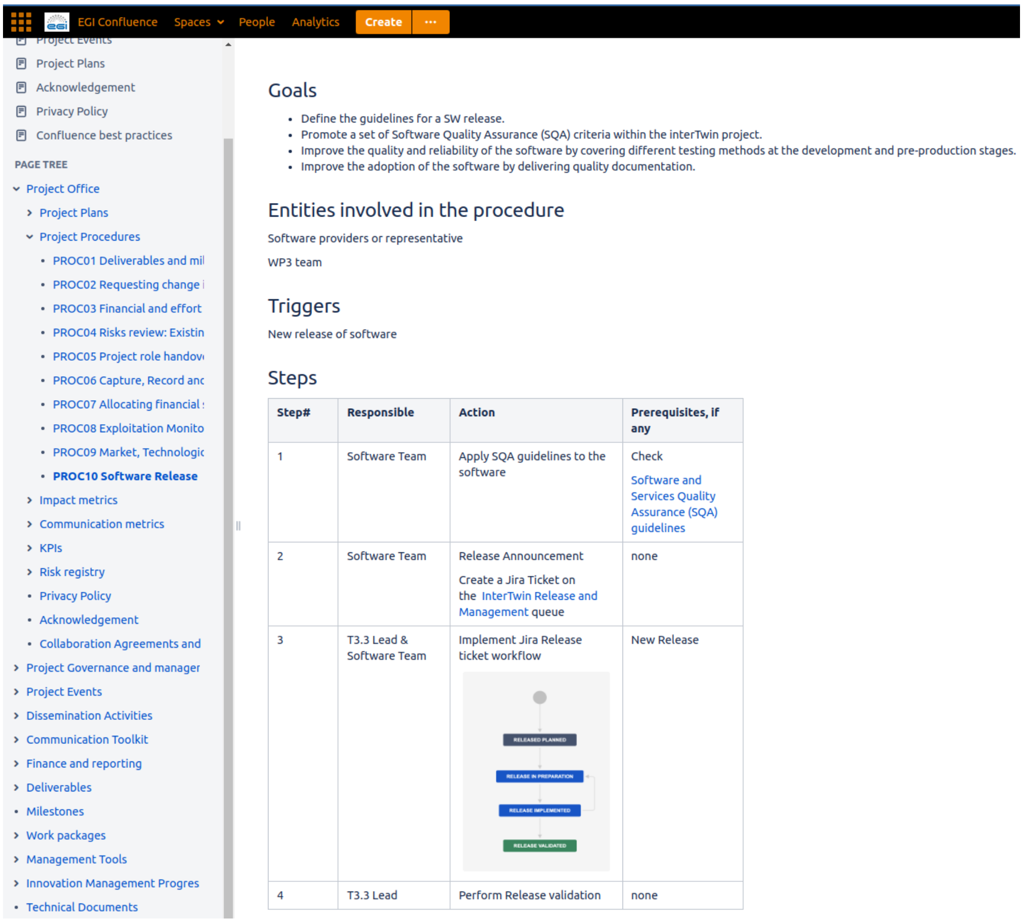
This service generates a report in pdf (shown in **Figure 3**) and json format.

For each SW component declared in our internal confluence page:

- **https://confluence.egi.eu/display/interTwin/SW+information%3A++Platform+services%2C+DTE+core%2C+DTE+thematic+modules%2C+DT+applications** (internal)

a Jira ticket is created to track its evolution towards the inclusion in the release; this is shown for one SW component in **Figure 4**. The full list of Jira tickets created for the interTwin release is at: **https://jira.egi.eu/projects/ITRM/issues/** (internal). The above-mentioned SQAaaS assessment reports are included in each of the corresponding Jira tickets.

The report in json format is parsed to extract the results of the assessment and included in the ticket as well, c.f. **Figure 5**. This is how we track the evolution of any given component towards the interTwin release.



