

**NEANIAS**  
**Novel EOSC services for Emerging Atmosphere,  
Underwater and Space Challenges**

**Deliverable**

---

**Deliverable: D7.3 Software Assessment Methodology**

**30/06/2020**



NEANIAS is funded by European Union under Horizon 2020 research and innovation programme via grant agreement No. 863448.

## D7.3 Software Assessment Methodology

D7.3 Software Assessment Methodology D7.3 Software Assessment Methodology

### Document Info

Project Information			
Acronym	NEANIAS		
Name	Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges		
Start Date	1 Nov 2019	End Date	31 Oct 2022
Program	H2020-EU.1.4.1.3. - Development, deployment and operation of ICT-based e-infrastructures		
Call ID	H2020-INFRAEOSC-2018-2020	Topic	H2020-INFRAEOSC-2019-1
Grant No	863448	Instrument	RIA
Document Information			
Document Id	D7.3		
Document Title	D7.3 Software Assessment Methodology		
Due Date	30-06-2020	Delivery Date	10-07-2020
Lead Beneficiary	SZTAKI		
Beneficiaries (part.)	NKUA, INAF, SZTAKI, CITE, MEEO, ALTEC, GARR, UOP		
Editor(s)	József Kovács (SZTAKI)		
Authors (s)	József Kovács (SZTAKI), Attila Farkas (SZTAKI), E. Sciacca (INAF), C. Bordiu (INAF), Claudio Pisa (GARR), Giorgos Papanikos (CITE), Konstantinos Kakalettris (CITE), Michalis Konstantopoulos (NKUA), Carmela Manetta (ALTEC)		
Contributor (s)	Mel Krokos (UOP), Laura Vettorello (MEEO)		
Reviewer(s)	Nikos Chondros (NKUA)		
Workpackages	WP7-Delivery		
Version	V1.0	Stage	Final
Version details	Revision: 1217 . Last save: 2020-07-10 , 09:56 Pages: 39 . Characters: 62.248		
Distribution	Public	Type	Report
Keywords	EOSC, Security, Underwater Research, Planetary Science, Astrophysics, Atmospheric Research, Software Assessment, Service Management		

## Document Change Record

Version	Date	Change Description	Editor	Change Location (page/section)
1.0	10/7/20	Document version submitted to EC	József Kovács	

## Disclaimer

NEANIAS is a Research and Innovation Action funded by European Union under Horizon 2020 research and innovation programme, via grant agreement No. 863448.

NEANIAS is project that comprehensively addresses the 'Prototyping New Innovative Services' challenge set out in the 'Roadmap for EOSC' foreseen actions. It drives the co-design, delivery, and integration into EOSC of innovative thematic services, derived from state-of-the-art research assets and practices in three major sectors: underwater research, atmospheric research and space research. In each sector it engages a diverse set of research and business groups, practices, and technologies and will not only address its community-specific needs but will also enable the transition of the respective community to the EOSC concept and Open Science principles. NEANIAS provides its communities with plentiful resource access, collaboration instruments, and interdisciplinary research mechanisms, which will amplify and broaden each community's research and knowledge generation activities. NEANIAS delivers a rich set of services, designed to be flexible and extensible, able to accommodate the needs of communities beyond their original definition and to adapt to neighboring cases, fostering reproducibility and re-usability. NEANIAS identifies promising, cutting-edge business cases across several user communities and lays out several concrete exploitation opportunities.



This document has been produced receiving funding from the European Commission. The content of this document is a product of the NEANIAS project Consortium and it does not necessarily reflect the opinion of the European Commission. The editor, author, contributors and reviewers of this document have taken any available measure in order for its content to be accurate and lawful. However, neither the project consortium as a whole nor the individual partners

that implicitly or explicitly participated in the creation and publication of this document may be held responsible for any damage, financial or other loss or any other issue that may arise as a result of using the content of this document or any of the project outputs that this document may refer to.

The European Union (EU) was established in accordance with the Treaty on the European Union (Maastricht). There are currently 28 member states of the European Union. It is based on the European Communities and the member states' cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice, and the Court of Auditors (<http://europa.eu.int/>).

## Table of Contents

<b>Document Info</b> .....	<b>2</b>
<b>Document Change Record</b> .....	<b>3</b>
<b>Disclaimer</b> .....	<b>4</b>
<b>Table of Contents</b> .....	<b>5</b>
<b>Tables of Figures &amp; Tables</b> .....	<b>7</b>
<b>Abstract</b> .....	<b>8</b>
<b>1. Introduction</b> .....	<b>9</b>
<b>2. Software assessment guidelines</b> .....	<b>10</b>
2.1. Development .....	10
2.1.1. <i>Software development</i> .....	10
2.1.2. <i>Testing</i> .....	11
2.1.3. <i>Security</i> .....	12
2.1.4. <i>Data handling</i> .....	13
2.1.5. <i>Integration to core services</i> .....	14
2.2. Operation .....	15
2.2.1. <i>Documentation, Ticketing, Helpdesk</i> .....	15
2.2.2. <i>Monitoring</i> .....	17
2.2.3. <i>Service Deployment</i> .....	17
2.2.4. <i>Security</i> .....	18
2.2.5. <i>Licensing</i> .....	19
<b>3. Guidelines for service management</b> .....	<b>20</b>
3.1. General requirements for services.....	20
3.1.1. <i>Top Management Commitment &amp; Responsibility</i> .....	20
3.1.2. <i>Documentation</i> .....	20
3.1.3. <i>Plan Do Check Act (PDCA) Cycle</i> .....	22
3.2. Process-specific requirements for services.....	24
3.2.1. <i>Service Portfolio Management (SPM)</i> .....	24
3.2.2. <i>Service Level Management (SLM)</i> .....	25
3.2.3. <i>Customer Relationship Management (CRM)</i> .....	26
3.2.4. <i>Release and Deployment Management (RDM)</i> .....	26
3.2.5. <i>Service Availability and Continuity Management (SACM)</i> .....	26
3.2.6. <i>IT Security Management (ISM)</i> .....	27
3.2.7. <i>Incident and Service Request Management (ISRM)</i> .....	27
<b>4. Service Level Agreement</b> .....	<b>28</b>
4.1. NEANIAS Corporate Level SLA .....	28
4.2. NEANIAS Service SLA: Specific per Service .....	29

---

## D7.3 Software Assessment Methodology

---

D7.3 Software Assessment Methodology	D7.3 Software Assessment Methodology
<b>5. Quick checklist for assessment</b> .....	<b>31</b>
<b>6. Conclusion</b> .....	<b>34</b>
<b>References</b> .....	<b>35</b>
<b>List of acronyms</b> .....	<b>36</b>
<b>Appendix I – Service Management Policy</b> .....	<b>37</b>
<b>Appendix II – SMS Capability / Maturity Assessment</b> .....	<b>38</b>

## Tables of Figures & Tables

### *Document Figures*

<b>Figure 1 - PDCA Cycle</b>	<b>23</b>
<b>Figure 2 - SMS Capability / Maturity Assessment scope &amp; goals</b>	<b>38</b>
<b>Figure 3 - SMS Capability / Maturity Process Assessment</b>	<b>39</b>
<b>Figure 4 - SMS Capability / Maturity Assessment Results</b>	<b>39</b>

### *Document Tables*

<b>Table 1 Document Tracking Information</b>	<b>22</b>
<b>Table 2 Example of incident handling response time depending on quality of support</b>	<b>30</b>
<b>Table 3 Quick checklist to assess the key fields related to development and operation</b>	<b>33</b>

## Abstract

The NEANIAS project aims at delivering TRL8 (i.e. system complete and qualified) innovative services in the Underwater, Atmospheric and Space research sectors to contribute to the European Open Science Cloud (EOSC).



This document is deliverable D7.3 Software Assessment Methodology of the NEANIAS project. It contains an overview of various methodologies for assessing software and service components from various aspects.



## **1. Introduction**

The purpose of this deliverable is to collect software assessment methodologies to identify deficiency of software development and operation for Neanias services. These methodologies are formalised as high-level guidelines without low-level details (where possible) to provide applicability for the most type of software and service components. The guidelines intend to formalise recommendations instead of enforcing strict execution of step-by-step procedures, since every possible detail would be hard to be covered. The assessment methodologies collected in this document aim to help improving the quality of the software and service components in the Neanias infrastructure.

The structure of the document is as follows. In Section 2 the focus is on the software development and operation. Several methods to perform assessment are introduced in various fields like source code handling, testing, security, data handling and integration aspects for development or ticketing, helpdesk, monitoring, etc. for operation. This section is the most essential content of the deliverable. Section 3 gives a high-level overview of Fitsm-based service management processes, their requirements and the derived assessment methods. Recommendations are formalised for designing Service Level Agreement for Neanias services. Corporate and per-service SLA templates are introduced in details. In section 5 a checklist is created to collect questions in order to raise attention to the different fields for assessment. This checklist guides the reader to identify and cover the areas where assessment may potentially find any deficiency. The document ends with a Conclusion in Section 6.

