D5.2 SMS Status, Plan and SQA activities

31/07/2024

Abstract

The document serves three purposes. First, explain and analyse the current status of the EOSC Core Platform Service Management System (ECP SMS) being developed by the team that works on Lot 1 of the EOSC procurement contract 'Managed Services for the Development, Integration, Deployment and Operation of the Federated EOSC Core Platform,' while taking into account the lessons learned and experiences gained from the SMS developed in EOSC Future. A second purpose is to propose innovative features, methods and tools to facilitate the scalability, automation, and self-service in the EOSC SMS. These features will be analysed and prioritised by the product teams in the coming months after the submission of this deliverable and a subset of them might be implemented during the project. Thirdly, it aims to define a Release and Deployment Management process, as well as the methodologies of Continuous Integration (CI) and Software Quality Assurance (SQA) that will be promoted and enforced for all software components being developed by EOSC Beyond.

Funded by
the European Union

EOSC Beyond receives funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101131875.

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.

Document Description

D5.2 SMS Status, Plan and SQA activities			
Work Package number 5			
Due date	31/07/2024	**Actual delivery date:**	01/08/2024
Nature of document	Report	**Dissemination level**	Public
Document Status	Under EC Review	**Version**	1
Lead Partner	EGI Foundation		
Authors	Montserrat Gonzalez (MG), (EGI) Joao Pina (JP), (LIP) John Shepherdson (JWS), (CESSDA ERIC)		
Reviewers	Kostas Koumantaros (KK), (GRNET) Mario David (MD), (LIP) Pavel Weber (PW), (KIT) Mark Dietrich (MD), (EGI)		
Approved by	Technical Coordination Board (TCB)		
Public link	https://documents.egi.eu/document/4114		
DOI	https://doi.org/10.5281/zenodo.13152186		
Keywords	SMS, Service Management, SQA, Software Quality		

Revision History

Issue	Date	Description	Author/Reviewer
V 0.1	16/05/2024	Abstract, Sections 1 and 1.X, 2, 2.1-3 written by MG Sections 2.4, 2.5, 2.6 written by MG and JP Sections 3 and 3.X written by JP and JWS Section 4, 5, 6 written by MG, JP, and JWS	Montserrat Gonzalez (EGI), Joao Pina, (LIP), John Shepherdson (CESSDA ERIC)
V 0.2	16/07/2024	Review and incorporation of feedback from Kostas Koumantaros and Mario David.	Montserrat Gonzalez (EGI), Joao Pina, (LIP), John Shepherdson (CESSDA ERIC), Kostas Koumantaros (GRNET), Mark Dietrich (EGI)
V 0.3	26/07/2024	Review and incorporation of feedback from Pavel Weber and Mark Dietrich.	Montserrat Gonzalez (EGI), Joao Pina, (LIP), John Shepherdson (CESSDA ERIC), Pavel Weber (KIT)
V 0.4	30/07/2024	Approval by the board	TCB
V 1.0	**31/07/2024**	**Final**	

Table of content

Figures

Tables

Executive Summary

In this deliverable, the steps taken to elaborate the proposed enhancements of the EOSC Service Management System (SMS), as well as the Software Quality Assurance (SQA) activities carried out in order to implement SQA methodologies throughout the project, are described in detail.

The Service Management System (SMS) is an EOSC core service. Its current implementation is being developed by EOSC EU Node Procurement Lot 1. Through this deliverable, EOSC Beyond proposes innovation features for the SMS using the EU Node SMS deployed by Procurement Lot 1 as a starting point. EOSC Beyond innovation tools and methods targeting SMS might therefore have a direct impact on the EOSC EU Node.

The proposal of innovation features for the EOSC EU Node SMS follows these steps: (1) Understand and analyse the current status of the EOSC EU Node SMS developed by Procurement Lot 1; (2) Identify the needs of the EOSC EU Node, Node pilots, and future EOSC services developed by EOSC Beyond that can be addressed through enhanced features in the SMS; (3) Investigate and propose methods and tools to improve the EOSC EU Node SMS; (4) Analyse and deliver the initial assessment of the EOSC EU Node SMS innovation opportunities with the collaboration of EOSC Beyond's 'Task 7.1 GAP Analysis & Requirements Gathering'.

EOSC Beyond's WP7 will assess and investigate requirements related to these innovations and T5.4 will act as stakeholders. If approved by the project, the proposed enhancements will be included on the EOSC Beyond technical roadmap. In the second half of the EOSC Beyond project, there will be an update on the status of innovation features. The update will be explained in D6.1 SMS Status and SQA activities.

The SQA activities at this initial stage of the project were focused on establishing methods and best practices for software development. An agreement was made by the EOSC Beyond partners that the software components developed on the EOSC Beyond project will incorporate several best practices from the Agile methodologies such as (1) usage of a ticket system to track the technical roadmap, (2) adoption of a code versioning system, (3) implementation of continuous integration, delivery, and deployment practices. To support such practices, a Release and Deployment Management plan with strict Software Quality Assurance Criteria (SQA-C) attributes was elaborated. The assessment of SQA attributes will be made either by the project or by the SQAaaS service [9]. All those activities will be later reported in D6.1 SMS Status and SQA activities.

1. Introduction

To better understand the purpose of this deliverable and how it contributes to the project objectives, this section will summarise the EOSC Beyond objectives and expected results. Moreover, it will provide an overview of the deliverable's structure, its tasks and contributions, and the impact of the deliverable on the overall project.

1.1. The EOSC Beyond project

The European Open Science Cloud (EOSC) is developing as a federation of services, tools, data and other resources from hundreds of providers at local, national, regional and European with the aim of increasing discovery, access, and reuse of Open Science resources from European research communities in order to accelerate discovery times and improve science excellence in Europe.

The EOSC Beyond project aims to support the growth of EOSC in terms of integrated providers and active users by providing new EOSC core technical solutions. Therefore, developers of scientific application environments will be able to compose a diverse portfolio of EOSC resources and offer them to researchers.

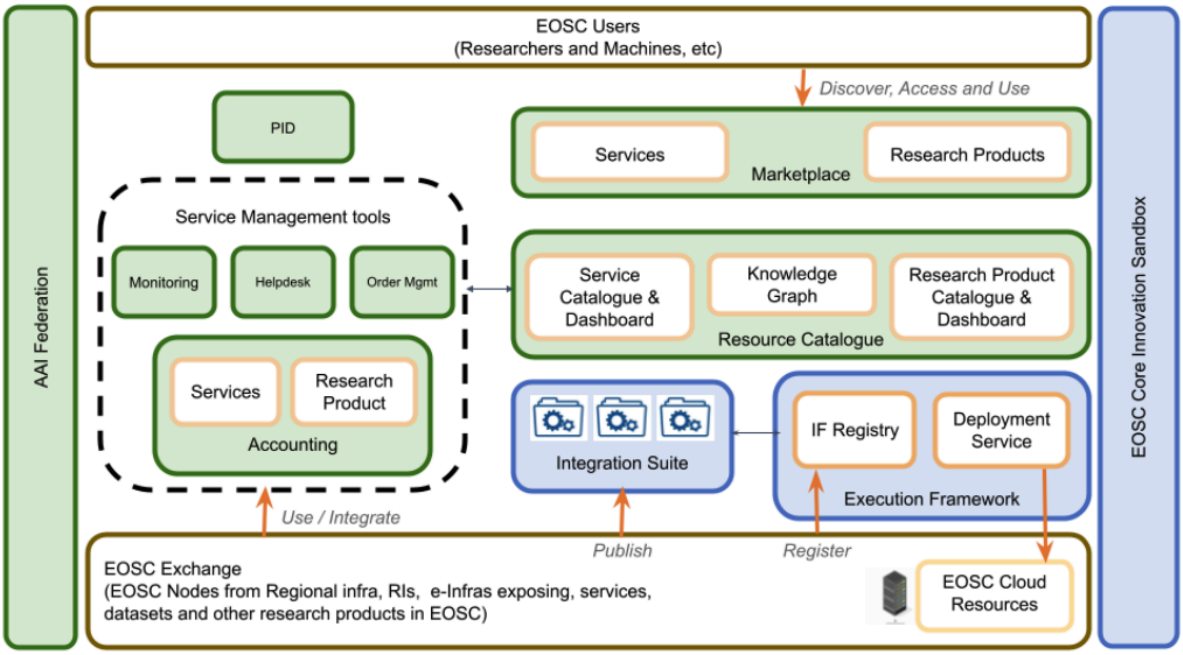


Figure 1 - The EOSC Core components overview shows new capabilities in blue and improved capabilities in green that will be developed by EOSC Beyond.

To achieve the project's ambition, existing EOSC core solutions are improved and new ones are added through these Key Exploitable Results (KER):

1. Improved existing EOSC Core services and framework.

2. EOSC Core Innovation Sandbox with automation tools for testing and validating new EOSC Core services, and allowing providers to validate integration with the EOSC Platform before deploying services.
3. EOSC Integration Suite as a portfolio of software adapters to the EOSC Core and horizontal services of the EOSC Exchange.
4. EOSC Execution Framework for composability and deployment of multi-supply resources, which will interoperate with relevant dataspaces.
5. EOSC-SIMPL interoperability technical guidelines.
6. Piloting EOSC Thematic and Regional/National Nodes.
7. Sustainability and exploitation plans for the EOSC Core and horizontal services.

Development will be driven by co-design, gathering requirements, and software validation in collaboration with a portfolio of diverse use cases representing needs of EOSC nodes and initiatives operating at national level (e-Infra CZ in Czechia, NFDI in Germany), regional level (NI4OS in South East Europe), and thematic European nodes (CESSDA, CNB-CSIC, ENES, LifeWatch, METROFood-RI, and Instruct-ERIC).

The results of EOSC Beyond will support Open Science for modern data-intensive and multidisciplinary science, enabling scientists, organisations, and countries to discover, access, and reuse an increasing number of resources. As part of the EOSC Platform, which is currently implementing the Minimum Viable EOSC (MVE) as an integrated operational environment delivering selected horizontal services and the EOSC Core, EOSC Beyond will add additional components and capabilities to the EOSC Core.

1.2. Scope and purpose of the deliverable

EOSC Beyond incorporates the latest evolution of EOSC and the concept of EOSC Nodes. The EC funded the development of the first node of this network, the EOSC EU Node, through EOSC procurement contracts. This will make the EU Node operational during the autumn of 2024. The EU Node will operate the current EOSC Core services at European level and make them available to other EOSC nodes and providers for integration. Thus, it will initially be the primary target of EOSC Beyond innovation activities. Due to the parallel timeline of the innovation activities within the EOSC Beyond project and the EOSC procurement work, the result of the innovations will not be included in the current development of the EU Node, but the results will be available for future development of the EU Node. Nevertheless, the EU Node may adopt minor improvements in its agreement with the EC and its implementing entities as long as they do not affect current procurement contracts.

Taking into account the points above, the ‘D5.2 SMS Status, Plan and SQA activities’ fall under the scope of the EOSC Core Platform SMS (ECP SMS) and its associated best practices. The document has three purposes.