- Project Events
- Project Plans
- Acknowledgement
- Privacy Policy
- Confluence best practices

PAGE TREE

- Project Office
  - Project Plans
  - Project Procedures
    - PROC01 Deliverables and mil
    - PROC02 Requesting change i
    - PROC03 Financial and effort
    - PROC04 Risks review: Existin
    - PROC05 Project role handov
    - PROC06 Capture, Record anc
    - PROC07 Allocating financial :
    - PROC08 Exploitation Monito
    - PROC09 Market, Technologic
    - **PROC10 Software Release**
  - Impact metrics
  - Communication metrics
  - KPIs
  - Risk registry
  - Privacy Policy
  - Acknowledgement
  - Collaboration Agreements and
- Project Governance and manager
- Project Events
- Dissemination Activities
- Communication Toolkit
- Finance and reporting
- Deliverables
- Milestones
- Work packages
- Management Tools
- Innovation Management Progres
- Technical Documents

Goals

- Define the guidelines for a SW release.
- Promote a set of Software Quality Assurance (SQA) criteria within the interTwin project.
- Improve the quality and reliability of the software by covering different testing methods at the development and pre-production stages.
- Improve the adoption of the software by delivering quality documentation.

Entities involved in the procedure

Software providers or representative

WP3 team

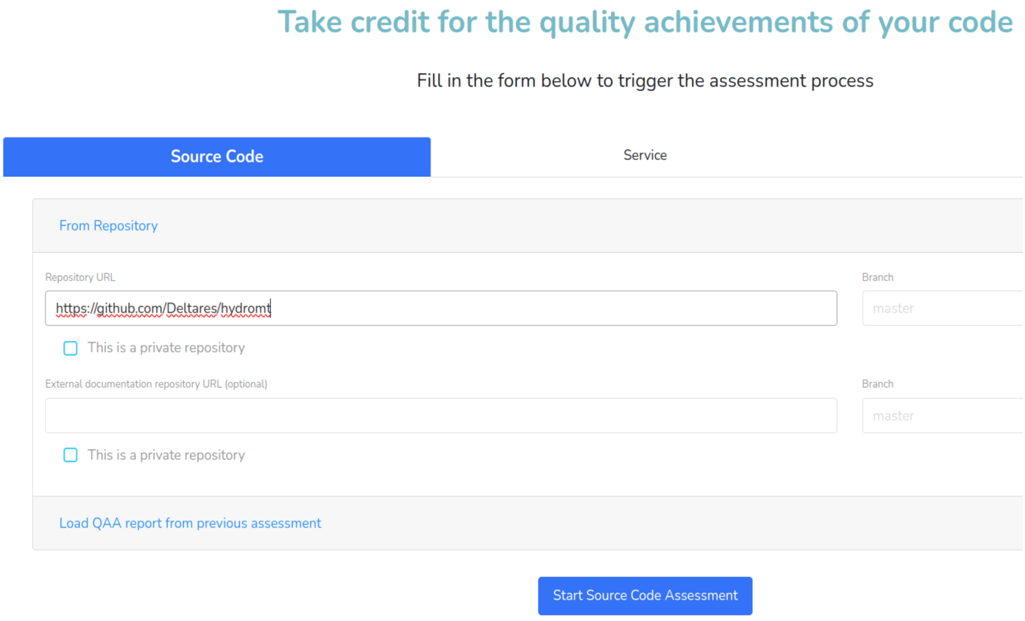
Triggers

New release of software

Steps

Step#	Responsible	Action	Prerequisites, if any
1	Software Team	Apply SQA guidelines to the software	Check Software and Services Quality Assurance (SQA) guidelines
2	Software Team	Release Announcement Create a Jira Ticket on the InterTwin Release and Management queue	none
3	T3.3 Lead & Software Team	Implement Jira Release ticket workflow	New Release
4	T3.3 Lead	Perform Release validation	none

Figure 1 - Procedure for the SW release within interTwin.



