



pharaon

PILOTS FOR HEALTHY AND ACTIVE AGEING

Grant Agreement: 857188

D4.3

Developer guidelines and templates - Final



This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 857188.



Document Information

Deliverable number:	D4.3
Deliverable title:	Developer guidelines and templates - Final
Deliverable version:	1.0
Work Package number:	WP4
Work Package title:	Technology Support Tools
Due Date of delivery:	M54 (May 2024)
Actual Date of delivery:	M54 (May 2024)
Dissemination level:	Public (PU)
Type	Other (O)
Editor(s):	Andrej Grguric (ENT), Miran Mosmondor (ENT)
Contributor(s):	Miran Mosmondor (ENT) Andrej Grguric (ENT) Carlos Fernández Sánchez (INDRA) Stefania D'Agostini (ENG)
Reviewer(s):	Ângela Rodrigues (CDC), Elisabete Pitarma (CDC)
Project name:	Pilots for Healthy and Active Ageing
Project Acronym	PHArA-ON
Project starting date:	01/12/2019
Project duration:	60 months
Rights:	PHArA-ON Consortium

Document history

Version	Date	Beneficiary	Description
0.1	14.02.2024	ENT	Table of Contents proposal and initial content considerations for each chapter
0.2	25.03.2024	ENT	Initial content and figures added to Introduction (Section 1.2), <i>Developers' Handbook</i> Structure and content (Section 4.2) and Progress timeline section (Section 4.3).
0.3	26.04.2024	ENT	Completed section on Internal review process (Section 4.4.)
0.4	06.05.2024	ENT	Completed <i>Developers' Handbook</i> Structure and content (Section 4.2), initial content to Software documentation tools and technology (Section 4.1)
0.5	13.05.2024	ENT	Completed <i>Developers' Handbook</i> Software documentation tools and technology (Section 4.1), updated and finalized Chapter 4.
0.6	17.05.2024	ENT	Added Appendix A: Technology Guidelines reference list
0.7	20.05.2024	ENT	Updates to Section 2 Pharaon DevSecOps phases and work organization, and Section 3 3 Pharaon Technical One-Stop-Shop.
0.8	21-05-2024	INDRA	Add description of Docusaurus tool in section 4.1.2
0.9	22-05-2024	ENT	Updated Summary and Progress beyond state-of-the-art section
0.10	23-05-2024	ENT	Finalized Conclusion, Structure, and minor improvements to Section 3.
0.11	24-05-2024	ENG	Updated Section 2.2
0.12	27-05-2024	CDC	Review and revision of D4.3.
0.13	28-05-2024	ENT	Minor editor updates.
1.0	29-05-2024	ENT	Deliverable final updates and wrap up

PM efforts per beneficiary having contributed to the deliverable document¹

#	Partner	PM effort on D4.3 deliverable report
15	INDRA	0.01
18	CDC	0.10
20	MAIN	0.05
27	ASC	0.01
28	ENT	1.00
45	ENG	0.10
TOTAL	Pharaon Consortium	1.27

Acknowledgement: This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 857188.

Disclaimer: The content of this publication is the sole responsibility of the authors, and in no way represents the view of the European Commission or its services.

¹ Efforts for contributing to the preparation of document that describes the work done

Executive Summary

This document reports on the work carried out within the task T4.1 “Developer guidelines & templates” of the fourth work package (WP4) of the PHArA-ON (in further text “Pharaon”) project.

This deliverable is the third and final one within task T4.1, aiming to provide support for technical development activities in the Pharaon project. The main outcome of the task are the guidelines for developers and technical users and the realization of Pharaon *Technical One-Stop-Shop*. The *Technical One-Stop-Shop* serves as the main entry point through which developers (both internal and external recruited through the open calls in WP6) can access all the documentation, guidelines, different repositories, recommended open-source tools, best practices and the most important technical information needed to work on the Pharaon technical aspects. The work continued on finalization of the established Pharaon *Developers’ Handbook* (realized using Gitlab Wiki as primary entry point and available to registered developers through private Pharaon Gitlab at: <https://gitlab.com/pharaongroup/developers-handbook>), to provide all the technical guidelines and serve many different technical activities of the Pharaon partners who are adapting, integrating, testing, and validating their technologies within the Pharaon and Open Call (OC) pilots.

Previous version of this deliverable focused on establishing (1) Pharaon *Technical One-Stop-Shop*, overall Pharaon SDLC methodology and processes, recommended open-source tools and best practices and initial guidelines needed at the beginning of the project (documented in D4.1), and (2) on the Pilot Entry Pages that have been added as a top-level entry and all parts of the *Technical One-Stop-Shop* that have been extended with new content following the needs of the Pharaon on Open Call 1 technical work (documented in D4.2).