- The first purpose is to explain the current status of the ECP SMS being developed by the team working in the EOSC Lot 1 procurement contract ‘Managed Services for the Development, Integration, Deployment and Operations of the Federated EOSC Core

Platform' and that takes into consideration the experiences and lessons learned from the SMS developed in the EOSC Future project.

- Following an examination of the current status, the second purpose is to propose innovative features to facilitate ECP SMS scalability, automation, and self-service by investigating, developing, and implementing methods and tools.
- A third purpose is to define the technical implementation of a Release and Deployment Management process, which is part of the ECP SMS, and the Continuous Integration (CI) and Software Quality Assurance (SQA) methodologies that will be promoted and enforced for all software components being developed in EOSC Beyond.

1.3. Structure of the deliverable

The document is structured in the following way:

- In the first part of the document, with main contributions from T5.4 Enhanced ECP SMS and Data Policies, there is a status report on SMS, an analysis of SMS and a proposal for innovation features that will be most beneficial to EOSC users and operators of the EOSC EU Node.
- The second part of the document, with contributions from T5.5. Release management and Software Quality Assurance includes a technical implementation of the Release and Deployment Management process, which is part of the ECP SMS, as well as the promotion and enforcement of Continuous Integration (CI) and Software Quality Assurance (SQA) methodologies for all software components being developed in EOSC Beyond.
- Finally, the document ends with conclusions and next steps that tie together the two previous sections that address work related to the SMS from a different perspective.

1.4. Contributions to the deliverable and outcomes

Different tasks contribute to D5.2, which is used by other tasks as input. This diagram summarises the contribution to the deliverable and how it contributes to other EOSC Beyond tasks.

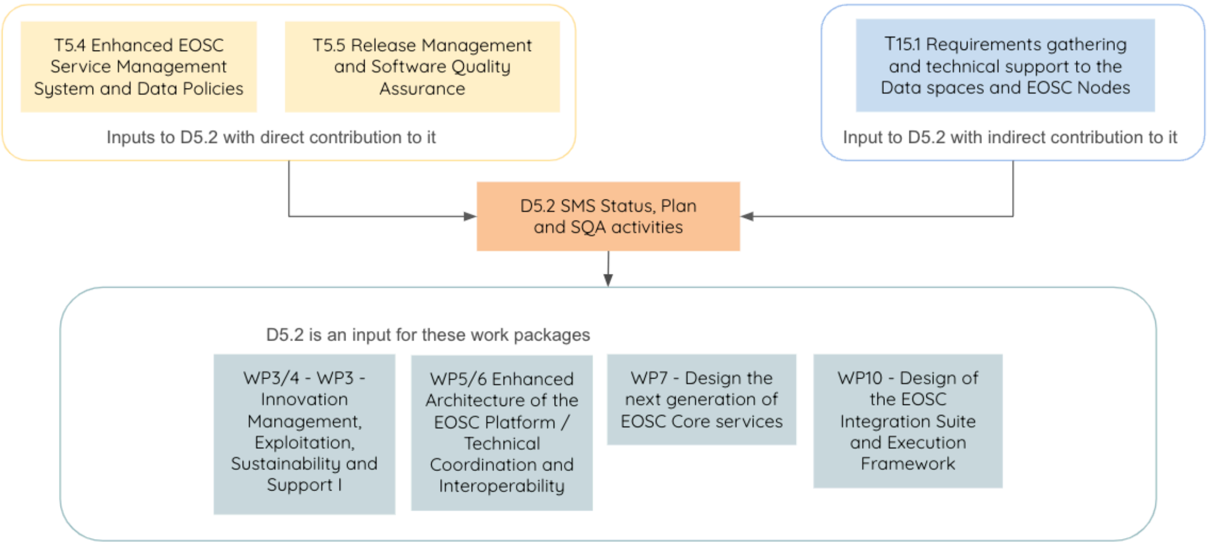

T5.4 Enhanced EOSC Service Management System and Data Policies
T5.5 Release Management and Software Quality Assurance
Inputs to D5.2 with direct contribution to it
T15.1 Requirements gathering and technical support to the Data spaces and EOSC Nodes
Input to D5.2 with indirect contribution to it
D5.2 SMS Status, Plan and SQA activities
D5.2 is an input for these work packages
WP3/4 - WP3 - Innovation Management, Exploitation, Sustainability and Support I
WP5/6 Enhanced Architecture of the EOSC Platform / Technical Coordination and Interoperability
WP7 - Design the next generation of EOSC Core services
WP10 - Design of the EOSC Integration Suite and Execution Framework

Figure 2 - EOSC Beyond tasks that provide input to D5.2 and WP that use D5.2 as an input

2. ECP SMS Status and Innovation Proposals

The current ECP SMS implementation is part of the EC-funded EOSC Lot 1 procurement contract 'Managed Services for the Development, Integration, Deployment and Operations of the Federated EOSC Core Platform', which will be operational by autumn 2024. The EOSC Procurement Lot 1 develops the EOSC EU Node Core services. The EOSC Procurement Lot 2 and 3 develop the EOSC EU Node Exchange services.

This section explains what the ECP SMS implementation will look like once in full production by autumn 2024 thanks to information shared by the EOSC Lot 1 procurement team.

The ECP SMS status will be analysed and methods and tools will be proposed for facilitating scalability, automation, and self-service that EOSC Beyond could develop and improve the overall SMS maturity consolidation. As mentioned in section 1.2, it is important to note that due to the parallel timeline of the innovation activities within the EOSC Beyond project and the EOSC procurement work, the innovations will not be included in the EU Node's current development, but will be available for future developments.

2.1. Objective and scope of the ECP SMS

The objective of the ECP SMS is to structure, plan, implement, monitor, and continually improve all service management processes that allow services and operational roles in delivering the minimum level of the ECP and supporting interaction with all Exchange services (services provided either by EU Node procurement contractors or third party providers that will be onboarded on the EOSC EU Node Exchange). Therefore, the SMS scope includes the ECP services and EOSC Exchange services. The existing EOSC Core services which are managed by SMS and which will be improved in EOSC Beyond are:

EOSC Beyond will improve the Platform architecture and the following existing EOSC Core services and framework:

- Authentication and Authorization Infrastructure Architecture (AAI)
- Persistent Identifiers
- Resource Catalogue
    - Including EOSC Resource Profiles and Models
- Metadata Knowledge Graph
- Marketplace
- Accounting
- Monitoring
- Messaging
- Helpdesk
- SMS, whose improvement plan is included in the current deliverable, including
    - Release Management and Software Quality Assurance
- Policy recommendations
- Release Management and Software Quality Assurance, whose improvement plan is included in the current deliverable.

In addition, the EOSC Beyond project will develop three new services:

- EOSC Core Innovation Sandbox
- EOSC Integration Suite
- EOSC Execution Framework

2.2. Tools for the ECP SMS

The ECP SMS is being developed based on the FitSM family of standards[1] , a lightweight ITSM framework that enables flexibility and adaptability. The tools used to implement the ECP SMS are:

- Confluence, to document FitSM processes, policies and procedures.
- Jira, to manage communication of tasks related to ECP SMS.
- Zammad, an open-source and self-hosted helpdesk platform.

2.3. FitSM framework applied to the ECP SMS

In order to implement a FitSM-based SMS, three levels are required: governance, control, and operational.

[1] FitSM-1 Process model (https://www.fitsm.eu/fitsm-parts/)

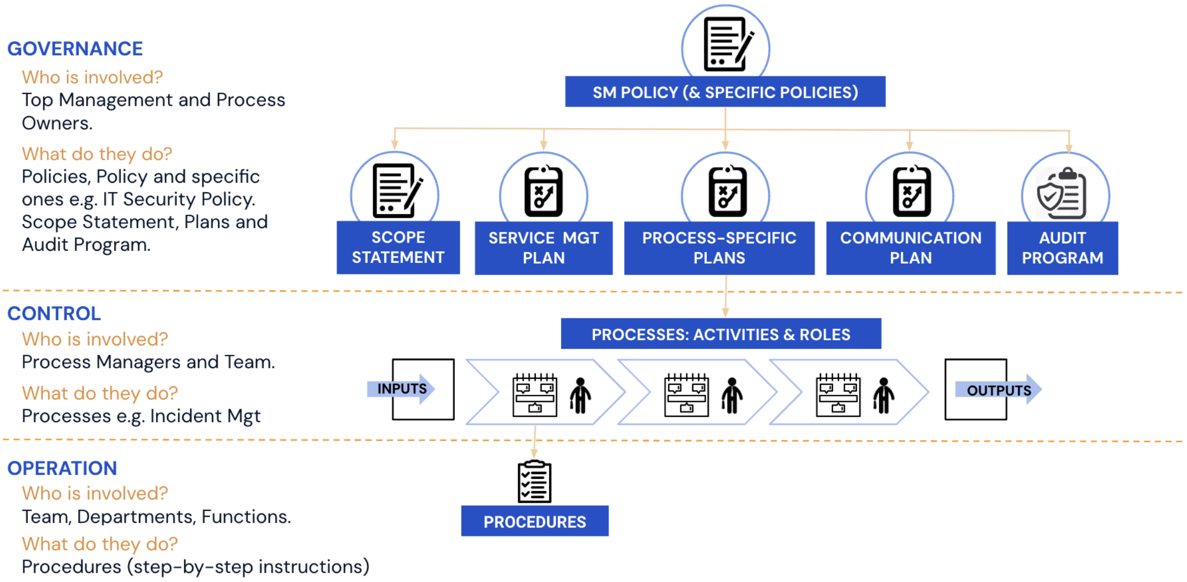

GOVERNANCE
Who is involved?
Top Management and Process Owners.
What do they do?
Policies, Policy and specific ones e.g. IT Security Policy. Scope Statement, Plans and Audit Program.
SM POLICY (& SPECIFIC POLICIES)
SCOPE STATEMENT
SERVICE MGT PLAN
PROCESS-SPECIFIC PLANS
COMMUNICATION PLAN
AUDIT PROGRAM
CONTROL
Who is involved?
Process Managers and Team.
What do they do?
Processes e.g. Incident Mgt
PROCESSES: ACTIVITIES & ROLES
INPUTS
OUTPUTS
OPERATION
Who is involved?
Team, Departments, Functions.
What do they do?
Procedures (step-by-step instructions)
PROCEDURES

Figure 3. Service Management overview and layers according to the FitSM family of standards (DOI 10.5281/zenodo.11403119)

- Governance level: involves top management and process owners and includes the Service Management (SM) policy and other relevant policies, such as IT security, the SMS plan on how to implement the overall ITSM system, process-specific plans, and the communication plan, as well as the assessment program, if necessary.
- Control level: includes the process managers and process teams, as well as all the process definitions that outline the expected inputs, outputs, roles, and responsibilities.
- Operational level: is made up of teams and staff who create and execute procedures that include step-by-step instructions.