## 2. Software assessment guidelines

This section intends to identify the field of development and operation where assessment methodologies can be applied to help improve the quality of the field in question. The high-level assessment methodologies and recommendations are introduced in subsections, where each one covers an aspect of development or operation.

### 2.1. Development

#### 2.1.1. Software development

From very early in the design stage, the software must be designed and implemented using a Service-Oriented Architecture (SOA). This is especially critical for NEANIAS services, since the end-goal is integration with the European Open Science Cloud (EOSC), a service-based system. As also noted in Section 3.1.1, commitment to a SOA should start from the top-level management and apply to all teams, including architects, engineers, and developers.

All NEANIAS services use a version control system to track the development and version history of each service. There is one NEANIAS-specific repository at <https://gitlab.neanias.eu> which uses the Git source code control system; however, services are free to choose any other repository, public or private. The advantage of the NEANIAS repository is that it supports Continuous Integration/Continuous Deployment (CI/CD) via the Gitlab infrastructure (<https://about.gitlab.com>), a method that we will elaborate on later in this document. CI/CD helps improve the quality of the service, and also reduce the cost of development, by identifying issues early so that they are easier to be addressed.

There are many well-known patterns for managing source code branches (e.g., see <https://martinfowler.com/articles/branching-patterns.html>). Each service chooses its own strategy and applies it during initial development and maintenance (after deployment), so that incidents can be addressed promptly.

Versions are tracked in the repository (e.g., using Git tags) so that issues against a deployed version are managed at the proper version branch. Each service chooses its own versioning strategy, such as version numbering and frequency. However, in addition to tags, the service provider may generate releases for stable versions (<https://docs.gitlab.com/ee/user/project/releases>), which are “snapshots” of the code ready to be used/deployed (e.g. already compiled or packaged). The service provider documents each release sufficiently through the “release notes”, so that consumers of the service can identify major/minor updates and the list of changes in the deployed version. Integrations across services target such named, released snapshots that the service provider has declared as stable.

Artefacts of deployment, such as docker images and/or virtual machines, are also tracked in a separate repository, be it the NEANIAS Gitlab Docker registry or the NEANIAS Virtual Machine repository (based on OpenStack Glance). However, any other such service may be selected,

---

### D7.3 Software Assessment Methodology

---

D7.3 Software Assessment Methodology  
such as Docker Hub. This is important so that releases are reproducible in different but compatible environments (e.g., for disaster recovery).

Regarding source code quality, NEANIAS services use, when feasible, static program analysis tools to evaluate software quality. Such tools produce various metrics, such as the cyclomatic complexity of the source code, and the source code coverage of the tests. Depending on the programming language, they may also identify potential issues, such as use of uninitialized variables. Service developers address all such identified issues before any new release of the software. Each NEANIAS service reports what kind of source code analysis is performed during development, to assure its users about the quality of the deployed product.

The assessment of the service development should consider whether the service is following the methodology and guidelines described in section 3 “Software Implementation” and section 9 “General Recommendations and Remarks” of Deliverable D7.1.

#### 2.1.2. Testing

The general approach of the verification methodology is based on the verification of all the requirements listed in the service definitions with different methods according to the nature of the requirement (functional, non-functional) and to the guidelines for the service integration also in an existing environment. The verification methodology will define an operational plan reporting tests identifier, test description and testing procedure.

Validation activities are performed at three levels:

- Unit testing: they aim at validating the correct implementation, functionality and performance of each component of the service;
- Integration tests: within the integrated environments, integration tests are very important since they allow verifying that the communications among the various components are performed correctly;
- Acceptance tests: acceptance tests are end-to-end tests that allow simulating users and administrators' behaviours on the platform; acceptance tests shall cover all the possible cases, including incomplete workflows, corrupted data, and so on.

While unit and integration tests are defined and executed by the implementation team and are highly recommended, acceptance tests are required and have to be agreed with the User Board.

Testing can be performed automatically, thus reducing the cost of finding bugs, through Continuous Integration / Continuous Delivery (CI/CD) pipelines, as the ones provided by the NEANIAS Gitlab. These pipelines can also be leveraged for the streamlined delivery of application updates. Unit and integration tests, as well as build and deployment scripts can be defined through a set of YAML-based definitions and files. For a more detailed insight on this topic, please refer to deliverable D7.1 section 3.3.2.

---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

The assessment of software testing for a specific service should take into account the following parameters:

- whether a testing strategy has been designed;
- whether the following types of tests have been defined, their test coverage and their adherence to the testing strategy:
  - unit tests
  - scalability tests
  - performance tests
  - system tests
  - user acceptance tests

Ideally, newly implemented code and features are initially deployed to the Dev environment, where a basic testing takes place for early identification of bugs and other issues. The most intensive testing takes place in the Staging environment, running complete test suits to detect bugs and corner cases, obtaining performance metrics and, in general, finding whatever needs to be improved before final deployment to Production. Before this, typically the service provider will run a “Smoke Test”, covering the minimal set of functionalities required for the application to work, thus assessing its stability and allowing for a quick rollback if needed.

Web Services testing need to check the functionality, reliability, performance, and security of the underlying Application Program Interface (API). It is similar to unit testing for the software code. A Web Service may be tested manually or creating an ad-hoc automation code (using e.g. curl-based scripts) or using an off-the shelf automation tool (like Postman, see e.g. <https://www.postman.com/use-cases/api-testing-automation/>).

GUI testing may apply the same strategy for web-based applications and desktop based applications developed in NEANIAS. Although manual based testing may be applied in some cases, an automated GUI testing may improve the quality of the service helping testers and developers to perform testing more accurately and within time constraints. Many automated GUI testing tools are available open-source or proprietary (see e.g. <https://www.softwaretestinghelp.com/best-gui-testing-tools/>) with many options for testing web-based and desktop-based applications. For web-based applications, the chosen tool will depend on the target browser compatibility of the service.

#### 2.1.3. Security

Following steps are recommended to develop the NEANIAS services to maintain the confidentiality, integrity and availability of the processed information.

- Requirements review
  - Performing analysis of processed information and provided functionalities in order to address also security requirements.
- Code review

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

- Addressing security design choices after performing application decomposition and threats categorization.
- Integration & Testing Review
  - Automated code review tools and independent code review could be setup and executed as step of the integration and testing phases.
- Deployment Review
  - After system deployment penetration testing could be executed for the most critical services.
- Maintenance Review:
  - Penetration testing could be executed also during maintenance phase with vulnerability scan as described in section 2.2.5.
- Protect code repository.
  - Using an affordable and protected core repository service
- Adhere to the OWASP guidelines (e.g., <https://owasp.org/www-project-top-ten> )
  - This is useful to identify and address well known web exploits early in the development phase.

#### 2.1.4. Data handling

NEANIAS will collect and utilize numerous datasets from a variety of domains such as earth observation, underwater optical and acoustic data, weather data and climate models, (radio)astronomy, and planetary observation missions. These datasets include Research Datasets, Project Output and Software-related data.

Regarding handling the different types of data generated during the NEANIAS project, the FAIR principle should be followed in line with EU expectations and with EOSC rules. FAIR data are data which meet principles of findability, accessibility, interoperability, and reusability. A quick summary of requirements for each of the four principles are as follows:

Findability:

F1. (meta)data are assigned a globally unique and eternally persistent identifier.

F2. data are described with rich metadata.

F3. (meta)data are registered or indexed in a searchable resource.

F4. metadata specify the data identifier.

Accessibility:

A1 (meta)data are retrievable by their identifier using a standardized communications protocol.

A1.1 the protocol is open, free, and universally implementable.

A1.2 the protocol allows for an authentication and authorization procedure, where necessary.

A2 metadata are accessible, even when the data are no longer available.

Interoperability:

---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

- I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (meta)data use vocabularies that follow FAIR principles.
- I3. (meta)data include qualified references to other (meta)data.

#### Reusability:

- R1. meta(data) have a plurality of accurate and relevant attributes.
  - R1.1. (meta)data are released with a clear and accessible data usage license.
  - R1.2. (meta)data are associated with their provenance.
  - R1.3. (meta)data meet domain-relevant community standards.

The way (datasets, methods, responsibilities) how the NEANIAS project will support the requirements listed above is detailed in deliverable D1.5 Data Management Plan. This DMP contains the Data Management Strategy to make NEANIAS data FAIR through detailing the internal policy, open data guidelines and platforms.