Take credit for the quality achievements of your code

Fill in the form below to trigger the assessment process

Source Code
Service
From Repository

Repository URL: https://github.com/Deltares/hydromt

Branch: master

□ This is a private repository

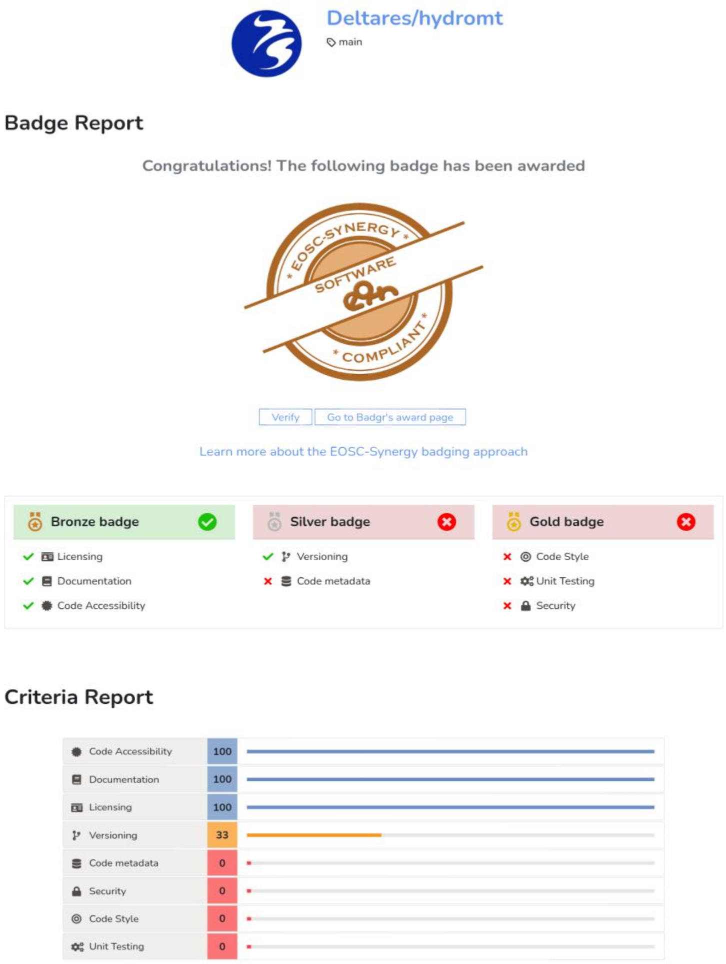
External documentation repository URL (optional):

Branch: master

□ This is a private repository

Load QAA report from previous assessment
Start Source Code Assessment

Figure 2 - SQAaaS service to perform an assessment on the source code.



Deltares/hydromt

main

Badge Report

Congratulations! The following badge has been awarded

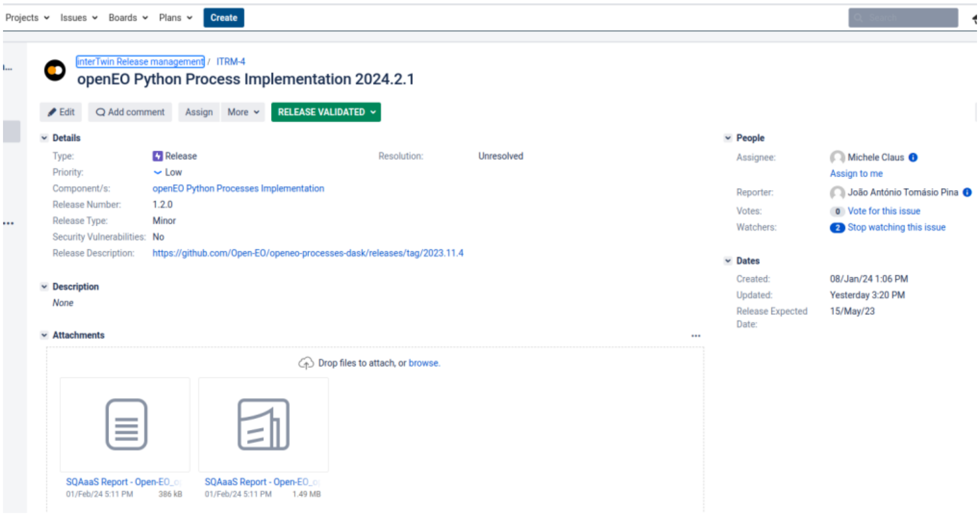
Learn more about the EOSC-Synergy badging approach

Bronze badge ✓	Silver badge ✗	Gold badge ✗
✓ Licensing	✓ Versioning	✗ Code Style
✓ Documentation	✗ Code metadata	✗ Unit Testing
✓ Code Accessibility		✗ Security

Criteria Report

Code Accessibility	100
Documentation	100
Licensing	100
Versioning	33
Code metadata	0
Security	0
Code Style	0
Unit Testing	0

Figure 3 - SQAaaS assessment report for Deltares hydromt component.



interTwin Release management / ITRM-4

openEO Python Process Implementation 2024.2.1

Details

Type:	Release	Resolution:	Unresolved
Priority:	Low		
Component/s:	openEO Python Processes Implementation		
Release Number:	1.2.0		
Release Type:	Minor		
Security Vulnerabilities:	No		
Release Description:	https://github.com/Open-EO/openeo-processes-dask/releases/tag/2023.11.4		