The FitSM standard process model was adapted to meet ECP SMS requirements as follows:

- Divide SPM (Service Portfolio Management) into two parts: SPM for EOSC Core services and exchange resources developed by the EOSC EU Node procurement, and SPM for third-party EOSC Exchange services.
- Leave out the Supplier and Federation Member Relationship process for now, as it will be reevaluated for inclusion if needed.
- Merged CHM (Change Management) and RDM (Release and Deployment Management) to create ChaRDM.

Hence the tailored ECP SMS control level contains the following 13 processes:

- Process 1: Service Portfolio Management for EU Node procurement Core and Exchange services (SPM-Lots)
- Process 2: SPM for third-party EOSC Exchange services (SPM-Onboarding)
- Process 3: Service Level Management (SLM)
- Process 4: Service Reporting Management (SRM)
- Process 5: Service Availability and Continuity Management (SACM)
- Process 6: Capacity Management (CAPM)

- Process 7: Information Security Management (ISM)
- Process 8: Customer relationship management (CRM)
- Process 9: Incident and Service Request Management (ISRM)
- Process 10: Problem Management (PM)
- Process 11: Configuration Management (CONFM)
- Process 12: Change management and Release and deployment management (ChaRDM)
- Process 13: Continual Service Improvement (CSI)

2.4. Status of the ECP SMS and opportunities for innovation

The objectives and activities described below describe what the EOSC EU Node SMS procurement team expects to accomplish when the SMS processes reach the execution stage in the autumn 2024. With an understanding of the expected status and how it will be implemented in the SMS processes, opportunities for innovation will be identified.

Innovation activities are proposed on the basis that they will improve the scalability, automation, and self-service, facilitate an easier alignment between SMS processes, improve the overall SMS maturity consolidation, simplify the data available in SMS processes, and connect the data available in SMS processes to the services part of the EOSC Core portfolio.

SPM for EU Node for procurement Core and Exchange services (SPM-Lots)
Objective
Establish and maintain a definitive list of EU Node procurement Core and Exchange services, and service components that indicate which services are at 'production quality'.
Activities involved in this process
• Evaluation of new or changed service proposals. • Management of the evolution of the services throughout their lifecycle. • Identification and communication with service representatives. • Planning changes to the services.
Expected process result
• A complete up-to-date Service Portfolio for EU Node procurement Core and Exchange services is documented.
Opportunities for innovation
• Feature 1: Automated synchronisation of the Service Portfolio for the EOSC Core services and the Service Portfolio for the EOSC Exchange services developed by the EOSC EU Node procurement The ECP Service Portfolio is manually created in Confluence for EOSC Core and EOSC Exchange services developed by the EU Node. The SPM-Lots in Lot 1 manually create a copy of the Exchange service portfolio (of the Exchange services developed by Lots 2 and 3). User story: **As** EOSC EU Node SMS staff **I want to** have automatically synchronised the ECP SMS Service Portfolio and the

Exchange Service Portfolio so that the teams working in the different Portfolios (which have interrelated services and dependencies)
so that the teams working in the different Portfolios (which have interrelated services and dependencies) do not have to manually copy and paste information. By creating a copy of information, it may also become outdated and inaccurate.

Table 1 - SPM-Lots process objectives, activities, and expected outputs

SPM for third-party EOSC Exchange services (SPM-Onboarding)

Objective

Develop and manage related activities to on/off-board and audit providers and resources to EOSC-Exchange compliant with the EOSC Rules of Participation (RoP) adopted by the EOSC Co-programmed Partnership and Tripartite Governance.

Activities involved in this process

- Define the structure of the EOSC Onboarding Team (EOT) and its coordination.
- Define the procedures for on/off-board and audit providers and resources to EOSC-Exchange.
- Management of the evolution of the services throughout their lifecycle.
- Identification and communication with service representatives.
- Planning changes to the services.

Expected process results

- Procedures to on/off-board and audit providers and resources to EOSC-Exchange that will be manually checked.
- Providers of the EOSC EU Node will be offered a web hosted portfolio of third-party EOSC Exchange services.

Opportunities for innovation

- Feature 2: Automated onboarding and auditing procedures for third-party EOSC Exchange providers and services
  The onboarding and auditing procedures are now a mix of automated and manual review of compliance with the RoP.

  User story:
  As EOSC EU Node SMS staff
  I want to see the outcome of predefined checks of the onboarding requirements, defined in an onboarding procedure, when a new provider or service requests their addition to the EOSC Exchange
  so that the onboarding procedure is more efficient.

  As Service Provider
  I want to have access to the pre-defined requirements for the service onboard
  so that the onboarding of my resources to EOSC is easier and less time consuming

- Feature 3 Automated creation of a third-party EOSC Exchange service portfolio
  When the onboarding to EOSC Exchange is completed, the service portfolio shall be populated with third-party EOSC Exchange services.

  User story:
  As EOSC EU Node SMS staff/EOSC third-party provider
  I want to see the EOSC Exchange service (third-party) portfolio populated after the

onboarding is completed **so that** providers can immediately see what services they have onboarded (in previous versions, there was a delay in publication). Note: The EOSC EU Node has not yet defined the onboarding process for EOSC Exchange.

Table 2. SPM-Onboarding process objectives, activities, and expected outputs

Service Level Management (SLM)
Objective
To maintain service catalogues, and to define and evaluate agreements on service quality with customers and suppliers.
Activities involved in this process
• Measure service performance against service targets defined in the procurement contract, notifying and recording any deviations.
Expected process results
• End users of the EOSC EU Node will be offered web hosted catalogues of EOSC Core resources, EOSC Exchange services developed by the EU Node, third-party EOSC Exchange services. • Service performance shall be evaluated against service targets defined in SLAs.
Opportunities for innovation
• Feature 4: Automated synchronisation of the third-party EOSC Exchange services portfolio and catalogue Synchronise the third-party EOSC Exchange services portfolio (Developed by the Lot 1) with the Service catalogue (Service Catalogue & Dashboard service + Research Product Catalogue & Dashboard service by EOSC Beyond). User story: **As** EOSC EU Node SMS staff/EOSC third party provider **I want to** see the third-party EOSC Exchange service portfolio automatically synchronised with the EOSC Exchange service catalogue after the onboarding is completed **so that** end users can request access to the onboarded services. • Feature 5: Automated synchronisation of the EU Node EOSC Exchange services portfolio and catalogue Synchronise the EOSC Exchange services portfolio developed by the EU Node (documented in Confluence by Lot 2 and 3) and service catalogue (Service Catalogue & Dashboard service + Research Product Catalogue & Dashboard service by EOSC Beyond). User story: **As** EOSC EU Node SMS staff/EOSC service provider **I want to** see the EOSC Exchange service portfolio created by the EU Node automatically synchronised with the EOSC Exchange service catalogue (in previous versions, there was a delay in synchronisation that create miscommunication when providers could see their services onboarded but the end users could not see them) **so that** end users can request access to the services (i.e. 1. Cloud Container

Platform, Compute (Virtual Machines) Infrastructure, Bulk Data Transfer, File Synchronisation and Sharing, Interactive Notebooks, and Large File Transfer).

- Feature 6: Visibility of Service Level Agreements and Operational Level Agreements
  Due to the current system, Operating Level Agreements and Service Level Agreements cannot automatically be viewed leading to a high management complexity.

  User Story:
  As Service provider/Node representative
  I want to see all active OLA and SLA for a given service/node
  so that I can keep track of operation requirements and expected service level.

- Feature 7: Automate the service performance evaluation against SLAs
  Monitoring services generate monthly reports. SLA compliance is then manually assessed. An automatic assessment of SLA levels for service monitored parameters could be defined by the service owner in the Monitoring services.

  User story:
  As EOSC EU Node SMS staff/EOSC Service provider
  I want to automatically assess SLA compliance based on monthly monitoring reports
  so that the SLA compliance assessment is simplified and efficient

- Feature 8: Store or allow download of the service performance evaluation against SLAs
  Monitoring services could allow downloading or storing the service performance evaluation against SLAs.

  User story:
  As EOSC EU Node SMS staff
  I want to be able to download or have access to an historic repository of SLA compliance reports based on monthly monitoring reports
  so that SLA compliance assessment history can be accessed and shared.

- Feature 9: Automated messaging when SLAs are at risk
  Monitoring services could alert the service owner when SLA parameters are nearing non-conformity or when there is an SLA breach.

  User story:
  As EOSC EU Node SMS staff/Service provider
  I want to be notified when SLAs are at risk of non-conformity
  so that problems can be explored and solved, or mitigation strategies can be developed to provide the best service possible.

Table 3 - SLM process objectives, activities, and expected outputs

Service Reporting Management (SRM)
Objective
For the EOSC EU Node Core Platform, to specify all service reports and process reports, and ensure they are produced according to specifications in a timely manner to support decision-making.
Activities involved in this process
• Define all the reports to be produced on a regular basis. • Define templates for service reports to standardise the report structure and support effective and repeatable reporting. • Provide guidelines to create service reports. For example, on how to give the report a unique name (ID), describing the purpose of the report, identifying its audience, defining its frequency, outlining the intended contents of the report, and defining its format and method of delivery.
Expected process results
• List and specification of all the agreed reports. • Actual reports produced on a regular basis.
Opportunities for innovation
• Feature 10: Automated report generation Using the report templates created based on the SLAs, automate report generation as the SACM process provides data for service availability (via the Lot 1 Monitoring and Accounting Frameworks service) and the CAPM process (via information shared by the provider and the Service Capacity Plan) provides data for performance and utilisation. User story: **As** EOSC EU Node SMS staff/Service provider **I want to** reports based on SLAs be automatically generated using the SACM and the CAPM data **so that** the reporting process is more efficient.

Table 4 - SRM process objectives, activities, and expected outputs

Service Availability and Continuity Management (SACM)
Objective
At the EOSC EU Node Core Platform, ensure sufficient service availability to meet agreed requirements and adequate service continuity during exceptional circumstances.
Activities involved in this process
• Identify the most critical service availability and continuity requirements based on SLAs and other sources of information. • Identify and assess risks related to service availability and continuity. • Monitor service availability and continuity, and record the results on an ongoing basis. • Maintain service availability and continuity plans.
Expected process results
• Service availability and continuity plans. • Monitoring of service availability and continuity plans. • Service availability and continuity reports.
Opportunities for innovation
• Feature 11: Automating the service availability and continuity reports could be combined with Feature 11. User story: **As** EOSC EU Node SMS staff/Service provider **I want to** reports based on availability and continuity be automatically generated using the SACM data **so that** the reporting process is more efficient.