FAIR principles should be followed both at project level as well as where it is possible be supported by the software components. According to the guidelines on FAIR Data Management, the FAIR data handling mechanism implemented in NEANIAS is detailed in deliverable D6.1 Core Services Architecture, Design Principles and Specifications. According to this, the relevant thematic and core services will support FAIR data handling 1) through the utilisation of the NEANIAS Service Catalogue and Data Catalogue components realised by Zenodo 2) through the utilisation of standard HTTP REST API and by utilising further techniques detailed in D6.1 in Section 2.7 FAIR principles.

Regarding the assessment of the FAIR data handling, the design of functionalities of the software in question should address the relevant questions and requirements listed above for each of the 4 categories. As a guideline for the design and implementation, the relevant techniques described in D6.1 should be followed.

#### **2.1.5. Integration to core services**

NEANIAS aims to provide a wide range of services across the three thematic research sections it targets. To support these, but also to produce additional value adding services within the context of EOSC as well as to make use of existing EOSC services, there is great value to be gained from investing in the interoperation and reuse of core functionality. Each service is expected to have diverse needs and may need to utilise different underpinning services to support its operation. Still, some core services are expected to provide horizontal functionality needed across most of the offerings.

To ensure conformity, reuse and interoperability with the available offerings, performed service assessment will need to ensure that, at a minimum, some core services will be horizontally utilised by all services. Such core services include:

---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

- AAI Service – The centralised authentication service to apply uniform identity management as well as define high level authorisation policies will need to be utilised across all services
- Accounting Service – To aggregate and publish the necessary accounting information, services will need to push respective traces according to the resources being modelled for the purpose
- Logging Service – The logging service serves as a central repository for low level monitoring and operation troubleshooting. NEANIAS services are expected to integrate with the centralised repository

With respect to the runtime environment and product discoverability, some of the core services to be reused include

- Service Instance Registry – To facilitate discovery and enable seamless integration, the runtime environment of collaborating services will need to be discoverable through the usage and respective updating of the service instance registry
- Research Product Catalogue – Research products will need to be registered within the catalogue, whether metadata only registration is available, or fully registered along with the respective data. More details on data handling can be found in Section 2.1.4.

Parallel to the service level integrations concerning core functionality reuse, at the infrastructure level the following aspects will be assessed

- Compute and storage infrastructure conformance and compatibility will be expected to ensure hosting and operations of the NEANIAS services. This assessment will mostly target the service runtime operation environment, ensuring that the services are able to operate within the provided NEANIAS infrastructure, in addition to any explicit external resources
- Integration with the Monitoring solution employed across all NEANIAS services to track the status and health of the available resources. The monitoring integration is further elaborated in later subsections. From the service development perspective, required hooks may need to be made available to facilitate in depth understanding of a service's operational status

Depending on the core business and use case that each service may need to serve, integrations to respective core or otherwise reusable services will be expected. These services can be part of the NEANIAS offerings, or other EOSC available services.

## 2.2. Operation

### 2.2.1. Documentation, Ticketing, Helpdesk

In this section, the assessment guidelines that are to be considered with respect to the operation of NEANIAS services pertinent to the available documentation, ticketing system usage and respective helpdesk utilization are described. The respective approach to these aspects has already been described, defined and made available through related tools in respective deliverables [5][6].

A wide range of documentation material, relevant to the Service Management System (SMS) employed by NEANIAS is described and detailed in the later section of Guidelines for service management . In this section, the service specific documentation, not directly governed by the

---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

SMS approach (although implicitly required and utilized) is handled. Aspects of the documentation made available that will be part of the assessment process, include:

- Hosted under the common NEANIAS documentation
- Relevant links to external material provided
- Maintained in the dedicated source control system made available for the purpose
- Versioning of the documentation allowing the reader to track the documentation relevant to a reference service
- The content available includes
  - How the service fits in the NEANIAS ecosystem
  - User documentation and manual, extensive at the level relevant to the service application and usage
  - Developer documentation or links to related resources
    - API documentation detailing the usage, endpoints and models, or links to respective documentation
  - Release notes and features available, or links to relevant information
  - EOSC availability and related information
  - Links and information with respect to available deployments, version artefacts, access methodology and policies around usage
  - Any additional content that can further assist the reader to gain better understanding of the status, availability, functionality and integrations available to provide a thorough understanding to the user

With respect to task and progress monitoring, the usage of a suitable ticketing system is expected to assist the development, integration and operation phases of the services. For the inter-service tasks, each service provider can utilise external or even NEANIAS provided ticketing systems to track the work and individual task assignments and progress as well as organise the necessary release procedure.

For intra-service communication, troubleshooting and alignments such as integrations, error reporting, release planning, dependency resolution etc, a single NEANIAS wide repository will be utilised. It is expected that all service providers will ensure that all needed personnel are registered and tracks appropriately all relevant tasks. For the assessment process, several factors can be considered and extracted by monitoring the ticketing system, such as:

- Response time
- Number of open issues
- Number of resolved issues
- Escalation of tickets when required
- Linking of issues to easily identify dependencies
- Relating tickets and resolutions to specific service versions
- Linking of issues to available documentation and known behaviour
- Any other qualitative or quantitative metric relative to the usage, adoption and facilitation of issue resolution through the ticketing system

Similar to the inter-service ticketing, NEANIAS offers a Help Desk facility through which it aims to assist and support its end users. The main issues tackled by the Help Desk are Incident Reports and Service Requests. It is expected that all service providers will ensure that the needed personnel are registered and tracks appropriately relevant reports. For the



---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

assessment process, several factors can be considered and extracted by monitoring the ticketing system, such as:

- Response time
- Number of open issues
- Number of resolved issues
- Proper identification and separate handling of Incidents Reports and Service Requests
- Escalation of issues as required
- Linking of issues to easily identify dependencies
- Relating tickets and resolutions to specific service versions
- Linking of issues to available documentation and known behaviour
- Timely follow-up with additional information, identified resolutions, known workarounds and other helpful information
- Any other qualitative or quantitative metric relative to the usage, adoption and facilitation of issue resolution and service provision

#### **2.2.2. Monitoring**

Monitoring of the NEANIAS services and metrics collection is fundamental to guarantee a reliable service to its users and to meet the service-related KPIs.

To ensure that metrics can be effectively collected, services need to expose monitoring endpoints. The endpoints implementation and the information thereby exposed can be assessed by considering the following parameters:

- whether monitoring endpoints are implemented;
- whether the monitoring endpoints can be accessed and the data correctly consumed by the monitoring tools specified in D7.1, i.e. Prometheus or Nagios;
- whether the collected metrics include the recommended minimal set as specified in D7.1, i.e.: service availability/uptime, number of returning users, used vs. free server resources, total number of (HTTP) requests, server errors;
- whether the collected metrics are functional to measure the target KPIs associated to the service.

To assess the health of a service the following parameters should be considered:

- whether the NEANIAS monitoring and alerting tools (i.e. Nagios, Grafana) show any active alert;
- whether the service can be correctly accessed and consumed;
- whether the service-related logs contain recent/unaddressed error messages;

#### **2.2.3. Service Deployment**

This section deals with the assessment of NEANIAS Service deployments. To this regard, the following parameters should be considered:

- if the service is designed according to a highly available (HA) architecture, whether the service is deployed in a highly available fashion
- if the service is deployed in HA, intentionally making unavailable the service components can assess:
  - the minimal subset of service components needed to keep the service fully operational;

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

- the minimal subset of service components needed to keep the service partially operational.
- whether the service is fully available through:
  - IPv6 only;
  - IPv4 only;
  - Both IPv4 and IPv6.
- whether the service is directly connected to:
  - a commercial network infrastructure;
  - a research network (e.g. GÉANT or an NREN network).
- whether a service backup policy has been defined; whether a backup system has been set up, according to the defined policy, and whether the backup system has been tested.

#### 2.2.4. Security

An ISO27001-like process should be put in place to ensure correct and secure operations of information processing facilities. Documented procedures should be prepared for operational activities associated with information processing and communication facilities.

A process with the following steps is suggested to be executed by whoever manages the production infrastructure. The result of the process is an assessment of security status on production environment.

- Services Vulnerability Scan
  - Vulnerability Management process begins with the execution of security scans, performed through the use of automatic tools updated with the latest critical issues worldwide known, which allows the identification of vulnerabilities in the analysed systems, providing the necessary support to the assessment of the risks associated with them. This assessment makes it possible to determine the corrective actions necessary for the removal of the vulnerability or the elements useful for accepting the associated risk (for example, the exploitation of a vulnerability leads to a negligible damage compared to the cost necessary to implement the corrective actions)
  - The security scan shall meet the following requirements
    - Frequency: The scan shall be performed on at least a quarterly basis. Additional scans can be scheduled if necessary. The execution of the scan shall be scheduled in agreement with the Technical Manager of the Centre, identifying the most suitable moment to minimize the impact on the operations activities. The execution of the 4 vulnerability scans per year (1 every 3 months) has different objectives:
      - 2 scans are considered "primary", i.e. they identify the critical issues and assign actions or carry out binding checks on what has been done
      - 2 scans are considered as control, i.e. they allow monitoring of any new vulnerability to be taken into account, but no specific actions are required.