The focus of this deliverable, D4.3, is on the completion of all remaining guidelines about Pharaon technology, especially the guidelines about technology that have not yet been finalized by November 2022. This is primarily related to dozens of newly created technology guidelines about Ecosystem Hub and how to integrate technical components in order to comply with Pharaon integration standards. Several new guidelines have been created related to other Pharaon technologies, best practices, and how-to guidelines.

The work on this deliverable also focused on creating a subset of *Developers’ Handbook* for participants of Open Call 2 (WP6). Within the work on this deliverable, all technology guidelines content has been aligned and improved through the established improved internal review process that was crucial for creating effective and high-quality guidelines.

Progress beyond the state of the art and play

The work and activities reported within this task and deliverables were crucial in assisting Pharaon developers to understand how to use various Pharaon platforms and technologies. This provided them and other technical stakeholders with information about the technical aspects of Pharaon overall development, including adaptations, integration, deployment, and maintenance, and helped ensuring that the development process was consistent and repeatable. The task took a holistic approach with respect to overall technical management and developments and mapped established industry common practices and newest software engineering trends to Pharaon to facilitate overall knowledge sharing and adoption of best practices. The task deal with structuring the relevant information to make it easily accessible and especially useful for the day-to-day hands-on work of Pharaon technical personnel, which was especially challenging in the Pharaon environment where technologies are hosted in a multi-cloud hybrid environment setting. The key was to perform improvements continuously during the development of *Developers' Handbook* guidelines and other aspects of described *Technical One-Stop-Shop* and involved multiple stages with the main objective of enhancing the accuracy and quality of the content for developers.

Contents

Executive Summary	5
Progress beyond the state of the art and play	6
Acronyms & Abbreviations.....	9
1 Introduction.....	10
1.1 Overview	10
1.2 Relation to other tasks and deliverables	10
1.3 Structure of the deliverable	12
2 Pharaon DevSecOps phases and work organization	13
2.1 Pharaon technical support tools inputs.....	13
2.2 Pharaon DevSecOps phases.....	13
2.3 Pharaon technical work organization	14
3 Pharaon <i>Technical One-Stop-Shop</i>	17
3.1 High-level overview.....	17
3.2 <i>Developers' Handbook</i>	18
3.3 Code repositories and registries	18
3.4 Tools.....	18
3.5 Pilot entry pages	22
4 <i>Developers' Handbook</i>	26
4.1 Software documentation tools and technology	26
4.1.1 Gitlab Wiki	27
4.1.2 Docusaurus	28
4.2 Structure and content.....	28
4.3 Timeline.....	31
4.4 Guidelines improvement process	33
4.4.1 Overall iterative improvement process.....	34
4.4.2 Technology guidelines internal review process.....	35
4.4.3 The results	38
5 Conclusion	39
Appendix A: Technology Guidelines reference list.....	40

Figures

Figure 1.1 Overview of main Pharaon technical activities and WP4 tasks	11
Figure 1.2 Inputs and outputs of D4.3 concerning other project work	11
Figure 2.1 Pharaon technical support tools (WP4) inputs sources	13
Figure 2.2 Selection of tools mapped to Pharaon DevSecOps lifecycle phases (from D4.1)	14
Figure 2.3 Short overview of technical work organization in Pharaon	15
Figure 2.4 WP4 in support of technical work in Pharaon.....	15
Figure 2.5 Comparison of the Pharaon sandboxes mapped to Pharaon work packages (from D4.11)	16
Figure 3.1 Updated Pharaon <i>Technical One-Stop-Shop</i> structure	17
Figure 3.2 Final high-level structure of <i>Developers' Handbook</i>	18
Figure 3.3 Tools overview (from D4.2)	19
Figure 3.4 Pharaon external developers onboarding process (based on input from D4.2).....	22
Figure 4.1 Key features of software documentation tools	27
Figure 4.2 Final structure of <i>Developers' Handbook</i>	30
Figure 4.3 <i>Developers' Handbook</i> timeline overview.....	31
Figure 4.4 <i>Developers' Handbook</i> structure in intermediate version from D4.2	32
Figure 4.5 Structure of <i>Developers' Handbook</i> subset for OC2	33
Figure 4.6 Overall iterative guidelines improvement process	35
Figure 4.7 Technology guidelines internal review process	36
Figure 4.8 The structure of internal review checklist and evaluation form	37
Figure 4.9. Snapshot of internal review checklist results in first iteration.....	38

Tables

Table 3.1 Template for tools survey (from D4.2)	19
Table 3.2 Pharaon tools outcomes mapped to DevSecOps phases (from D4.2)	20
Table 3.3 Pilot entry pages welcome screen content (from D4.2).....	24
Table 4 Technology Guidelines reference list table	40