Table 5 - SACM process objectives, activities, and expected outputs

Capacity Management (CAPM)
Objective
At the EOSC EU Node Core Platform, ensure sufficient capacities are provided to meet agreed service levels and performance requirements
Activities involved in this process
• Identify service performance and capacity requirements based on SLAs and predicted demand. • Identify the current level of capacities plus information on the past, current and future (predicted) utilisation of resources. • Capacity planning considers human, technical and financial resources. • Monitor capacity, resource utilisation, and service performance.
Expected process results
• Capacity plans. • Monitoring of capacity plans.
Opportunities for innovation
• Feature 12: Automated reporting on capacity Currently, Capacity Management is a manual process in which each service provider delivers a report at regular intervals. The information gathering for the report requires information from several individual services: Order Management, Service Availability and Continuity Management, Service Level Management and Configuration Management. Capacity Management reporting per service could be automated by collecting information from the SLA, Marketplace and Order Management (service part of EOSC Beyond T9.4). This could be achieved with the implementation of Features 10 and 21. User story: **As** Service provider **I want to have** regular reports of the usage of my services and analyse the current delivered capacity against my total capacity **so that** I can plan my resources well ahead.

Table 6 - CAPM process objectives, activities, and expected outputs

Information Security Management (ISM)
Objective
At the EOSC EU Node, to manage information security effectively through all activities performed to deliver and manage services, so that the confidentiality, integrity and accessibility of relevant information assets are preserved.
Activities involved in this process
• Ensuring all EOSC Node partners follow the agreed security policies and requirements. For example, taking part in the incident response. • Produce a complete and up-to-date information asset inventory and assess information security risks. • Plan and implement security control to risks and threats. • Produce any guidelines and documentation for security management.
Expected process results
• EOSC EU Node IT Security plan. • Up-to-date inventory of assets. • Approved information security policies. • Information security risks and threat assessments. • Documented information security risks controls. • Documented security and requirements to be considered as part of Quality Assessment and deployment.
Opportunities for innovation
Currently, ISM-related activities are mainly implemented by following the ITSRM2 methodology as requested by the EC. It is important to note at this stage that CMDB input will be required, as will security incident reports raised by users or service providers. • Feature 13: Automated notification of security incidents User Story: **As** Security officer user of a given service/Node **I want to** be notified of all security incidents immediately if the operational state of the affected services changes. Also knowing what services depending on it are affected. **so that** security incidents can be viewed and responded to quickly

Table 7 - ISM process objectives, activities, and expected outputs

Customer relationship management (CRM)
Objective
At the EOSC EU Node, ensure all customer interactions are properly managed and conduct appropriate customer analysis.
Activities involved in this process
• Set up an initial customer database. • Select communication channels for customer engagement. • Define a way to perform and document the results of a service review. • Define a way to record, respond to and follow up a customer complaint • Define a way to evaluate customer satisfaction on a regular basis, e.g. (online) surveys. • Perform CRM reviews via customer satisfaction surveys.
Expected process results
• Customer database. • Communication channels for customer engagement. • Procedure for a service review. • Procedure for customer complaints. • Results of customer satisfaction surveys.
Opportunities for innovation
• Feature 14: Automated user satisfaction survey Through the Order Management System, for example after the user has accessed the service for a period, a survey can be created to ask how it is their experience with the service, what works well and what could be improved. User story: **As** EOSC EU Node SMS staff/Service provider **I want to** know what is the satisfaction level of the service users **so that** I understand the positive perceptions to exploit and the negative perceptions to improve.

Table 8 - CRM process objectives, activities, and expected outputs

Incident and Service Request Management (ISRM)
Objective
The ISRM ensures the stable operation of the infrastructure according to the service quality defined in the Service Level Management (SLM)
Activities involved in this process
• Maintain updated incident records from customers. • Defines the escalation procedure for the EOSC Ticketing System. • Classify and manage major incidents. • Provide at regular intervals reports about the major incidents.
Expected process results
• Maintain the Helpdesk service operation. • Automatic ticket escalation procedure after classification and definition.
Opportunities for innovation
● Feature 15: Automated Knowledge Database (KB) and Frequently Asked Questions Currently, the population of the Knowledge Database is manual and made by the first-level support for their internal guidance and not visible to external users. User Story: **As** Service provider **I want to** read the frequently asked questions by users of the service **so that** I can provide to the users of a service a list of Frequently Asked Questions regarding the EOSC ecosystem ● Feature 16: Automated reply to FAQ Currently, questions are answered manually, and having automated responses to FAQ (e.g. using AI) and when staff are off duty will be beneficial. User Story: **As** EOSC EU Node SMS staff **I want to** be able to provide to the end users of EOSC services and service providers automated replies to FAQ **so that** they can access to information even when staff is off duty and to decrease the time of the SMS staff replying FAQ

Table 9 - ISRM process objectives, activities, and expected outputs

Problem Management (PM)
Objective
Maintain updates of the Known Error Database (KEDB) and provide solutions to recurrent problems.
Activities involved in this process
• Maintain updates of the Known Error Database (KEDB) • Provide solutions to recurrent problems.
Expected process results
• Maintain the KEDB update. • A solution to recurrent problems.
Opportunities for innovation
The KEDB requires rich statistical information to be effective and as the Helpdesk service under the EOSC Node is new, future improvements are not foreseen due to lack of data. ● Feature 17: Automated Known Error Database (KEDB) **As** EOSC EU Node SMS staff **I want** to have tools at my disposal to automatically analyse information from all incident responses **so that** I can easily detect recurrent problems on the EOSC infrastructure.

Table 10 - PM process objectives, activities, and expected outputs

Configuration Management (CONFM)

Objective

Provide and maintain a logical model of all configuration items and their relationships and dependencies. This model and relationships are recorded in a configuration management database (CMDB).

Activities involved in this process

- Maintain the definitions of all CI and relationship types
- Plan regular verifications of the configuration information held in the CMDB
- Ensure that configuration verifications are conducted and identified nonconformities are addressed
- Take a configuration baseline when needed

Expected process results

The CONFM is a key process of supporting the EOSC federation. The information stored on the CMDB is critical to the good behaviour of all EOSC Nodes ecosystems and information stored should be of good quality. For this to be achieved:

- Store all service endpoints and operational parameters required for the EOSC EU Node service operation
- Store all service endpoints of all federated EOSC Nodes.
- Keep updated all names and contacts of the service operators.

Opportunities for innovation

At the EOSC EU Node, the CMDB is composed of two independent services - the GOCDB and Confluence. This combination makes it harder to maintain information in a coherent state, especially in dynamic environments like the EOSC Nodes. Improvements to the CMDB would greatly enhance the following services: EOSC EU Node configuration, EOSC Interoperability Framework, Change and Release Management, Incident and Problem Management and Service Level Management.

- Feature 18: Improve the EOSC Service Data Model

  The Service Data Model is an essential piece of configuration management, which defines the dependencies between a service or a Service Portfolio Catalogue and all the underlying service assets required to run the service. The current EOSC model needs to be revised and adapted to the user requirements with the new EOSC Nodes topology.

  User Story:
  As a Service Provider of a new core service or new EOSC Node
  I want to know which configuration information for my services is required to be provided to the EOSC EU Node
  so that I can plan how the service/node should be included in the EOSC ecosystem.

- Feature 19: Adapt the CMDB to the Service Data Model

  The current GOCDB implementation was tailored to be used in federated environments but lacks many of the modern functionalities and services metadata schema required to run under the EOSC Service Data Model. A new CMDB, based on a federated model able to support the adopted EOSC Service Data Model, should be implemented.

  User Story:

As an EOSC EU Node/Service provider
I want to have an information system that allows me to see my services and their dependencies
so that I have one single entry point to view my service and dependencies.

- Feature 20: Improve the EOSC configuration information
  With the current implementation, not all service operational functionalities are allowed

  As an EOSC EU Node SMS staff/EOSC third-party provider
  I want to view and edit all operational information about a production service (e.g. the roles of staff, service endpoint, service capacity, downtime, or broadcast to users)
  so that I have a single entry point for editing information relevant to the operation of a service.

- Feature 21: Support of the EOSC Interoperability Framework and Deployment Service
  The Configuration Management system (CMS) should support the EOSC Interoperability Framework by delivering accurate up-to-date configuration of the EOSC-Core services, their components and relationships to facilitate the design and definition of overall federated EOSC architecture, composability and integration of resources

  User Story:
  As an EOSC EU Node SMS staff/EOSC third-party provider
  I need to provide information about a service in order for the service deployment to be able to make automated provisioning of services computing infrastructure
  so that automatic service provisioning of services will be possible.

Table 11 - CONFM process objectives, activities, and expected outputs

Change management and Release and deployment management (ChaRDM)
Objective
Ensure that all changes to the Configuration Items of the EOSC EU Node are properly handled and monitored. It also provides guidelines on how to deploy into production new services and/or updates.
Activities involved in this process
● Make sure all changes are registered, classified and monitored. ● Monitor the deployment of new or changed services and service components to the production environment. ● Evaluation of the releases.
Expected process results
● Provide and maintain a logical model of all configuration items and their relationships and dependencies. ● Provide ways to verify if the information is handled correctly. ● Verify periodically that the information in the CMDB is still valid.
Opportunities for innovation
● Feature 22: Integrate Change and Release Management with the CMDB All changes managed by the Change Management Process should be reflected in the CMDB for the CIs involved in the changes. For each CI in the CMDB, a list of attributes, for which the history of changes is documented, should be defined. Currently, the SMS changes and releases are tracked in Jira without any automatization procedure and without control from the service owners. User story: **As a** service provider **I want** to have access to the history of changes of service together with the current service information (e.g. functionalities available or version) **so that** I keep informed of any changes, have access to the current service information and have a fast response in case of a problem caused by a change.

Table 12 - ChaRDM process objectives, activities, and expected outputs

Continual Service Improvement (CSI)
Objective
Identify, record, prioritise, evaluate, approve and review opportunities for improvement for the EOSC EU Node.
Activities involved in this process
● Coordination of the activities for the improvement of the services ● Report on the improvement decisions and implementations
Expected process results
● Suggestions for improvements related to Service Management or either technical or non-technical service improvements.
Opportunities for innovation
● Feature 23: Organised registry of suggestions for improvement Automatically apply filters to organise the suggestions for improvement by service or area of improvement in a backlog. It is as simple as adding a filter in a submission form. Now they are received by email or expressed verbally and manually created and manually organised. User story: **As** EOSC EU Node SMS staff **I want to** know what are the suggestions for improvement for the SMS processes or services after they have been automatically organised (e.g. per process) **so that** I can plan and develop improvements to SMS processes/services.

Table 13 - CSI process objectives, activities, and expected outputs

2.5. Initial assessment of ECP SMS innovation opportunities

The ECP SMS innovation activities are assessed considering the alignment with the EOSC Beyond project goals, the impact vs complexity, the customer value, the available resources to develop them, and dependencies.

In order to do the assessment, stakeholders' input was also taken into account. The ECP SMS development team provided feedback to help us understand what would be most helpful to them. There might also be a need to collaborate with EU Node Procurement Lots 1, 2 and 3 to understand more details on how they operate, even if an initial analysis was conducted and a questionnaire provided by T5.4 was completed by Lot 1.