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

- Depth: Security scans performed with automated tools will have to test services and operating systems to identify weak points in applications. They will also have to analyse missing patches, default passwords and other common vulnerabilities that could be exploited by an attacker. This must be done for all assets, both owned and third-party.
- Data and service interruption: Any security scan should not include intentional searches on the contents of files or determine interruption of services.
- Password Policy
  - A minimum of eight characters and a maximum length of at least 64 characters
  - The ability to use all special characters but no special requirement to use them
  - Restrict sequential and repetitive characters (e.g. 12345 or aaaaaa)
  - Restrict context specific passwords (e.g. the name of the site, etc.)
  - Restrict commonly used passwords (e.g. p@ssw0rd, etc.) and dictionary words
  - Restrict passwords obtained from previous breach corpuses
  - Hashing using strong algorithms (such as SHA512)
- Services should follow the state of the art on cryptography to ensure secure operation
  - Servers shall be configured to use TLS 1.2 and should be configured to use TLS 1.3 as well. These servers should not be configured to use TLS 1.1 and shall not use TLS 1.0, SSL 3.0, or SSL 2.0.
  - The TLS server shall be configured with one or more public-key certificates and the associated private keys. TLS server implementations should support the use of multiple server certificates with their associated private keys to support algorithm and key size agility.
- Log management through NEANIAS centralized service and data analysis-oriented tools enables the execution of problem pattern recognition, fault cause detection, event association analysis and statistics report and dashboard generation focused on security monitoring.

### 2.2.5. Licensing

Software license in IT formalises a regulation on the usage and redistribution of the software. As it has been written in deliverable D7.1 in Section 5, the recommendation is to use Free and Open-Source Software licenses. In case of non-FOSS licenses, the service provider should carefully investigate the situation and make sure that the license enables the usage and operation of the software the way the service provider plans to do. The assessment in this topic is not detailed, however practical recommendations can be found in deliverable D7.1 issued by the Neanias project.

### 3. Guidelines for service management

This section details a Service Management related assessment methodology, where SM is based on FitSM. The approach in the following sections is 1) to investigate the different processes and requirements defined in FitSM, 2) to identify the most important steps to perform during the implementation of service management and 3) to derive the necessary assessment steps to make sure that the necessary service management related activities have been realised. Moreover, the service management processes are used to guide the quality provision of the services and the respective flows and procedures are used to assess the process.

#### 3.1. General requirements for services

##### 3.1.1. Top Management Commitment & Responsibility

A corner stone for the successful implementation of a Service Management System, is the involvement and respective commitment of the top management to planning, implementing, operating, monitoring reviewing and improving the service management system (SMS) and services. The top management, as per FitSM-0 [1] is defined as “Senior management within an organisation who has authority to set policies and exercise overall control of the organisation”.

To show evidence of its commitment, it is requested that the organisation, through its top management will:

- Assign one individual to be accountable for the overall SMS with sufficient authority to exercise this role
- Define and communicate goals
- Define a general service management policy
- Conduct management reviews at planned intervals

A core document through which the respective commitment is declared and elaborated is that of the service management policy. This policy, as per FitSM-0 [1], is expected to be expressed as a “documented set of intentions, expectations, goals, rules and requirements, often formally expressed by top management representatives in an organisation or federation”. Such a service management policy should include:

- A commitment to fulfil customer service requirements
- A commitment to a service-oriented approach
- A commitment to a process approach
- A commitment to continual improvement
- Overall service management goals

A sample of such a Service Management Policy document is presented in Appendix I – Service Management Policy.

##### 3.1.2. Documentation

To support the overall service management system continuity, quality and evaluation process, a number of aspects of the service management process need to be properly and consistently documented. This documentation requirement is horizontal to a number of general as well as process specific requirements. In this section, we aggregate the core required documents that must be generated and maintained across the various sources that produce them.

### D7.3 Software Assessment Methodology

#### D7.3 Software Assessment Methodology

With respect to overall service management system (SMS) documentation, the following three major documents are expected in order to define the high-level scope and plan of the implemented service management:

- Service Management Policy – As described in Top Management Commitment & Responsibility, this document indicates the top down commitment of the organisation to the implementation of the service management plan
- Service Management Scope & Plan – Part of the Service Management planning process of the Plan Do Check Act (PDCA) Cycle as described in the Planning section is the generation and maintenance of the respective documentation

In addition to the overall SMS documentation that is required to set and communicate the high-level goals of the service management, the process specific requirements that are chosen to be implemented need to be defined and documented. The respective subsections under section Process-specific requirements for services describe the selected processes as they relate to the NEANIAS SMS. As an overall requirement, the documentation that will accompany the definition of these processes will cover to varying level of detail the following process aspects:

- Description of the goals of the process
- Description of the inputs, activities and outputs of the process
- Description of process-specific roles and responsibilities
- Description of interfaces to other processes
- Related process-specific policies as applicable
- Related process- and activity-specific procedures as required

It is worth highlighting that the output of these processes as well as the execution of key activities of these processes need to be recorded and documented as a consequence of the iterative and repetitive nature of the Plan Do Check Act (PDCA) Cycle of the implemented SMS.

All generated documents will need to include information that is related to the following activities related to the creation, evaluation, and lifecycle of the document:

- Creation and approval
- Communication and distribution
- Review
- Versioning and change tracking

To cover this requirement, a document control section is suggested to be included within each generated document to facilitate the consistent tracking of this information. This table is a simplified, compact form of tracking that could be further enriched by process owners.

Identifier	[Unique document identifier]
Title	[A descriptive title for the document's content]
Location	[A consistent and immutable location identifier where the document can be retrieved from]
Owner	[Name of the person primarily responsible for maintaining and reviewing this document]
Version	[Version]
Last date of change	[Date]

---

D7.3 Software Assessment Methodology

---

## D7.3 Software Assessment Methodology

Next review due date	[Date]
Version & change tracking	[Version history & simple change log]

*Table 1 Document Tracking Information*

### 3.1.3. Plan Do Check Act (PDCA) Cycle

The Plan–Do–Check–Act (PDCA) cycle [4] is an iterative four-step management method used for the control and continuous improvement of processes and products. The four steps that comprise it are:

- Plan – Establish objectives and processes required to deliver the desired results
- Do – The do phase allows the plan from the previous step to be done. Small changes are usually tested, and data is gathered to see how effective the change is
- Check – During the check phase, the data and results gathered from the do phase are evaluated. Data are compared to the expected outcomes to see any similarities and differences
- Act – This act phase is where a process is improved. Records from the "do" and "check" phases help identify issues with the process. These issues may include problems, non-conformities, opportunities for improvement, inefficiencies and other issues that result in outcomes that are evidently less-than-optimal. Root causes of such issues are investigated, found and eliminated by modifying the process. Risk is re-evaluated. At the end of the actions in this phase, the process has better instructions, standards or goals. Planning for the next cycle can proceed with a better baseline. Work in the next do phase should not create recurrence of the identified issues; if it does, then the action was not effective

The iterative Plan Do Check Act cycle can be visualised as follows (visualisation available at [https://en.wikipedia.org/wiki/File:PDCA\\_Process.png](https://en.wikipedia.org/wiki/File:PDCA_Process.png) under the Creative Commons Attribution-Share Alike 3.0 Unported license <https://creativecommons.org/licenses/by-sa/3.0/deed.en>)

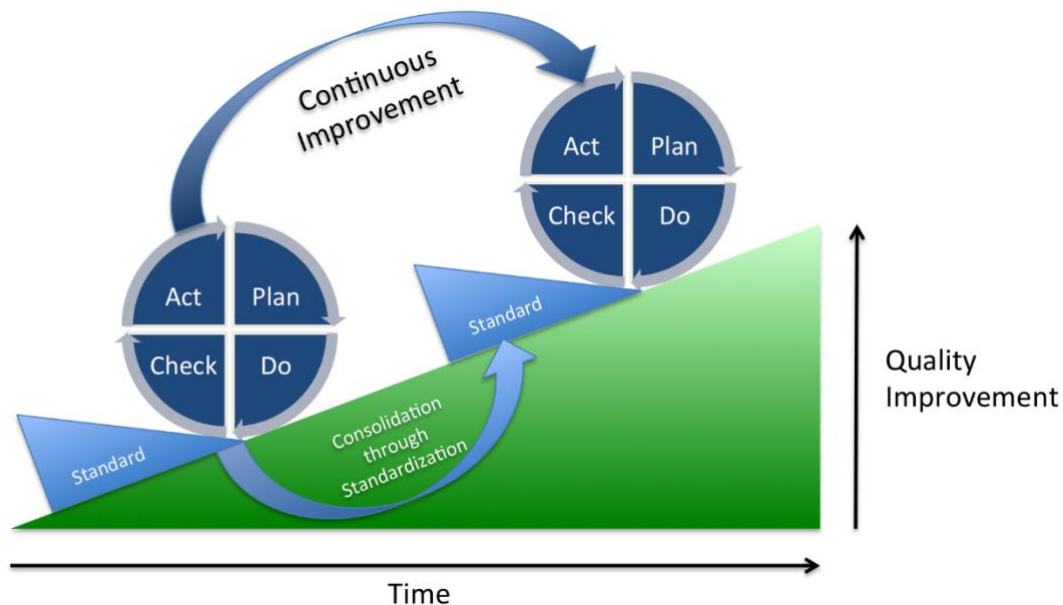


Figure 1 - PDCA Cycle

The respective general requirements that are put in place to facilitate the Plan Do Check Act phases within the context of the service management system, are listed in the following sections.