Description

None

Attachments

SQAaaS Report - Open-EO_o 01/Feb/24 5:11 PM 386 kB

SQAaaS Report - Open-EO_o 01/Feb/24 5:11 PM 1.49 MB

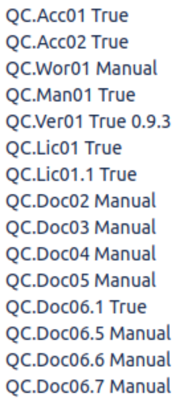
People

Assignee:	Michele Claus Assign to me
Reporter:	João António Tomásio Pina
Votes:	0 Vote for this issue
Watchers:	2 Stop watching this issue

Dates

Created:	08/Jan/24 1:06 PM
Updated:	Yesterday 3:20 PM
Release Expected Date:	15/May/23

Figure 4 -Jira ticket for the SW component openEO Python Process Implementation. The assessment reports (in pdf and json formats) are included in the ticket



```
QC.Acc01 True
QC.Acc02 True
QC.Wor01 Manual
QC.Man01 True
QC.Ver01 True 0.9.3
QC.Lic01 True
QC.Lic01.1 True
QC.Doc02 Manual
QC.Doc03 Manual
QC.Doc04 Manual
QC.Doc05 Manual
QC.Doc06.1 True
QC.Doc06.5 Manual
QC.Doc06.6 Manual
QC.Doc06.7 Manual
```

Figure 5 - Mandatory QCA included in the tickets, that the SW and DTE developers should check to prepare their components for release.

Some of the QC attributes can’t be automatically assessed, most are about the existence of certain documentation. Depending on the type of component there may be some types of documents that are not applicable to it. For example, in the case of a DTE application it does not make sense to have an administration guide.

Another case of manual assessment is QC.Wor01, since a working state version of the software component may not be in the main branch.

4 interTwin First DTE Release

There are four main categories of DTE components that fulfil the interTwin platform architecture shown in **Figure 6**, the web page:

https://www.intertwin.eu/intertwin-digital-twin-engine/ contains a detailed overview of the architecture and the 3 DTE categories:

1. The Digital Twin Engine infrastructure modules: provide specific capabilities on top of HW infrastructure, such as federated data and computing resources needed for modeling and simulation tasks:
    a. **https://www.intertwin.eu/dte-infrastructure-modules/**.
2. The Digital Twin Engine core modules: offer cross-domain capabilities, simplifying the creation and operation of data-intensive and compute-intensive DT applications:
    a. **https://www.intertwin.eu/core-dte-modules/**.
3. The Digital Twin Engine thematic modules (environment and physics): add-ons providing capabilities tailored to the needs of specific application groups:
    a. **https://www.intertwin.eu/thematic-modules-environment/**.
    b. **https://www.intertwin.eu/thematic-modules-physics/**.