Also, EOSC Beyond team members, who would be involved in the development of innovative features, provided feedback to determine what is feasible, and how much effort is involved. Additionally, a T5.4 representative attended the workshop 'Federation of nodes, technical aspects, and types of federating capabilities' organised by EOSC Beyond WP15 on June 10-12, 2024. As a result of the attendance, it was possible to identify whether requirements by EOSC Node pilots could be supported or trigger SMS innovation opportunities.

Table 14 shows the most relevant parameters for assessing innovation activities. Alignment with EOSC Beyond objectives is a default requirement. Using the descriptions in the user stories, a combination of impact and customer value is identified, summarised in the column ‘impact’, and complexity is assessed. By using t-shirt sizing, impact and complexity are initially estimated as follows:

Impact

- **Impact S**: benefits internal or EOSC SMS staff directly or indirectly.
- **Impact M**: benefits internal EOSC SMS staff and service providers directly or indirectly
- **Impact L:** benefits most EOSC SMS staff, Service providers and EOSC end users[2] directly or indirectly.

Complexity

- **Complexity S**: Simple problem. Well-known standard procedures, and practices to develop a solution. It does not exceed the capacities available.
- **Complexity M**: Complicated problem.It needs further analysis and design to verify if it does not exceed the capacities available but there is a certainty on how to develop a solution.
- **Complexity L**: Complicated problem. It needs further analysis and design to verify if it does not exceed the capacities available and there is a degree of uncertainty in how to develop a solution which will require a Proof of Concept (PoC).
- **Complexity XL**: Very complicated problem that requires further analysis and design. Exceeds the capacities and there is a degree of uncertainty in how to develop a solution.

[2] End users are individuals doing research (including academic and private sector researchers) interested in using services offered by EOSC. They are not organisations or Research Infrastructures offering services through EOSC or interested in services offered by EOSC.

Following the initial impact and complexity assessment, whose values may change when WP7 conducts an in-depth analysis of the features and their requirements, the prioritisation could be assigned using a matrix[3] of these two values.

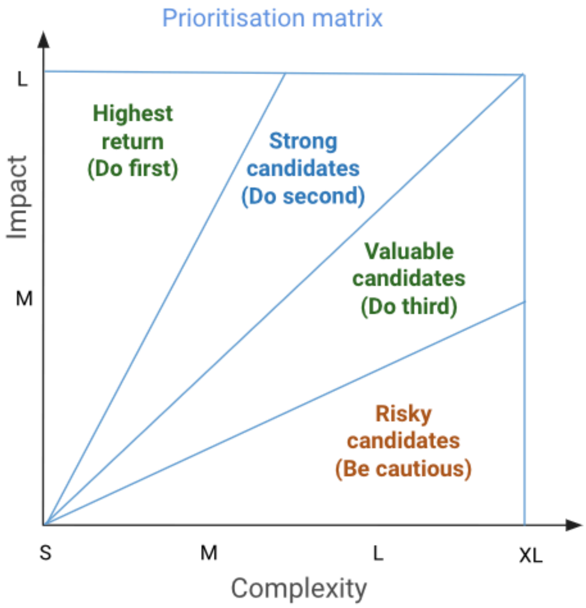

Prioritisation matrix
Impact
L
M
Highest return (Do first)
Strong candidates (Do second)
Valuable candidates (Do third)
Risky candidates (Be cautious)
S
M
L
XL
Complexity

Figure 4 - Prioritisation matrix based on impact and complexity.

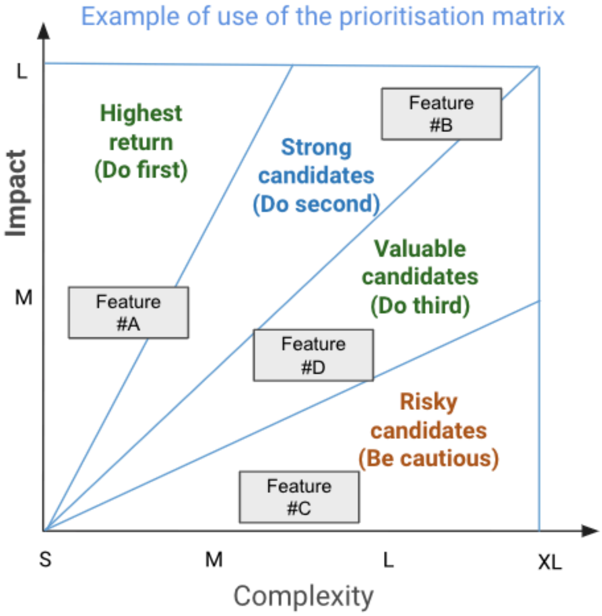

Example of use of the prioritisation matrix
Impact
L
M
Highest return (Do first)
Strong candidates (Do second)
Valuable candidates (Do third)
Risky candidates (Be cautious)
Feature #A
Feature #B
Feature #C
Feature #D
S
M
L
XL
Complexity

Figure 5 - Example of use of the prioritisation matrix based on impact and complexity.

[3] Corrected Impact vs Effort prioritisation chart by Clayton Kjos.

Initial assessment of the impact and complexity of ECP SMS Innovation activities			
Feature #	**Description**	**Impact**	**Complexity**
1	Automated synchronisation of the EOSC Core and EU Node EOSC Exchange services portfolios.	S	L
2	Automated onboarding and auditing procedures for third-party EOSC Exchange providers and services	M	M
3	Automated creation of a third-party EOSC Exchange service portfolio	M	M
4	Automated synchronisation of the third-party EOSC Exchange services portfolio and catalogue	M	L
5	Automated synchronisation of the EU Node EOSC Exchange services portfolio and catalogue	M	XL
6	Visibility of Service Level Agreements and Operational Level Agreements	M	L
7	Automated the service performance evaluation against SLAs	L	XL
8	Store or allow download of the service performance evaluation against SLAs	S	L
9	Automated messaging when SLAs are at risk	M	L
10, 11, 12	Automated report generation (including service availability and continuity reports and capacity reports)	M	L
13	Automated notification of security incidents	M	XL
14	Automated user satisfaction survey	M	XL
15	Automated Knowledge Database (KB) and Frequently Asked Questions	L	L
16	Automated reply to FAQ	L	L
17	Automated Known Error Database (KEDB)	M	XL
18	Improve the EOSC Service Data Model	M	L
19	Adapt the CMDB to the Service Data Model	M	L
20	Improve the EOSC configuration information	M	L
21	Support of the EOSC Interoperability Framework and Deployment Service	M	XL
22	Integrate Change and Release Management with the CMDB	M	L
23	Organised registry of suggestions for improvement	S	M

Table 14 - Initial assessment of the impact and complexity of ECP SMS Innovation activities

As a result of this initial assessment of impact versus complexity and the prioritisation given, the WP7 will use the features wishlist as an input for their requirements gathering, as well as the requests from other stakeholders, to gather requirements and for planning the development of the services within the scope of EOSC Beyond.

3. SQA activities

Quality is a very important trait in Software development and release; thus, it has been made part of this delivery. Each software component developed on the EOSC Beyond project will be required to follow the Software Quality Criteria set by Task 5.5 to be part of a release. The mandatory and optional Quality Criteria are detailed in this section. The criteria assessment will be either performed through a Version Control System repository with the help of the "Software Quality Assurance as a Service" (SQAaaS) from the EOSC Synergy project, executing CI/CD pipelines using a Jenkins service or from other source provided by software team and validated by the project.

The procedure and guidelines are described in Confluence Wikis (**Annex**), and the software components for the EOSC Beyond release will be tracked in Jira tickets under the control of Task 5.5.

3.1. Deployment and Release Management

This section assumes that a microservices approach will be taken with a controlled version deployment of components, for example using Docker containers deployed to Kubernetes clusters or using Ansible recipes targeting virtual machines. Thus, any given service consists of one or more components, with a component being the lowest level of Configuration Item concerning the deployment and release of services. It is also assumed that the majority of development teams use environments with version control. Those that do not have this setup in place are still expected to comply with the SQA and testing requirements but will need to use alternative methods to enable Continuous Integration and the associated automated testing and local deployment.

The software release procedure is described in [3].

To avoid using the terms 'deployment' and 'release' interchangeably, each process will be explained in different sections.

3.1.1. Deployment Management

"The primary objective of deployment is to place the software into a specific environment, ensuring it is functional and operational. It involves tasks like provisioning infrastructure, configuring settings and setting up the required runtime environment." [4].

There are many steps in the deployment management process and a number of well-known tools are available to underpin it. The starting point is strict version control. Without it, it is impossible to know what has been deployed where.

On EOSC Beyond there will be three deployment environments: development, testing and pre-production.

The development environment will be deployed locally by each software team. As these 'local' environments (and their associated code bases, test suites and test results) may not be accessible outside of their organisations, occasional audits will be performed to ensure each one complies with the requirements stated in this document, particularly for unit testing, SQAaaS, and the management of the software and dependencies.

The testing environment is where integration, functional and performance management testing takes place. When the test results are satisfactory, the combined components and hence the release are considered stable and can be deployed to the pre-production environment, which is used for 'play and display', i.e. it is a shop window for new and enhanced services. The configuration of the testing and pre-production environments should be identical, insofar as that is possible.

In order to ensure deployment can be done easily and consistently across multiple environments it should follow an Infrastructure as Code (IaC) approach and the deployment files (for example with tosca templates through the IM and Ansible roles and playbooks, or helms) should be subjected to version control. Preferably a configuration and deployment management tool should be used as part of the deployment pipeline since it ensures reproducibility.

The testing and pre-production deployment environments should be instrumented so that deployed services are observable. Metrics, logging and aggregation, monitoring and alerting are all necessary, in order to observe the performance and reliability of the infrastructure.

- Metrics - the raw measurements of resource usage or behaviour that can be observed and collected within a development environment.
- Logging - source code should use a logging framework and meaningful messages should be emitted (in English, by default) to help with debugging, error handling etc. Typical logging levels are FATAL, ERROR, WARN, INFO, DEBUG and TRACE (severity levels from high to low) and the default level should be set as appropriate for a given deployment environment.
- Aggregation - The practice of collecting, standardising, and consolidating log data. This is necessary when multiple components are emitting log messages in a given environment.
- Monitoring - The existing EOSC Beyond Monitoring Service will be used to record and make available the availability and reliability of deployed services.
- Alerting - alerting definitions have two components: a metrics-based threshold, and an action to perform when values exceed the threshold.