#### 3.1.3.1. Planning

In the planning phase of the service management system, two major general requirements according to FitSM-1 [2] provide input and help shape the plan.

- Defining the scope of service management
  - The scope of the service management system that will be implemented is defined and a scope statement is created
- Planning Service Management
  - A service management plan is created and maintained addressing at a minimum:
    - Goals and timing of implementing the SMS and the related processes
    - Overall roles and responsibilities
    - Required training and awareness activities
    - Required technology (tools) to support the SMS

#### 3.1.3.2. Doing

In the doing phase of the service management system, the service management plan is implemented and within its scope, the defined service management processes are followed in practice and enforced in accordance to the related policies and procedures. Further definitions and details on the implementation of these processes are available later in the document under section Process-specific requirements for services.

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

#### 3.1.3.3. Checking

In the checking phase of the service management system, the effectiveness and performance of the SMS and its service management processes are measured and evaluated based on suitable key performance indicators in support of defined or agreed targets. Assessments and audits of the SMS can be conducted to evaluate the level of maturity and compliance. Such assessments, within the context of NEANIAS description of work, are also planned to be executed through task T7.5 Services technical assessment, although some service and process specific aspects should be handled internally within each service provider.

The NEANIAS wide KPIs already set for overall monitoring can be used as input for this phase. Additionally, deliverable D7.1 Delivery activities methodology and plan [5] as well as deliverable D7.2 Software delivery infrastructure and tools [6] define explicit Quality Metrics that each service can expose as well as tools to utilise during the operation lifetime of the service in order to facilitate checking additional aspects of its effectiveness and performance.

In addition to service specific KPIs, the service management system itself and its implementation is subject to the Plan Do Check Act cycle. One of the tools that can be used to assist in the assessment of the maturity of the implemented scheme, is described in Appendix II – SMS Capability / Maturity Assessment.

#### 3.1.3.4. Acting

In the acting phase, taking as input the planning and checking phases and their outcome, nonconformities and deviations from targets can be identified and corrective actions can be taken to prevent them from recurring. Improvements can be planned and implemented according to the respective processes put in place.

The output of the relevant process can be documented through the processes that govern the Continual Service Improvement Management and related Change Management and Release & Deployment Management. Furthermore, deliverable D7.1 Delivery activities methodology and plan [5] as well as deliverable D7.2 Software delivery infrastructure and tools [6] define explicit tools and approaches in order for individual service providers to organise and prioritise fixes and enhancements through respective ticketing systems and backlog organisation.

The findings of the checking phase, as these are translated through the acting phase, need to be in alignment with the initial service scope statement or may even require re-validating the service scope in some cases. The planning phase that follows drives the next iteration of the Plan Do Check Act Cycle.

## 3.2. Process-specific requirements for services

In this section the most important assessment steps and fields are listed in relation to the process-specific requirements detailed in FitSM Part 1: Requirements Section 6. Each service must be maintained based on the FitSM processes listed in the next few sections. The detailed procedures will be defined later by WP7, however based on the requirements we have already collected, the assessment steps we suggest in this document are for the service owners to ensure their service is properly handled by the different processes.

### 3.2.1. Service Portfolio Management (SPM)

Service Portfolio Management is the process of maintaining an internal list which details all the services offered by the service provider (those in preparation, live and discontinued). The service portfolio includes meta-information about services such as their value proposition,



---

### D7.3 Software Assessment Methodology

---

#### D7.3 Software Assessment Methodology

target customer base, service descriptions, technical specifications, cost and price, risks to the provider, service level packages offered etc. It is essential that the service provider must make sure that the service in question is properly maintained in the service portfolio. From assessment point of view, the following step is suggested to be performed by the service providers for every service belonging to the provider:

- Make sure the service is properly registered and maintained as part of the service portfolio.
- The dynamic parameters (e.g. status) of new or (changed) service should show proper value in the portfolio
- All static parameters including (timescale, responsibilities, technology, communication, service acceptance criteria) must be properly set for the given service
- The organizational structure supporting the delivery of the service must be properly identified, including contact points of the parties involved

In the portfolio, the most important parameters for the services that should be revised should contain the following information:

- Service overview: the overview has to contain the service name status and a short service description. Also, it should provide information about the users of the service.
- Business case: the business cases should provide the addressed problem and the benefits of the service. Also, it should contain the competitors on the field and the key selling point of the service.
- Service management information: this block should provide information about the service owner and some contact information for further details on the service.
- Service architecture: the portfolio should contain an overview of the service architecture and the inner components and dependencies of the service.
- Finance and resources: finally, the portfolio should provide information about the financial status of the service and the used resources by the service.

#### **3.2.2. Service Level Management (SLM)**

The role of SLM is to maintain a service catalogue, and to define, agree and monitor service levels with customers by establishing meaningful service level agreements (SLAs) and supportive operational level agreements (OLAs) and underpinning agreements (UAs) with suppliers. Therefore, the following assessment steps are suggested:

- The live service must be listed in the catalogue
- The service should own an SLA (see Section 4)
- The SLA for the service should be reviewed at planned intervals
- A detailed plan about the performance evaluation against the targeted SLA is suggested to be prepared for the given service
- For multi provider services, OLAs and UAs are suggested to be created between the parties and these agreements should be reviewed at planned intervals
- In case the service has measurable performance, it is suggested to be evaluated against the targets defined in OLAs and UAs

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

#### **3.2.3. Customer Relationship Management (CRM)**

The customer relationship management is responsible to establish and maintain a good relationship with customers utilising the services. The following assessment steps are suggested by the service provider:

- In the service catalogue, every service customer should be identified
- For each customer, there shall be a designated contact responsible for managing the customer relationship and customer satisfaction.
- Every service description should contain a detailed plan for the following customer-related processes. The most important fields of the plan are:
  - communication mechanism with the customers
  - periodical reviews with the customers
  - service complaints from the customers shall be managed
  - customer satisfaction management

#### **3.2.4. Release and Deployment Management (RDM)**

The main purpose of RDM is to bundle changes of one or more configuration items to releases so that these changes can be tested and deployed to the live environment together. During this process, the whole release procedure has to be defined for each service. The following conditions should be fulfilled when assessment is performed regarding RDM of the service:

- In the service catalogue, every service should own a detailed release management policy.
- The release plan should contain the build and test process before the deployment and shall consider steps to be taken in case of unsuccessful deployment to reduce the impact on services and customers.
- The deployment of new or changed service components to the live environment shall be planned with all relevant parties, including affected customers.
- The release process should contain acceptance criteria, and these shall be agreed with the customers and any other relevant parties.
- Before deployment, the release shall be verified against the agreed acceptance criteria and approved.
- Finally, every release shall be evaluated for success or failure.

#### **3.2.5. Service Availability and Continuity Management (SACM)**

The main goal of SACM is to ensure sufficient service availability to meet agreed requirements and adequate service continuity in case of exceptional situations like unplanned hardware or service failure. The following conditions is suggested to be checked for a given service:

- In the service catalogue, in every service description it is suggested to check the existence of a detailed plan about service availability and continuity management
- The service availability and continuity requirements should be identified considering the SLA
- The service availability and continuity plans shall be created and maintained
- The planning shall consider measures to reduce the probability and impact of identified availability and continuity risks
- An approach to monitor service availability (and continuity) and to record the results on an ongoing basis should be defined

---

## D7.3 Software Assessment Methodology

---

### D7.3 Software Assessment Methodology

#### **3.2.6. IT Security Management (ISM)**

The purpose of the IT Security Management is to manage information security effectively through all activities performed to deliver and manage services so that the confidentiality, integrity and accessibility (CIA) of relevant assets are preserved.