Acronyms & Abbreviations

Term	Description
CD	Continuous Delivery
CI	Continuous Integration
DevSecOps	Development Security and Operations
DoW	Description of Work
OC1, OC2	Open Call 1, Open Call 2
QA	Quality Assurance
SDLC	Software Development Life Cycle
TM	Technical Manager
UAT	User Acceptance Testing
WP	Work Package
WP3	Pharaon WP3 Secure Interoperability Solution
WP4	Pharaon WP4 Technology Support Tools
WP5	Pharaon WP5 Technology Ecosystem Integration
WP6	Pharaon WP6 Ecosystem Evolution
WP7	Pharaon WP7 Pilot Deployment, User Validation, Optimisation and Evaluation

1 Introduction

1.1 Overview

This deliverable is the third and final one within task T4.1 “Developer guidelines & templates” aiming to provide support for technical development activities in the Pharaon project, primarily conducted in WP3, WP4, and WP5, and technical maintenance and support activities within WP7.

1.2 Relation to other tasks and deliverables

One of the main objectives of technical activities in the Pharaon project is to create interoperable open platforms, tools and technologies that are used by end users and external developers. Pharaon developers develop (plan, design, code, build, test, release) these solutions and include two main activities: firstly, adapting input technologies and platform to support user functional requirements defined in WP2, along with addressing security and interoperability requirements as part of WP3 activities. Secondly, performing technical ecosystem integration, testing and technical validation carried out within WP5 work.

Pharaon technical personnel are Pharaon developers, or any other technical personnel who operate developed solutions – deploy in production environment, maintain, upgrade, provide technical support, all within Pharaon pilots WP7 technical activities. These deployed solutions are used by service providers who provide these integrated solutions for end users. Also, the developed interoperable platforms, tools and technologies are also used and integrated by external developers in their OC1 and OC2 solutions.

WP4 delivers technology supporting tools to Pharaon developers during development in WP3 and WP5, Pharaon technical person for deployment and operations phase in WP7, and external developers. WP4 activities are not directly related to developing and providing services to end users, but technologies, tools, and other assets to help developers create and operate these interoperable platforms, tools, and technologies. The overview is illustrated with Figure 1.1.