3.1.2. Release Management

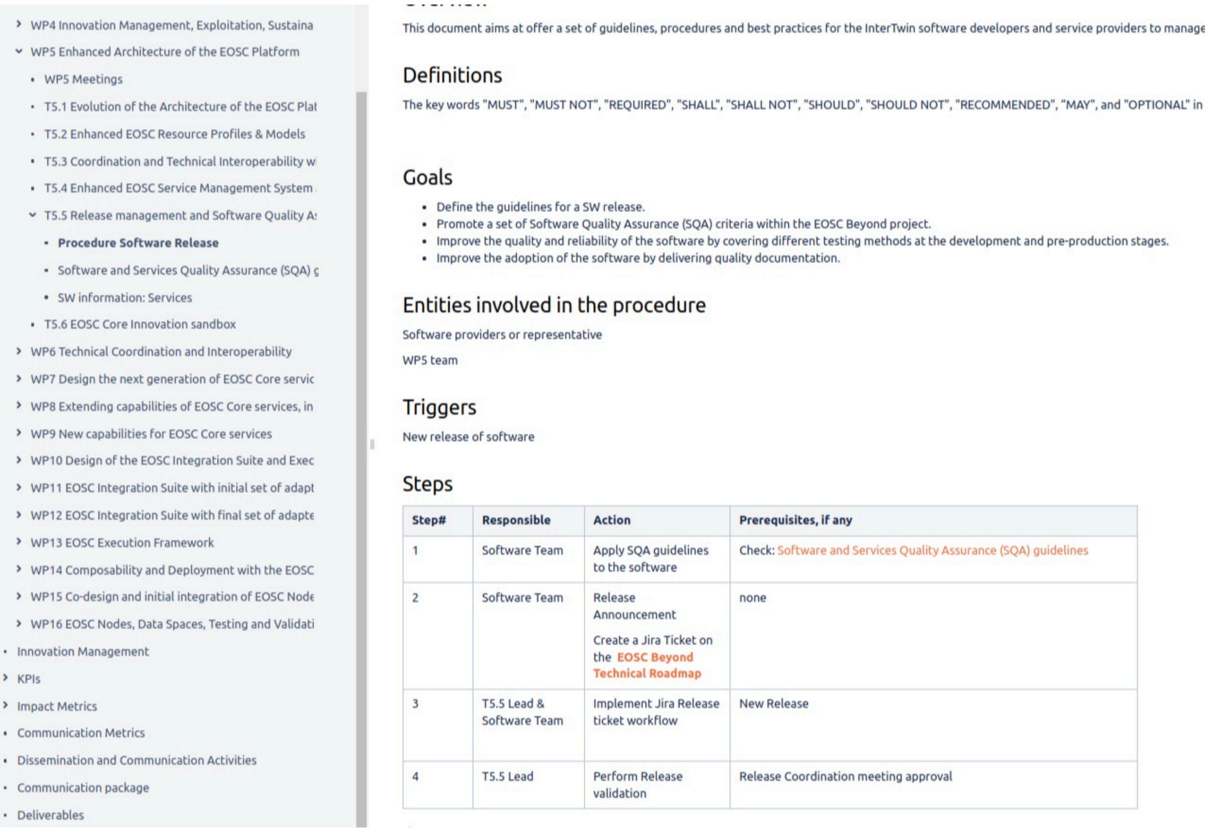
"The main objective of releasing software is to make it available to end-users, customers, or other stakeholders. It involves tasks like quality assurance, bug fixing, and finalising the software for general availability." [4].

The software release procedure has been established and the full description can be found in the following confluence page:

- https://confluence.egi.eu/display/EOSCBeyond/Procedure+Software+Release[4]

[4] Access restricted to EOSC Beyond Consortium members

The procedure (Figure 6) describes the goals, entities involved and necessary steps towards a software release. 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 EOSC Beyond.



- WP4 Innovation Management, Exploitation, Sustaina
- WP5 Enhanced Architecture of the EOSC Platform
  - WP5 Meetings
  - T5.1 Evolution of the Architecture of the EOSC Plat
  - T5.2 Enhanced EOSC Resource Profiles & Models
  - T5.3 Coordination and Technical Interoperability wi
  - T5.4 Enhanced EOSC Service Management System
  - T5.5 Release management and Software Quality A:
    - **Procedure Software Release**
    - Software and Services Quality Assurance (SQA) g
    - SW information: Services
  - T5.6 EOSC Core Innovation sandbox
- WP6 Technical Coordination and Interoperability
- WP7 Design the next generation of EOSC Core servic
- WP8 Extending capabilities of EOSC Core services, in
- WP9 New capabilities for EOSC Core services
- WP10 Design of the EOSC Integration Suite and Exec
- WP11 EOSC Integration Suite with initial set of adapt
- WP12 EOSC Integration Suite with final set of adapte
- WP13 EOSC Execution Framework
- WP14 Composability and Deployment with the EOSC
- WP15 Co-design and initial integration of EOSC Node
- WP16 EOSC Nodes, Data Spaces, Testing and Validati
- Innovation Management
- KPIs
- Impact Metrics
- Communication Metrics
- Dissemination and Communication Activities
- Communication package
- Deliverables

This document aims at offer a set of guidelines, procedures and best practices for the InterTwin software developers and service providers to manage

Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in

Goals

- Define the guidelines for a SW release.
- Promote a set of Software Quality Assurance (SQA) criteria within the EOSC Beyond 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

WP5 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 **EOSC Beyond Technical Roadmap**	none
3	T5.5 Lead & Software Team	Implement Jira Release ticket workflow	New Release
4	T5.5 Lead	Perform Release validation	Release Coordination meeting approval

Figure 6 - Procedure for the software release within EOSC Beyond

A Jira project [5] will be used to track the technical roadmap and to facilitate the release planning. It will document the planned and actual release dates, their versions and the versions of the components that they contain. The assessment will be performed in several stages according to the corresponding stage of the deployment environment together with proofs of compliance. At the date of the delivery, the full workflow is still not finalised but input from EOSC Beyond Technical Coordination Board and user requirements from WP15 are being assessed.

There are three main sections regarding the SW release in EOSC Beyond:

- **Mandatory criteria Stage 1** for the SW release. This stage will be performed once a version is pushed to be released and must be met in order for a release to be deployed into the development stage. The quality tests include criteria for code handling like source code management or code release best practices.
- **Mandatory criteria Stage 2** for the SW release. This second block of quality will include dynamic testing like API testing, integration testing and functional tests and can be either performed on the development infrastructure like the API, some functional tests or even some security tests. The remaining tests must be done on the testing infrastructure.

- **Optional Criteria** available for obtaining Badges using SQAaaS service. This optional set of criteria can either be for software or Services but for completeness we only mention the service badges.

In Figure 7 we can see a simplified view of a typical workflow of a release cycle on EOSC Beyond. In this cycle, once a release candidate is pushed to the repository the QA Stage 1 tests run and if successful it will be deployed into the development environment on which the other quality assurance tests can run. If tests are successful, the release it's then ready to be deployed in the testing environment. Once the release is in the testing environment the remaining of the QA stage 2 can be executed and if they are according to the Stage 2 QC criteria the release is considered implemented and ready to be deployed into the pre-production environment. The release will be considered validated once it's deployed successfully into the pre-production and considered ready by the Release Coordinator. If a release fails any of the stages it will be considered not ready and a new release cycle will be required to the software development team.

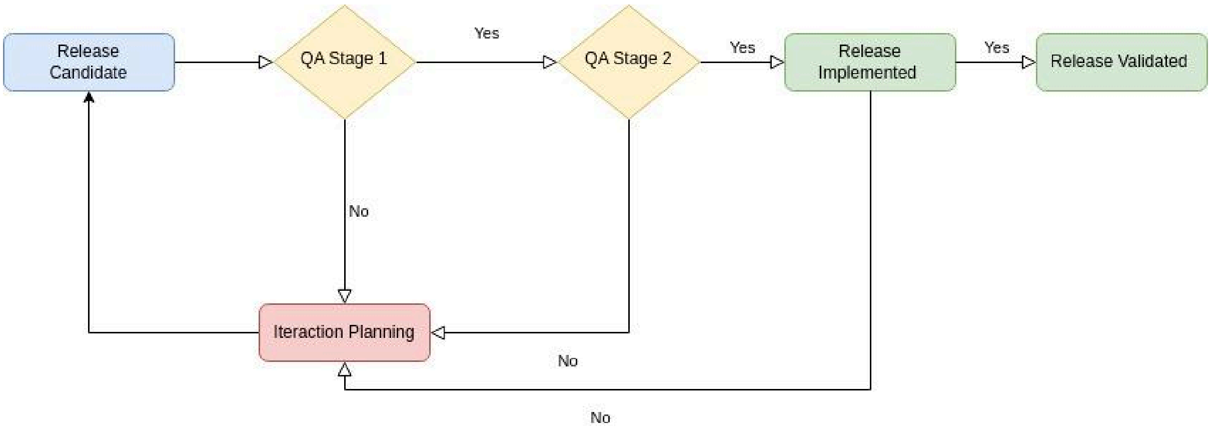


Figure 7 - A simplified view of a software release life cycle on EOSC-Beyond

3.2. Software Quality Assurance Criteria

In the framework of Task 5.5, we have defined a set of Quality Criteria Attributes (QCA)that software components have to pass to be included in the EOSC Beyond release. This quality criteria set was discussed and agreed upon with the developer teams. The document: "A set of Common Software Quality Assurance Baseline Criteria for Research Projects" [7] has served as the basis for the QCAs in EOSC Beyond.

3.2.1. SQA Mandatory attributes

The SQA mandatory criteria are split into two groups according to the type of tests. The first set of mandatory attributes (QA Stage 1) is dedicated to code handling and Static testing also known as white testing while the second group is dedicated to services and incorporates performance and functional testing also known as black box testing.

Once again worth mentioning that the QCAs mentioned in this delivery are abstract in the sense that they are agnostic with respect to the technologies and services used to assess them.

SQA Stage 1

Test to be performed when a new release is triggered.

- **Code Accessibility [QC.Acc]**: meaning code software must be open and accessible
- **Code Workflow [QC.Wor]:** A change-based approach is accomplished with a branching model like main, stable, feature etc.
- **Code Management [QC.Man]:** Issues must be tracked so having an issue tracking system facilities software development.
- **Semantic Versioning [QC.Ver]:** semantic version[5] should be the same across all software teams
- **Licensing [QC.Lic]** Licensing should be clearly stated and open source-source software is a MUST.
- **Documentation [QC.Doc]** Documentation should be prepared according to the target user: user or sys admin.

SQA Stage 2

Attributes that a software component must comply with. The tests can run either on the development infrastructure or the testing infrastructure.

- **API Testing [SvcQC.Api]** API testing tests should cover validating the features outlined in the specification (aka contract testing).
- **Integration Testing [SvcQC.Int]** Whenever a new version of a component is deployed to the test environment, it should be tested to ensure it works correctly with the other components it interacts with. These tests whenever possible should be automated.
- **Functional tests [SvcQC.Fun]**: Functional testing is a type of black-box testing. It involves the verification of the Service identified functionality, based on requested requirements and agreed design specifications. This type of Service testing focuses on the evaluation of the functionality that the Service exposes, leaving apart any internal design analysis or side effects to external systems.
- **Performance tests [SvcQC.Per]:** Performance tests check whether a service can meet requirements such as supporting a specified number of users simultaneously, responding to user input within a specified time period etc.