The description of a service in the service catalogue should contain a detailed IT Security Management plan to manage the confidentiality, integrity and accessibility of the stored information. The service owner should consider the following assessment related to ISM:

- The information policies are planned
- Physical, technical and organizational information security control plans shall be prepared in a manner that reduces the probability and impact of identified information security risks
- The information security policies and controls shall be reviewed at planned intervals
- An appropriate priority should be set for the identified information security events and incidents
- A consistent plan for access control should be planned as well

#### **3.2.7. Incident and Service Request Management (ISRM)**

The objective of ISRM is to restore normal or agreed service operation within the agreed timeframe after the occurrence of an incident and to respond to user service requests. In order to properly handle the incidents, the description of ISRM specifies requirements, which should be considered when assessment is performed related to ISRM:

- Incidents and service requests should be registered, classified and prioritized
- Escalation of incidents and service requests should be carried out
- Closure of incidents and service requests should be properly carried out
- There must be a database of information with known errors, workarounds, configuration and release information and an access should be provided for the personnel involved in ISRM
- There should be a definition of major incidents

## 4. Service Level Agreement

Software and service assessment may be defined through a Service Level Agreement. A **service-level agreement (SLA)** allows to define a commitment between the NEANIAS services providers and the final service consumers, end users or groups of customers. Particular aspects of the services, such as the quality, availability and responsibilities, should be detailed by all the NEANIAS service providers to the users.

Until all the NEANIAS services will be mature enough, a more general SLA would be suitable to cover generic guarantees and conditions for all the NEANIAS services. Once some services would require peculiar aspects to be covered, then more specific SLAs should be defined.

This section presents guidelines, templates and examples to create the suggested NEANIAS General SLA and the more specific NEANIAS Service SLAs.

### 4.1. NEANIAS Corporate Level SLA

Corporate Level SLAs are generic documents that cover all SLAs and Services offered by a provider. They are suitable for less mature services or in situations where the end users do not have specific or variable service level demands. A good approach in introducing IT Service Management may be to start with a basic Corporate Level SLA, covering generic guarantees and conditions for all services, and replacing it with more specific SLAs later.

A FitSM Sample template for a Corporate SLA can be found at [9]. As a general rule, the SLA must be defined following the SMART principles [10] i.e. it needs to be *specific*, clearly stating the key points of the agreement; *measurable*, establishing metrics and KPIs for availability, maintenance, incident response, etc.; *achievable*, setting realistic goals which can be consistently met; *relevant*, focusing on aspects really critical to the service purpose; and *time-based*, clearly defining time-scales for tracking and measuring service performance and reporting incidents.

The Corporate Level SLA should include the following sections:

- A **Document Tracking** section. As SLAs are meant to be flexible and the conditions of a service may evolve with time, a proper versioning of the document is required. Table 1 Document Tracking Information shows the minimum tracking information.
- The **Contractual Parameters** of the SLA. This section establishes the policies for renewal, modification or termination of the agreement, as well as possible penalties in case of unfulfillment.
- An **Overview of the SLA**. This section should provide a general description of the agreement, clearly identifying the involved parties and the agreement duration. It should also declare that *the Corporate Level SLA will be valid for all NEANIAS IT services provided to support research and business processes, if no other agreements are in place*. This means that terms in the Corporate Level SLA may be modified, extended or replaced by NEANIAS Service-specific SLAs.
- The **Service Agreement details**. This section should include those aspects of the service delivery, operation and management that are subject to the agreement. At the very minimum, it should cover:
  - The Service availability and operating times. The services included in the NEANIAS service catalogue should be in general delivered during 24 hours per

## D7.3 Software Assessment Methodology

### D7.3 Software Assessment Methodology

day, 7 days per week (i.e. 365 days or 8,760 hours). Planned and announced interruptions may reduce the effective operating time of a service thus a planned management should minimise the impact on operations.

- Target KPIs for availability/uptime. For each service provided, the minimum annual availability target should be high (e.g. up to 99.9%), but this may depend on the criticality of the service. Planned and agreed interruptions (e.g., for maintenance) are not considered as unavailability, since they are not part of the effective operating time.
- Details of Planned interruptions, incident handling & support channels. For all NEANIAS services interruptions should be planned and be announced in advance using formal channels (e.g. at least 3 days earlier). There is no restriction on which days and/or day hours planned interruptions can take place. Critical security upgrades can be done with shorter notice. Support and incident handling should be available on business days (e.g. between 9:00 and 18:00 EEST on Mondays to Fridays) through NEANIAS helpdesk. Target resolution time in case of incidents should depend on the individual priority according to incident criticality time (e.g. up to 10 business days for low priority incidents, less than 5 business days in more urgent cases). Any incident reported through the channels should be acknowledged and replied upon a target reaction time (e.g. up to 10 business days for low priority incidents, less than 5 or 3 business days in urgent and very urgent cases respectively).

#### 4.2. NEANIAS Service SLA: Specific per Service

A service-specific Service Level Agreement may be defined between a NEANIAS service provider and the final service consumers, end users or groups of customers. Such SLA may be agreed for a single service, or cover multiple services with similar provision and support features. Specific SLAs will prevail over the Corporate Level SLA, and thus are expected to be more exhaustive and tailored to the service peculiarities. A FitSM Sample template for a specific service can be found at [11].

In general, the structure of a Service Level SLA may be the same as the Corporate Level SLA. It should include the following aspects:

- A General description of the SLA. As in the case of the Corporate Level SLA, this should detail the scope of the service, including the reference to the NEANIAS service catalogue. It should also include the details of the service provider and the final service consumers, end users or groups of customers.
- An enumeration of the (technical and logical) service components, including third party dependencies (if any) that could eventually disrupt the service operation.
- Service operation metrics and KPIs, agreed in advance, including the overall service availability, target uptime, expected maintenance windows (for upgrading/bug fixing), performance metrics (that will depend on the characteristics of the service), incident resolution times or any other specific request fulfilment times.
- Exceptions and limitations of the SLA. This should include any exceptions regarding the scope and application of the SLA, for instance, increased response times in certain periods or circumstances (e.g. holidays).

### D7.3 Software Assessment Methodology

#### D7.3 Software Assessment Methodology

- Any other technical limitations and constraints, including workload, storage and concurrency limits (if any).
- Enumeration of the available support channels, including details on the support contact points and their hours of operation. This part should cover incident response and quality assurance handling, according to an appropriate priority scale based on the impact and urgency of the incident. In addition to resolving incidents, further standard service requests may be defined to be fulfilled through the available support channels. Response, resolution times and fulfilment times are provided as service level targets. An example of a quality of support for incident handling (see [12]) may be as follows:

Incident priority	Base time	Response	Medium Response time	Advanced time	Response time
<b>less urgent</b>	5 working days		5 working days		5 working days
<b>urgent</b>	5 working days		5 working days		1 working day
<b>very urgent</b>	5 working days		1 working day		1 working day
<b>top priority</b>	5 working days		1 working day		4 working hours

*Table 2 Example of incident handling response time depending on quality of support*

- Communication, reporting and escalation procedures, to provide the contacts used for SLA-related communications with eventual reports regarding the fulfilment of the SLA and the provisioning of the service. The service provider commits to inform the end user in case the SLA is violated or a possible violation is anticipated. For escalation and complaints, the defined service provider contact point shall be used and specific rules should be provided.
- Information on security and data protection policies.
- Additional responsibilities of the service provider and the customer.
- A Review Policy of the service performance against service level targets and of the SLA, at planned intervals with the customer according to defined rules.

## 5. Quick checklist for assessment

The previous sections detailed many different aspects for the assessment of the software components as well as the service itself. The various software development and service operation recommendations are summarised through questions organised into categories to provide a checklist for the software developers and service owners.

In the following table we collected the most relevant i.e. key questions which links to topics detailed in the above sections. The questions are formalised in order to give some help for the software developer and service owner what are the field to focus on when assessment is to be performed.

The table below contains Yes/No and Notes/Refs columns. For each question, the answer can be Yes extended with any note or reference to an item which proves that the topic has been appropriately addressed. In case the answer is No, Notes/Refs column may contain the tasks that still need to be sorted out. Questions in the table with No answers require further attention until all answer becomes Yes.

Categories/key questions	Yes/No	Notes/Refs
<b>1. Development</b>		
<b>1.1 Software Development</b>		
Is the software designed and implemented using a SOA?		
Is a version control system used?		
Source code branch management defined and applied?		
VM/docker images available (if needed)?		
Is any static program analysis tool applied?		
<b>1.2 Testing</b>		
Is Continuous Integration methodology applied?		
Is Continuous Deployment methodology applied?		
Is unit testing applied?		
Is integration testing applied?		
Have the integration tests been made available in the NEANIAS CI pipelines?		
Are the Test Scenarios / Use Cases / Acceptance Criteria documented?		
Is acceptance testing applied?		
<b>1.3 Security</b>		
Are the security technologies identified?		