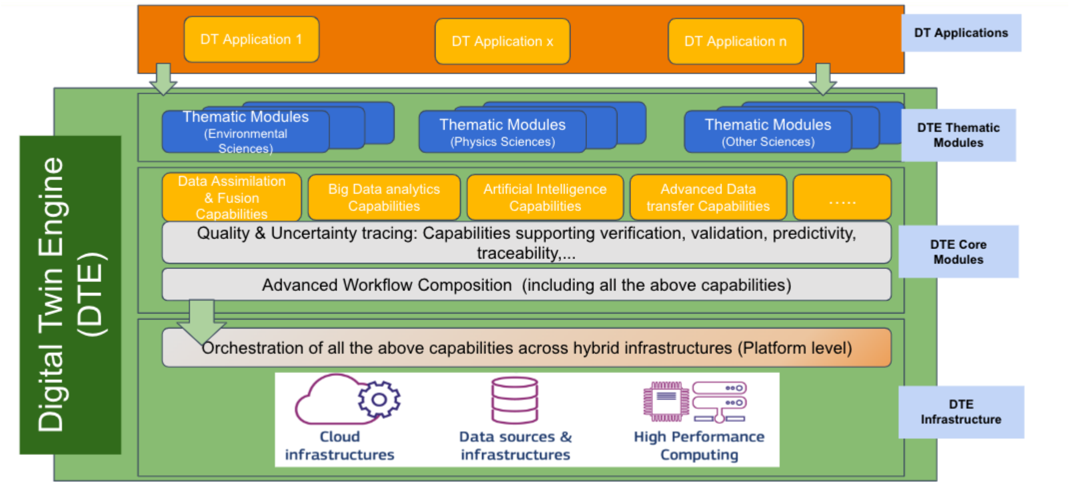
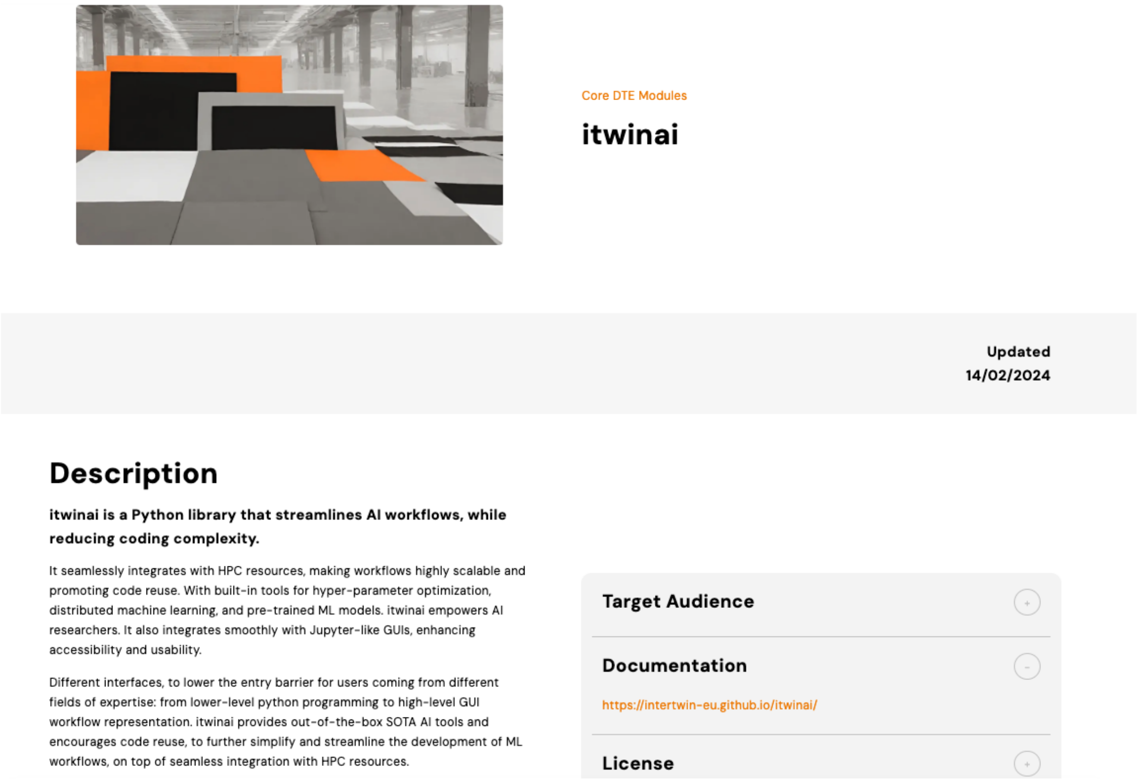

DT Application 1
DT Application x
DT Application n
DT Applications
Digital Twin Engine (DTE)
Thematic Modules (Environmental Sciences)
Thematic Modules (Physics Sciences)
Thematic Modules (Other Sciences)
DTE Thematic Modules
Data Assimilation & Fusion Capabilities
Big Data analytics Capabilities
Artificial Intelligence Capabilities
Advanced Data transfer Capabilities
.....
Quality & Uncertainty tracing: Capabilities supporting verification, validation, predictivity, traceability,...
Advanced Workflow Composition (including all the above capabilities)
DTE Core Modules
Orchestration of all the above capabilities across hybrid infrastructures (Platform level)
Cloud infrastructures
Data sources & infrastructures
High Performance Computing
DTE Infrastructure

Figure 6 - interTwin DTE architecture, showing on the right with blue boxes, the DTE categories.

The website exposes, for each component, a dedicated webpage (see example in **Figure 7**) where the information about the software is displayed (Description, Documentation link to source code, etc).



Core DTE Modules

itwinai

Updated
14/02/2024

Description

itwinai is a Python library that streamlines AI workflows, while reducing coding complexity.