3.2.2. SQA Optional Criterias

The Quality Criteria Attributes (QCA) are for the development teams that wish to obtain badges awarded by EOSC-Synergy through the SQAaaS[6]:

[5] Semantic Versioning 2.0.0, URL: https://semver.org

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

In addition to the mandatory QC, the following QC should be met in order to obtain a badge. There are three badges: Gold, Silver and Bronze. The attributes required to obtain a badge are the following:

Bronze badge:

Stage 2 must be met plus:

- **Documentation [QC.Doc]:** extra attributes required like treating documentation as a code (version controlled) code of conduct and ready to accept external participation.

Silver badge:

Bronze badge QCA must be met, plus:

- **Security Dynamic Analysis [QC.Sec]:** security assessment by external tools focuses on the service runtime analysis of the established security-related requirements, as part of the Dynamic Application Security Testing (DAST).

3.2.2.3 Gold badge:

Silver badge QCA must be met, plus:

- **Style [QC.Sty]** Code style requirements which 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.

3.3. Summary of the SQA criteria

The following table presents a summary of the QCA together with support from SQAaaS tools for the automatic assessment of specific quality metrics.

Quality Criteria	STAGE 1	STAGE 2	OPTIONAL	Support from SQAaaS[7]
Code Accessibility [QC.Acc]	🟩			supported
Code Workflow [QC.Wor]	🟩			no
Code Management [QC.Man]	🟩			no
Semantic Versioning [QC.Ver]	🟩			supported
Licensing [QC.Lic]	🟩			supported
Documentation [QC.Doc]	🟩			supported
API Testing [SvcQC.Api]		🟩		Partially
Integration Testing [SvcQC.Int]		🟩		partially

[7] https://docs.sqaaas.eosc-synergy.eu/quality_assessment_and_awarding/synergy_badging_approach

Quality Criteria	STAGE 1	STAGE 2	OPTIONAL	Support from SQAaaS[7]
Functional tests [SvcQC.Fun]		🟩		no
Performance tests [SvcQC.Per]		🟩		no
Metadata [QC.Met]			🟩	supported
Security Dynamic Analysis [QC.Sec]			🟩	supported
Style [QC.Sty]			🟩	supported

Table 15 - Summary of the Quality criteria and requirements per category together with support from SQAaaS

Looking at the table it's clear that the big majority of the SQA metrics adopted on the project are not covered by the current SQAaaS service meaning solutions need to be found in the project before the development starts. Also worth mentioning is that the SQAaaS service also required that the source code MUST use a Version Control System (VCS) Git or SVN. The use of the SQAaaS service it's not mandatory but in the case they use their internal assessment they will need to present proof of compliance.

3.4. SQA next steps

Now that the QCA are defined and agreed by the project it's time to discuss with the TCB how to execute them across the project. While Stage 1 criteria should be straightforward to put into practice since many software teams already comply with them, Stage 2 criteria may present a challenge. Functional tests, Performance tests and Integration Testing will need to take into account the following steps:

Integration Testing [SvcQC.Int]

- Define the scope and define which components are going to be tested;
- Scenario identification: identify one or two scenarios with high-impact
- Technology for the automation tool: Select a tool to automate integration testing. This will be done in conjunction with the Technical Coordination Board (TCB).
- Create test data: After identifying the test scenarios, create test data, both valid and invalid inputs (positive and negative test data) for testing.
- Create the test cases: Ensure there is a test case for each of the scenarios identified above. The inputs and expected outputs must be specified.
- Execute the test cases: Run the tests and record the results. This should be done by the Continuous Integration tool used by the project.
- Analyse the test results: find the root causes of the bugs, if any, and share the feedback with the developers.

Functional tests [SvcQC.Fun]

- *End to end testing: Simulates the end user's interactions with a service via its UI*. The Use Cases or User Stories that describe the requirements of each service form the basis of these tests, which should use an existing framework (such as Selenium) to walk through the actions a User would take in order to achieve a goal. Some of the requirements of this project are also requirements of the Lot 1 procurement, so the possibility to share the relevant parts of their end-to-end testing suite is being explored.

- *Sanity testing: Ensure the service works as expected after new features have been added.* Logically this is a form of regression testing, based on the most important/most frequently used requirements, so a subset of the end-to-end tests should be used first, for broad but shallow testing. If they pass, then the remaining end-to-end tests are applied.
- *Smoke testing: This is to sanity testing what sanity testing is to end-to-end testing.* This can be achieved by testing the top one or two requirements that define the essence of the service and moving on to sanity testing if they pass.

Performance tests [SvcQC.Per]

- Performance tests check whether a service can meet requirements such as supporting a specified number of users simultaneously, responding to user input within a specified time period etc.

This project does not have any such requirements, so it will investigate the possibility of adopting some or all of the performance requirements of the Lot 1 procurement.

4. Conclusions and next steps

4.1. SMS conclusions

The objective of the ECP SMS is to structure, plan, implement, monitor, and continually improve all service management processes that allow services and operational roles in delivering the minimum level of the ECP services and supporting interaction with all Exchange services (services provided either by EU Node procurement contractors or third party providers that will be onboarded on the EOSC EU Node Exchange).

Actions for innovation have been identified after the analysis of the current EOSC EU Node SMS implementation that is part of the EC-funded EOSC Lot 1 procurement contract ‘Managed Services for the Development, Integration, Deployment and Operations of the Federated EOSC Core Platform', which will be operational by autumn 2024 and that is not yet fully implemented. It is important to note that due to the parallel timeline of the innovation activities within the EOSC Beyond project and the EOSC procurement work, the innovations developed by EOSC Beyond will not be included in the EU Node's current implementation that is subject to procurement contracts, but will be available for future developments of the EU Node. Also we have considered the needs by EOSC Node pilots and other stakeholders within the EOSC Beyond project.

Innovations proposed aim to maximise impact and value for SMS staff, service providers, and end users. The proposal for innovations is based on the information available to date. Due to the parallel development of the EOSC EU Node SMS mentioned above and the pilot stage of the Nodes, there is a degree of uncertainty.

The assessment and further investigation of requirements related to these innovations will be carried out by WP7 and T5.4 will act as stakeholder. In the second half of the EOSC Beyond project, there will be an update on the status of innovation features that will be presented in D6.1 SMS Status and SQA activities.

4.2. SQA conclusions

Software Quality Assurance is essential in order to reduce code defects (and hence reduce the time spent to produce a stable release) and continuously improve the code base that underlies the various service components. In addition, they help to clarify some of the IPR issues relating to the reuse of the code, by clearly displaying a licence (which should be compatible with the licences of the associated 3rd party software dependencies).

Responsibility for Software Quality Assurance lies primarily in two places: with the teams that develop individual services and their components; and with the Software QA team that ensures all of the components of a release work together as expected and are stable enough to deploy to the pre-production environment.

The activities of Task 5.5 during the first months of the project established a Release Management procedure together with the definition of the Quality Assurance Attributes.

The approach followed and explained in this document gives the development teams the flexibility that they need to manage their own local environments and use the programming languages, scripts, 3rd party libraries and frameworks, tools and techniques that they are familiar with. However, they are bound by the requirements set both in the **3.2 Software Quality Assurance Criteria** section. Audits will be performed from time to time to ensure compliance.

The next steps will be to make sure the plan explained here is fulfilled together with the start of the Release Coordination regular meetings. An update of those activities will be later presented in D6.1 SMS Status and SQA activities.

5. Vocabulary, acronyms, and abbreviations

Term	Definition
Authentication and Authorization Infrastructure Architecture (AAI)	Source: https://eosc.eu/advisory-groups/aai-architecture/
Capacity Management (CAPM)	*Source: FitSM Overview and Vocabulary* *https://www.fitsm.eu/download/748/?tmstv=1718880360*
Change management and Release and deployment management (ChaRDM)	*Source: FitSM Overview and Vocabulary* *https://www.fitsm.eu/download/748/?tmstv=1718880360*
Configuration Item (CI)	The term is applied to a product, an allocated component of a product, or both that satisfies an end use function, has distinct requirements, functionality and/or product relationships, and is designated for distinct control in the configuration-management system. *Source: Wikipedia*
Configuration Management System (CMS)	This term is applied to the whole ecosystem system responsible for the configuration of the system (procedures, tools, services etc)
Configuration Management (CONFM)	*Source: FitSM Overview and Vocabulary* *https://www.fitsm.eu/download/748/?tmtv=1718880360*
Configuration Management Database (CMDB)	A system used to store information about hardware and software assets (commonly referred to as configuration items). *Source: Wikipedia*
Continual Service Improvement (CSI)	*Source: FitSM Overview and Vocabulary* *https://www.fitsm.eu/download/748/?tmtv=1718880360*
Continuous Integration	The practice of integrating source code changes frequently and ensuring that the integrated codebase is in a workable state. *Source: Wikipedia*
Customer relationship management (CRM)	*Source: FitSM Overview and Vocabulary* *https://www.fitsm.eu/download/748/?tmtv=1718880360*
Deployment Management	The process of making software available on target environments, such as servers, cloud platforms, or devices, for execution.

Term	Definition
	https://medium.com/cloud-native-daily
DevOps	Integrates and automates the work of software development (Dev) and IT operations (Ops) as a means for improving and shortening the systems development life cycle. *Souce: Wikipedia*
End user of EOSC services	Users are those individuals who access and benefit from the resources exposed through EOSC. *Source:* ***https://ai4eosc.eu/sria/user-environments/#:~:text=Users%20are%20those%20individuals%20who,the%20ones%20interacting%20with%20them.***
European Open Science Cloud Core Platform (ECP)	Platform being developed by the public procurement tender titled "Managed Services for the European Open Science Cloud Platform (EOSC)" (Ref. CNECT/LUX/2022/CD/0023) published under the Horizon Europe Research Infrastructures Work Programme 2022. *Source:* ***https://digital-strategy.ec.europa.eu/en/news/commission-announces-winners-eosc-procurement#:~:text=Lot%201%20%E2%80%93%20Core%20Federation%20Services,A.M.K.E%20and%20Netcompany%20Intrasoft%20SA.***
EOSC European Node (EOSC EU Node)	A reference node for EOSC, provided by the European Commission, that offers access to a diverse range of research outputs, services, and tools. *Source:* ***https://open-science-cloud.ec.europa.eu/***
EOSC Federated Capabilities	Capabilities required or offered across all EOSC Nodes . They are enabled by 1 or more EOSC Nodes. All Federated Capabilities require data to be shared between Nodes in agreed formats via agreed interfaces *Source: 'EOSC Beyond Piloting a network of EOSC Nodes' presentation delivered on 17/06/2024 by Diego Scardaci and Mark Dietrich.*
EOSC third-party provider	A service provider that offers resources to the EOSC but is not a party to EOSC procurement contracts.