### D7.3 Software Assessment Methodology

#### D7.3 Software Assessment Methodology

Are the user's roles identified?		
Is the authentication bypass tested (e.g. brute force attack)?		
Are all data validated before processing?		
Is the password stored in a secure manner (e.g., using hash)?		
Is every expected error condition properly handled?		
<b>1.4 Data handling</b>		
Is data handling implemented according to FAIR principles?		
Is the service facilitating the discovery, usage, publication of relevant research products in the respective catalogue?		
<b>1.5 Integration to core services</b>		
Is AAI service integration implemented?		
Is Accounting Service integration implemented?		
Is Logging Service integration implemented?		
Integration to underpinning services target latest stable releases of such services?		
Is the service properly registered in the Service Catalogue?		
Is the service environment discovery taking place through respective services?		
<b>2. Operation</b>		
<b>2.1 Documentation, Ticketing, Helpdesk</b>		
Documentation available?		
Registered in ticketing system?		
Ticketing system used to plan and track intra/inter-service activities and assist the Plan-Do-Check-Act cycle?		
Is helpdesk provided?		
Is HelpDesk monitored and relevant KPIs tracked and reported on?		
All documents include necessary tracking information?		
All processes followed have been properly defined and documented?		
Service Management Scope & Plan document available for the Service?		
Service Management Policy plan available for the Service?		



### D7.3 Software Assessment Methodology

#### D7.3 Software Assessment Methodology

<b>2.2 Monitoring</b>		
Are one or more monitoring endpoints implemented?		
Are the monitoring endpoints accessible by the NEANIAS monitoring tools?		
Provided metrics include service availability/uptime, number of returning users, used vs. free server resources, total number of (HTTP) requests, server errors?		
Provided metrics are useful to measure the service target KPIs?		
Are there any alerts related to the service in the NEANIAS monitoring tools?		
The service can be correctly accessed and consumed?		
Are there any recent/unaddressed error messages in the logs?		
<b>2.3 Deployment</b>		
Is the service deployable in HA?		
If the service is deployed in HA, is the minimal subset of service components needed to keep the service fully and partially operational known?		
Is the service accessible by both IPv4 and IPv6?		
Is the service connected to a research network?		
A service backup has been planned, implemented and tested?		
<b>2.4 Security</b>		
Are all reported vulnerabilities under critical level?		
Is the password recovery process secured?		
Are transmissions secured by using some cryptography mechanisms?		
Is the log saved and rotated?		
<b>2.5 Licensing and Service Level Agreement</b>		
Is the license permits the use case supported by the service?		
Is the service covered by a Corporate Level SLA?		
Has a service-specific Service Level Agreement been defined for the service?		

**Table 3 Quick checklist to assess the key fields related to development and operation**

## 6. Conclusion

The assessment methodologies introduced in this deliverable help improving the quality of software and service components in the Neanias infrastructure. The high-level approach gives the possibility for each type of software and service to apply the recommendations on the different fields. To summarise the fields, key questions have been derived for each of them. With the aim of filling up the checklist with as many positive answers as possible, the software components can be improved. The ways to improve and assess them are introduced in the corresponding subsections.

## References

- [1] FitSM-0 Overview and vocabulary: <https://www.fitsm.eu/download/280/>
- [2] FitSM-1 Requirements: <https://www.fitsm.eu/download/295/>
- [3] FitSM Sample Service Management Policy: <https://www.fitsm.eu/download/336/>
- [4] The Quality Toolbox, Nancy R. Tague, ASQ Quality Press, Jan 1 2005
- [5] NEANIAS D7.1 Delivery activities methodology and plan
- [6] NEANIAS D7.2 Software delivery infrastructure and tools
- [7] FitSM Guide ITSM Documentation Checklist: <https://www.fitsm.eu/download/327/>
- [8] FitSM-6 Capability Maturity Assessment: <https://www.fitsm.eu/download/312/>
- [9] FitSM-4 Sample Corporate SLA: <https://www.fitsm.eu/download/333/>
- [10] S.M.A.R.T. principles: [https://en.wikipedia.org/wiki/SMART\\_criteria](https://en.wikipedia.org/wiki/SMART_criteria)
- [11] FitSM-4 Template SLA: <https://www.fitsm.eu/download/357/>
- [12] EGI Quality of Support levels: [https://wiki.egi.eu/wiki/FAQ\\_GGUS-QoS-Levels](https://wiki.egi.eu/wiki/FAQ_GGUS-QoS-Levels)

## D7.3 Software Assessment Methodology

### D7.3 Software Assessment Methodology

## List of acronyms

Acronym	Description
<b>SLA</b>	Service Level Agreement
<b>SMS</b>	Service Management System
<b>CI</b>	Continuous Integration
<b>CD</b>	Continuous Delivery
<b>GUI</b>	Graphical User Interface
<b>OWASP</b>	Open Web Application Security Project
<b>EOSC</b>	European Open Science Cloud
<b>AAI</b>	Authentication, Authorisation, Identification
<b>KPI</b>	Key Performance Index
<b>HA</b>	High Availability
<b>TLS</b>	Transport Layer Security
<b>SSL</b>	Secure Sockets Layer
<b>FOSS</b>	Free and Open Source Software
<b>PDCA</b>	Plan Do Check Act
<b>SLM</b>	Service Level Management
<b>OLA</b>	Operational Level Agreements
<b>UA</b>	Underpinning Agreement
<b>CRM</b>	Customer Relationship Management
<b>RDM</b>	Release and Deployment Management
<b>SACM</b>	Service Availability and Continuity Management
<b>ISM</b>	IT Security Management
<b>CIA</b>	Confidentiality, Integrity, Accessibility
<b>ISRM</b>	Incident and Service Request Management

## Appendix I – Service Management Policy

The following Service Management Policy sample is an extract of the FitSM Sample Service Management Policy [3]. It is provided as an example of the information and structured expected in such a policy document without necessarily aiming to restrict service providers. It is within the scope of the work of NEANIAS task T7.1 to further guide and assist service providers in compiling a fitting Service Management Policy document for their organisation.

# Service Management Policy

## 1. IT-Business alignment

The provision of IT services shall be aligned to customer and user needs.

- Services shall be delivered to a defined quality, sufficient to satisfy requirements identified from business processes.
- A clear service portfolio shall be developed and maintained as a basis for all service delivery and service management activities.
- For all services, a corporate level SLA and / or specific SLAs, which have been agreed with relevant stakeholders, shall be in place.

## 2. Process approach

To effectively manage all IT services and underlying components, a process-based approach to service management shall be adopted.

- All required processes shall be defined, communicated and improved based on business needs and feedback from people and parties involved.
- All roles and responsibilities for managing services (including roles as part of service management processes) shall be clearly defined.

## 3. Continual improvement

Services and service management processes shall be continually improved.

- Feedback from business stakeholders shall be used to continually improve services and service quality. All proposals for improvements shall be recorded and evaluated.
- Service management shall be improved based on continual monitoring of process performance and effectiveness.

## 4. Training & awareness

Through trainings and awareness measures, it shall be ensured that staff involved in service management activities can perform effectively according to their assigned roles.

## 5. Leadership

Top management is committed to this policy and its implementation. It provides the resources required to implement and improve service management and enhance customer satisfaction with IT services.