It seamlessly integrates with HPC resources, making workflows highly scalable and promoting code reuse. With built-in tools for hyper-parameter optimization, distributed machine learning, and pre-trained ML models. itwinai empowers AI researchers. It also integrates smoothly with Jupyter-like GUIs, enhancing accessibility and usability.

Different interfaces, to lower the entry barrier for users coming from different fields of expertise: from lower-level python programming to high-level GUI workflow representation. itwinai provides out-of-the-box SOTA AI tools and encourages code reuse, to further simplify and streamline the development of ML workflows, on top of seamless integration with HPC resources.

Target Audience

Documentation

https://intertwin-eu.github.io/itwinai/

License

Figure 7 - Web page describing the itwinai component

The components that have been included in the first release have been the results of the first DTE internal release described in the deliverables D5.2[R4], D6.2[R5], D7.3[R6] and D7.4[R7] and which have passed the mandatory Quality Criteria defined in the procedure described in **Section 3**.

An overview of the status of the SW and DTE components is given next.

There were 44 Jira tickets created for as many components. From those, 6 components are still in the early stages of development and will not be released yet. The other 38 components will be part of the release, meaning that they have been through the SQAaaS assessment, and all pass the mandatory Quality Criteria. Furthermore, and regarding the optional Quality Criteria, 7 components have earned the EOSC-Synergy Bronze badge, 1 the Silver badge and 1 the Gold badge.

5 Conclusions

During the first period of the project, Task 3.3 has created a procedure for software release of the interTwin DTE. It has successfully used the SQAaaS from EOSC-Synergy to assess the mandatory attributes of the Quality Criteria needed for any given component to be part of the first release. Furthermore, several components have passed optional criteria, thus earning badges resulting from the assessment.

The tracking of the individual components through Jira tickets, has proven to be an appropriate way for developers to work their components towards inclusion in the release. Not all criteria are possible to be verified automatically with the CI/CD Jenkins pipelines, namely the type of documentation, accordingly the developer teams have performed this manual step.

Last but not least, we reported about the first interTwin DTE release, showing the platform architecture with a large fraction of the components as part of the release and some statistics about the software management process through Jira tickets.

The task T3.3 is planning to extend the software release procedure and activities for the second release of the DTE to include artifact release management. This activity has not been included in the first release cycle as many components have not yet reached the needed level of maturity.

6 References

References	
No	**Description / Link**
R1	Pablo Orviz, Alvaro Lopez, Doina Cristina Duma, Mario David, Jorge Gomes, Giacinto Donvito, **“A set of Common Software Quality Assurance Baseline Criteria for Research Projects”**, 2017, https://digital.csic.es/handle/10261/160086 (Github repository: https://github.com/indigo-dc/sqa-baseline).
R2	Key words for use in RFCs to Indicate Requirement Levels, Scott O. Bradner, **Internet Engineering Task Force** (1997-03): https://datatracker.ietf.org/doc/rfc2119/
R3	Smith AM, Katz DS, Niemeyer KE, **FORCE11 Software Citation Working Group. 2016. Software citation principles. PeerJ Computer Science** 2:e86 https://doi.org/10.7717/peerj-cs.86
R4	Spiga, D., Millar, P., Atherton, L., Antonacci, M., & Ciangottini, D. (2023). **interTwin D5.2 – First DTE Infrastructure Software Release (Final).** Zenodo. https://doi.org/10.5281/zenodo.10418216
R5	Campos, I., Moltó, G., Jacob, A., Orviz, P., Caballer, M., Bunino, M., Zoechbauer, A., & Fiore, S. (2023). interTwin D6.2 First release of the DTE core modules (1 Under EC review). Zenodo. https://doi.org/10.5281/zenodo.10224213
R6	Fiore, S., Fronza, M., Elia, D., Donno, D., Donno, E., Backeberg, B., Jacob, A., Claus, M., & Pagè, C. (2023). **interTwin D7.3 First version of the thematic modules for the environment domain (1 Under EC review).** Zenodo. https://doi.org/10.5281/zenodo.10224252
R7	Sinha Ray, G., Campos, I., Pidopryhora, Y., Vallero, S., Gennai, A., Vallecorsa, S., & Tsolaki, K. (2023). **interTwin D7.4 First version of the thematic modules for the physics domain (1 Under EC review).** Zenodo. https://doi.org/10.5281/zenodo.10224277