Term	Definition
EOSC end users	Individuals doing research (including academic and private sector researchers) interested in using services offered by EOSC. They are not organisations or Research Infrastructures offering services through EOSC or interested in services offered by EOSC.
EOSC Exchange services developed by EU Node procurement	Services being developed by the EU Node procurement Lots 2 and 3.
EOSC Exchange third party services	A service offered to the EOSC by a provider that is not a party to EOSC procurement contracts. At this stage in time, the EOSC Exchange services are supposed to be onboarded to the EU Node.
EU Node Core Services	Services being developed by the EU Node procurement Lot 1, such as: the Web Portal Front Office, the Resource Catalogues and Registry Services, the Application Workflow Management engine, the Federated Identity Management and Single-Sign-On solution, the Monitoring and Accounting function, and the overall Service Management System and service integration. *Source:* ***https://digital-strategy.ec.europa.eu/en/news/commission-announces-winners-eosc-procurement#:~:text=Lot%201%20%E2%80%93%20Core%20Federation%20Services,A.M.K.E%20and%20Netcompany%20Intrasoft%20SA***.
Incident and Service Request Management (ISRM)	*Source: FitSM Overview and Vocabulary https://www.fitsm.eu/download/748/?tmstv=1718880360*
Information Security Management (ISM)	*Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Infrastructure as code (IaC)	The process of managing and provisioning computer data centre resources through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. *Souce: Wikipedia*
Key Exploitable Results (KER)	An identified main interesting result, which has been selected and prioritised due to its high potential to be “exploited”. *Source:*

Term	Definition
	https://intellectual-property-helpdesk.ec.europa.eu/system/files/2022-02/HEU%20Results%20platform.pdf
Minimum Viable EOSC (MVE)	*Source: EOSC Beyond Proposal Part B: Technical description.*
Node Core Services	Required to enable the operation of a given Node. Not all Nodes need all possible Core Services for their own operation. Core Services might be provided by: Components from the EU Node (with copies installed in the Node), improved components from EOSC Beyond (with copies installed in the Node or SaaS), components from other providers. *Source: 'EOSC Beyond* *Piloting a network of EOSC Nodes' presentation delivered on 17/06/2024 by Diego Scardaci and Mark Dietrich.*
Node representative	Person that represents an EOSC Node.
Problem Management (PM)	*Source: FitSM Overview and Vocabulary* https://www.fitsm.eu/download/748/?tmtv=1718880360
Release Management	The process of making the deployed software available to end-users or customers. *Souce:* https://medium.com/cloud-native-daily
Release planning	A collaborative process where the stakeholders determine which product backlog items will be included in upcoming releases. It involves prioritising user stories, estimating effort, and setting release goals, timelines, and scope. *Source:* https://www.scrum-institute.org/
Security officer	*Member of staff that develops tasks on information security management (ISM) defines and manages controls that an organisation needs to implement to ensure that it is sensibly protecting the confidentiality, availability, and integrity of assets from threats and vulnerabilities.* *Source:* https://en.wikipedia.org/wiki/Information_security_management

Term	Definition
Service Availability and Continuity Management (SACM)	*Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Service Level Management (SLM)	*Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Service Management System (SMS)	The idea of a service management system (SMS) is a central concept, where the context of the management system is the organisational context of the service provider, and the subject is to plan, deliver, operate and control (IT) services. *Source: **https://www.fitsm.eu/***
SMS staff	The staff who work on the service management system processes, activities and procedures. *Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Service Portfolio Management (SPM)	*Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Service Reporting Management (SRM)	*Source: FitSM Overview and Vocabulary* ***https://www.fitsm.eu/download/748/?tmtv=1718880360***
Software Quality Assurance (SQA)	A means and practice of monitoring all software engineering processes, methods, and work products to ensure compliance against defined standards. *Source: Wikipedia*
SQAaaS	Software Quality Assurance as a Service. *See:* **https://docs.sqaaas.eosc-synergy.eu/**
Version Control (VC)	The software engineering practice of controlling computer files and versions of files; primarily source code text files, but generally any type of file. *Source: Wikipedia*
Version Control System (VCS)	A software tool that automates version control. *Source: Wikipedia*

Table 16 - Vocabulary, acronyms, and abbreviations

6. References

References	
No	**Description/Link**
[1]	Fitsm.eu. 2024. FitSM-1 Process model [online] Available at: https://medium.com/cloud-native-daily/deploying-vs-releasing-understanding-the-difference-473d04abbd2c
[2]	Clayton Kjos 2023. Replace your 2x2 Impact vs. Effort Matrix! [online] Available at: https://www.claytonkjos.com/blog/t88kggug5sz3p5asz8nx9pymze1d7n
[3]	EGI.eu. 2024. EOSC Beyond software release procedure. [online] Available to project members only at: https://confluence.egi.eu/display/EOSCBeyond/Procedure+Software+Release
[4]	Medium.com. 2023. Deploying vs. Releasing: Understanding the Difference. [online] Available at: https://medium.com/cloud-native-daily/deploying-vs-releasing-understanding-the-difference-473d04abbd2c
[5]	EGI.eu. 2024. EOSC Beyond issue tracker. [online] Available to project members only at: https://jira.egi.eu/browse/EBTCB
[6]	Testsigma.com. 2024. Integration Testing: What it is, Best Practices & Examples. [online] Available at: https://testsigma.com/guides/integration-testing/
[7]	A set of Common Service Quality Assurance Baseline Criteria for Research Projects (DOI https://doi.or/10.20350/DIGITALCSIC/12533)
[8]	The Common Software Quality Assurance Baseline Criteria -[online] available at: http://hdl.handle.net/10261/160086
[9]	Software Quality Assurance as a Service: Encompassing the quality assessment of software and services (https://doi.org/10.1016/j.future.2024.03.024)

Annex A. SQA Criteria

Copy of the confluence Software and Services Quality Assurance (SQA) guidelines

Mandatory Criteria (QA Stage 1)

The following criteria are deemed mandatory for any SW component release to be **moved to QA stage 1** in the Technical Roadmap Jira.

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 [@https://semver.org] 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 licence to be freely used, modified and distributed by others. Non-licensed software is exclusive copyright by default. The licence MUST be compliant with the Open Source Definition [@https://opensource.org/osd].

- **[QC.Lic01.1]** Licences 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:
        - A one-paragraph description of the application.
        - A "Getting Started" step-by-step description of how to get a development environment running (prerequisites, installation).
        - Automated test execution how-to.
        - Links to the external documentation below (production deployment, user guides).
        - Versioning specification.
        - Author list and contacts.
        - Acknowledgements.
- **[QC.Doc06.5]** Developer, when applicable:
    - Private API documentation.
    - Structure and interfaces.
    - Build documentation.
- **[QC.Doc06.6]** Deployment and Administration, when applicable:
    - Installation and configuration guides.
    - Service Reference Card, with the following RECOMMENDED content:
        - Brief functional description.
        - List of processes or daemons.
        - Init scripts and options.
        - List of configuration files, location and example or template.
        - Log files location and other useful audit information.
        - List of ports.
        - Service state information.
        - List of cron jobs.
    - Security information.
    - FAQs and troubleshooting.
- **[QC.Doc06.7]** User, when applicable:
    - Public API documentation.

    - Command Line Interface (CLI) reference.
    - Tutorials, HOW TO's

Mandatory Criteria (QA Stage 2)

The following criteria are deemed mandatory for any SW component release to be **moved to QA stage 2** in the Technical Roadmap Jira.

1. API Testing [SvcQC.Api]

API testing MUST cover validating the features outlined in the specification (aka contract testing).

- **[SvcQC.Api01]:** Any change in the API that is not compliant with the OAS MUST NOT pass contract testing. The use of OAS SHOULD narrow down the applicable set of test cases to the features described in the specification, avoiding unnecessary assertions.
- **[SvcQC.Api02]:** API testing MUST include the assessment of the security-related criteria outlined in [SvcQC.Sec] section.
- **[SvcQC.Api03]:** API testing SHOULD involve the use of test doubles, such as mock servers or stubs, that act as a validation layer for the incoming requests.

2. Integration Testing [SvcQC.Int]

Integration testing refers to the evaluation of the interactions among coupled Services or parts of a system that cooperate to achieve a given functionality.

- **[SvcQC.Int01]:** Whenever a new functionality is involved, integration testing MUST guarantee the operation of any previously working interaction with external Services.
    - **[SvcQC.Int01.1]**: When using APIs, contract testing MUST detect any disruption in the communication between provider and consumer endpoints, through the validation of the API specification SvcQC.Api01.
    - **[SvcQC.Int02]:** Integration testing MUST NOT rely on non-operational or out-of-the-warranty services.
    - **[SvcQC.Int03]:** On lack of automation, ad-hoc pilot Service infrastructures and/or local testbeds MAY be used to cope with the integration testing requirements.
    - **[SvcQC.Int04]:** In the presence of CI environments, integration tests SHOULD be suitable for automated testing.

3. **Functional tests [SvcQC.Fun]**

Functional testing is a type of black-box testing. It involves the verification of the Service identified functionality, based on requested requirements and agreed design specifications. This type of Service testing focuses on the evaluation of the functionality that the Service exposes, leaving apart any internal design analysis or side effects to external systems.

- **[SvcQC.Fun01]:** Functional testing SHOULD tend to cover the full scope –e.g. positive, negative, edge cases– for the set of functionality that the Service claims to provide.
    - **[SvcQC.Fun01.1]** When using APIs, contract testing MUST detect any disruption in the features exposed by the provider to the consumer, through the validation of the API specification.SvcQC.Api01.
    - **[SvcQC.Fun01.2]:** Functional tests SHOULD include the Web Interface or Graphical User Interface (GUI) of the Service.
    - **[SvcQC.Fun02]:** Functional tests SHOULD be checked automatically.
    - **[SvcQC.Fun03]:** Functional tests SHOULD be provided by the developers of the underlying

4. Performance tests [SvcQC.Per]

- **[SvcQC.Per01]:** Performance testing SHOULD be carried out to check the Service performance under varying loads.
- **[SvcQC.Per02]:** Stress testing SHOULD be carried out to check the Service to determine the behavioural limits under sudden increased load.
- **[SvcQC.Per03]:** Scalability testing MAY be carried out to check the Serviceability to scale up or scale down when its load reaches the limits.
- **[SvcQC.Per04]:** Elasticity testing MAY be carried out to check the Serviceability to scale out or scale in, depending on its demand or workload.

SQAaaS Badges (Optional Criteria)

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

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

Bronze badge:

1. QA Stage 1 must be met, plus:

2. 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 in order to establish the positive social attitudes expected within the community of code contributors.

Silver badge:

1. Bronze badge QCA must be met, plus:

2. Metadata [QC.Met]

Metadata for the software component provides a way to achieve its full identification, thus making software citation viable [@doi:10.7717/peerj-cs.86]. 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.
- **Unit [QC.Uni]** Unit testing evaluates all the possible flows in the internal design of the code so that its behaviour becomes apparent. It is a key type of testing for the early detection of failures in the development cycle.
- **[QC.Uni01]** Minimum acceptable code coverage threshold **SHOULD** be 70%.

3. Security [QC.Sec]

A security assessment is essential for any software production. 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) [@https://owasp.org/www-community/Source_Code_Analysis_Tools] 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).

Gold badge:

1. Silver badge QCA must be met, plus:

2. 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.