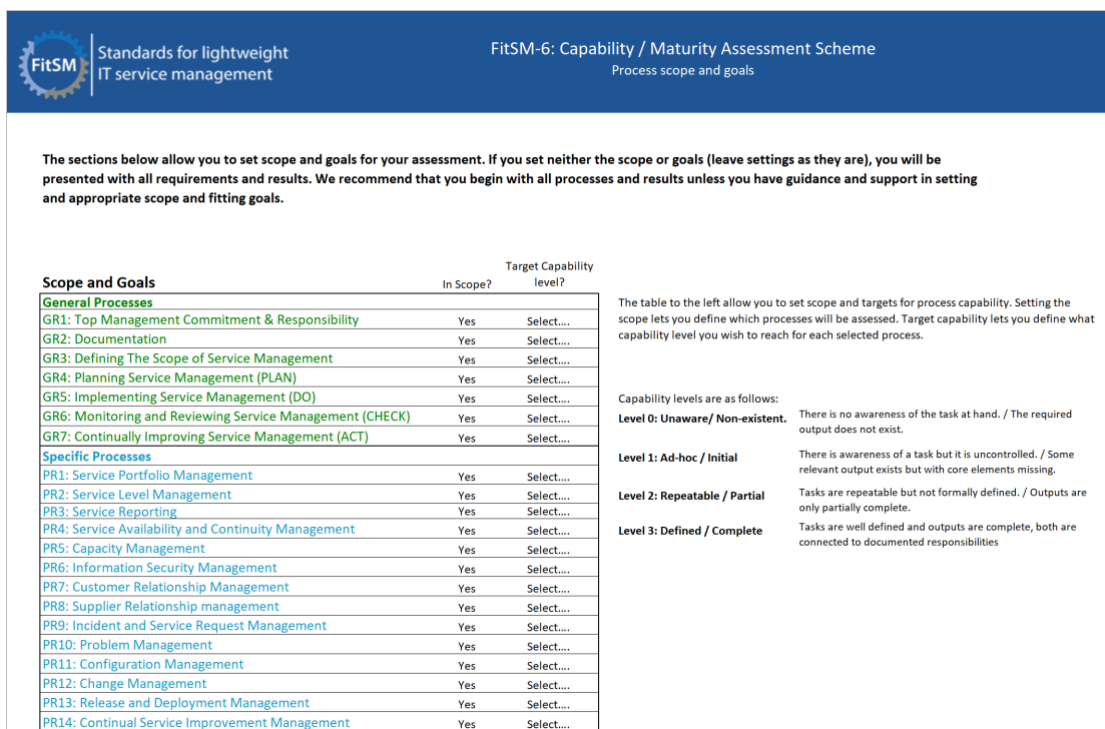
D7.3 Software Assessment Methodology

D7.3 Software Assessment Methodology

## Appendix II – SMS Capability / Maturity Assessment

The Capability Maturity Assessment tool [8] offered by FitSM-6 provides a capability/maturity assessment model to allow organisations and service providers to check and demonstrate their current capabilities in the FitSM processes and their overall IT service management (ITSM) maturity. This tool can be utilised as part of the assessment methodology within the context of the NEANIAS service assessment.

The available tool offers a wide range of topics, giving the ability to set target capability levels both at the level of General Processes, as well as Specific Processes. Individual targets can be set to different levels ranging from Non-Existent (Level 0) to Complete (Level 3).



The sections below allow you to set scope and goals for your assessment. If you set neither the scope or goals (leave settings as they are), you will be presented with all requirements and results. We recommend that you begin with all processes and results unless you have guidance and support in setting and appropriate scope and fitting goals.

Scope and Goals	In Scope?	Target Capability level?
<b>General Processes</b>		
GR1: Top Management Commitment & Responsibility	Yes	Select...
GR2: Documentation	Yes	Select...
GR3: Defining The Scope of Service Management	Yes	Select...
GR4: Planning Service Management (PLAN)	Yes	Select...
GR5: Implementing Service Management (DO)	Yes	Select...
GR6: Monitoring and Reviewing Service Management (CHECK)	Yes	Select...
GR7: Continually Improving Service Management (ACT)	Yes	Select...
<b>Specific Processes</b>		
PR1: Service Portfolio Management	Yes	Select...
PR2: Service Level Management	Yes	Select...
PR3: Service Reporting	Yes	Select...
PR4: Service Availability and Continuity Management	Yes	Select...
PR5: Capacity Management	Yes	Select...
PR6: Information Security Management	Yes	Select...
PR7: Customer Relationship Management	Yes	Select...
PR8: Supplier Relationship management	Yes	Select...
PR9: Incident and Service Request Management	Yes	Select...
PR10: Problem Management	Yes	Select...
PR11: Configuration Management	Yes	Select...
PR12: Change Management	Yes	Select...
PR13: Release and Deployment Management	Yes	Select...
PR14: Continual Service Improvement Management	Yes	Select...

The table to the left allow you to set scope and targets for process capability. Setting the scope lets you define which processes will be assessed. Target capability lets you define what capability level you wish to reach for each selected process.

Capability levels are as follows:

- Level 0: Unaware / Non-existent.** There is no awareness of the task at hand. / The required output does not exist.
- Level 1: Ad-hoc / Initial** There is awareness of a task but it is uncontrolled. / Some relevant output exists but with core elements missing.
- Level 2: Repeatable / Partial** Tasks are repeatable but not formally defined. / Outputs are only partially complete.
- Level 3: Defined / Complete** Tasks are well defined and outputs are complete, both are connected to documented responsibilities

Figure 2 - SMS Capability / Maturity Assessment scope & goals

For each general and specific process, a set of tasks, activities and achievements are identified, and assessors can evaluate the level of maturity for each requirement.

D7.3 Software Assessment Methodology

D7.3 Software Assessment Methodology

FitSM Standards for lightweight IT service management		FitSM-6: Capability / Maturity Assessment Scheme			Process requirement assessment			Note: If questions are in pale grey this is because they are set as not in scope in the "Process scope and goals" tab	
Topic area	Type of requirement	Requirement code	Requirement according to FITSM-3: Edition 2016 – Version 2.1	Capability Level	Descriptions	Self assessment score	Rationale for score	Evidence (e.g. available documents / records / URIs)	
<b>General Processes</b>									
GR1: Top Management Commitment & Responsibility	Task / activity	GR1.1	Top management of the organisation(s) involved in the delivery of services shall show evidence that they are committed to planning, implementing, operating, monitoring, reviewing, and improving the service management system (SMS) and services. They shall: - Assign one individual to be accountable for the overall SMS with sufficient authority to exercise this role - Define and communicate goals - Define a general service management policy - Conduct management reviews at planned intervals.	1- Ad hoc	Top management deals with service management tasks primarily on a reactive basis. They are generally aware of their responsibilities to communicate goals and policies as well as monitoring and reviewing the effectiveness of the SMS. Tasks related to this are performed to the best of knowledge in the individual situation, and do not follow a formal and/or easily reproducible approach.	Select....			
				2- Repeatable	Approval and review of the general service management policy by top management happens at regular intervals and with a clear understanding of the related tasks. Goals and policies are effectively communicated with communication mechanisms and channels being used in a consistent manner. At regular intervals, top management reviews the effectiveness of the service management system and records the key results and follow-up actions.				
				3- Defined	Top management's responsibilities in the service management context are clearly defined and documented, and in particular, the role of a senior responsible owner has been defined and assigned to a top management representative. Approval and review of the service management policy is performed in a formal way, and to ensure effective communication of goals and policies, communication plans are created that clearly indicate what to communicate, how, when, to whom and by whom. Formal management reviews of the overall service management system are conducted at well-planned intervals.				
GR2: Documentation	Output / achievement	GR2.1	The service management policy shall include: - A commitment to fulfill customer service requirements - A commitment to a service-oriented approach - A commitment to a process approach - A commitment to continual improvement - Overall service management goals	1- Initial	An overall service management policy has been documented, but lacks clear service management goals as well as a clear commitment to all core principles of service management.	Select....			
				2- Partial	An overall service management policy has been documented, which covers clear service management goals and a commitment to some key service management principles.				
				3- Complete	An overall service management policy has been documented, which covers all required elements including clear service management goals and a commitment to fulfilling customer service requirements, following a service- and process-oriented approach, as well as applying the principle of continual improvement.				
GR2: Documentation	Output / achievement	GR2.1	The overall SMS shall be documented to support effective planning. This documentation shall include:	1- Initial	Some initial service management-related documentation is available, while key documents, like the service management scope statement, an overall service management policy or a service management plan				

Figure 3 - SMS Capability / Maturity Process Assessment

Based on the targets set and respective assessment, the process capability results can be extracted and used for the evaluation of the IF Service Management System employed.

FitSM Standards for lightweight IT service management		FitSM-6: Capability / Maturity Assessment Scheme			Process capability results				
FitSM Processes	Scope and goals		Requirements		Capability assessment				Targets
	In Scope	Capability goal	Requirement code	Level 0	Level 1	Level 2	Level 3		
GR1: Top Management Commitment & Responsibility	Yes	N/A	GR1.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR1.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
GR2: Documentation	Yes	N/A	GR2.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR2.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR2.3	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR2.4	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR2.5	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
GR3: Defining The Scope of Service Management	Yes	N/A	GR3.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR4: Planning Service Management (PLAN)	Yes	N/A	GR4.1	Not Answered	Not Answered	Not Answered
GR4.2	Not Answered	Not Answered	Not Answered			Not Answered	No target set		
GR4.3	Not Answered	Not Answered	Not Answered			Not Answered	No target set		
GR5: Implementing Service Management (DO)	Yes	N/A	GR5.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR5.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
GR6: Monitoring and Reviewing Service Management (CHECK)	Yes	N/A	GR6.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR6.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
GR7: Continually Improving Service Management (ACT)	Yes	N/A	GR7.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			GR7.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
PR1: Service Portfolio Management	Yes	N/A	PR1.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR1.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR1.3	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR1.4	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
PR2: Service Level Management	Yes	N/A	PR2.1	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.2	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.3	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.4	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.5	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.6	Not Answered	Not Answered	Not Answered	Not Answered	No target set	
			PR2.7	Not Answered	Not Answered	Not Answered	Not Answered	No target set	

Figure 4 - SMS Capability / Maturity Assessment Results

It is within the scope of the work of NEANIAS task T7.1 to further guide and assist service providers in utilising this tool to perform early assessment of employed service management processes as well as to produce a respective tool that will be more tailored to the NEANIAS implemented processes.