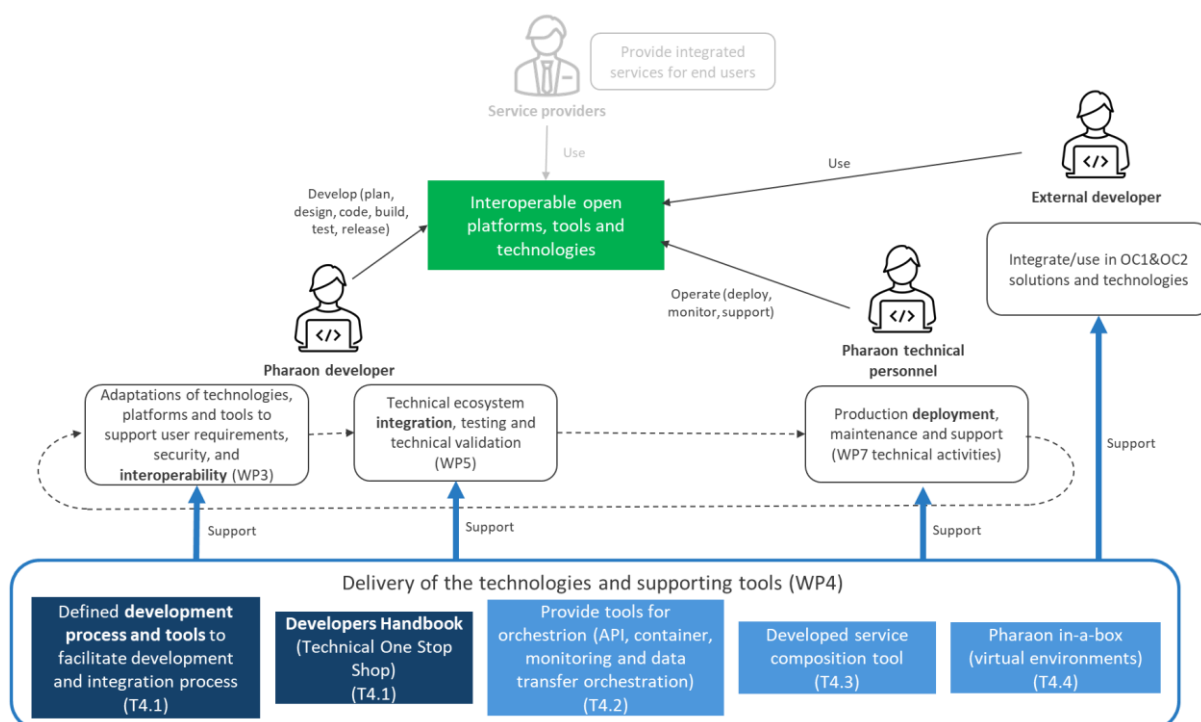


Figure 1.1 Overview of main Pharaon technical activities and WP4 tasks

The main objectives of this task are related to the definition of the development process and tools to facilitate the development and integration process, and the realization of the *Developers' Handbook*. The main inputs to this deliverable come mainly from WP3, WP5, and WP6 activities but other project activities related to the technical aspects also influenced this work. The specific inputs and outputs are shown in the following figure (Figure 1.2).

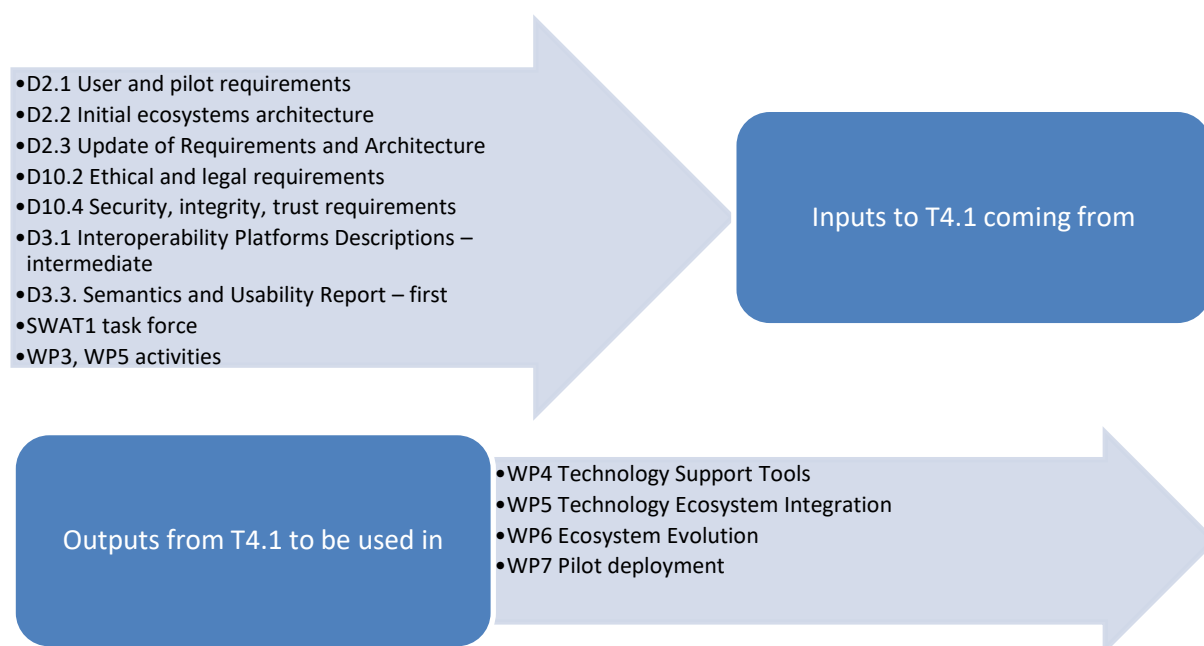


Figure 1.2 Inputs and outputs of D4.3 concerning other project work

1.3 Structure of the deliverable

This document describes the final version of guidelines for developers, and it is organized in the following sections:

- After the initial chapters, namely the executive summary and the list of acronyms used in the document, an Introduction to the document is given, including how it relates to other project tasks and deliverables.
- Chapter 2 briefly summarizes Pharaon DevSecOps phases and provides a final overview of the Pharaon technical work organization.
- Chapter 3 describes all the T4.1 work produced under the Pharaon Technical One-Stop-Shop umbrella term, work that was mostly completed in previous version of the deliverables but linked here to provide a reference location for all T4.1 activities in a single final document.
- Chapter 4 describes in detail the work carried out within this deliverable - the final content and structure of the *Developers' Handbook*, the documentation tools used, the evolution of the guidelines over time and the guidelines improvement process that was carried out to deliver high-quality guidelines.
- Chapter 5 concludes the report.
- Appendix A lists all the technology guidelines created and finalized within the work on this version of the deliverable.

2 Pharaon DevSecOps phases and work organization

2.1 Pharaon technical support tools inputs

As described in the previous version of the deliverable, the WP4 acts to support the technical work in Pharaon thus the technical support tools inputs were indirectly derived from different sources. The input sources for the Pharaon technical support tools (WP4) are summarized at a high level in the following figure (Figure 2.1).

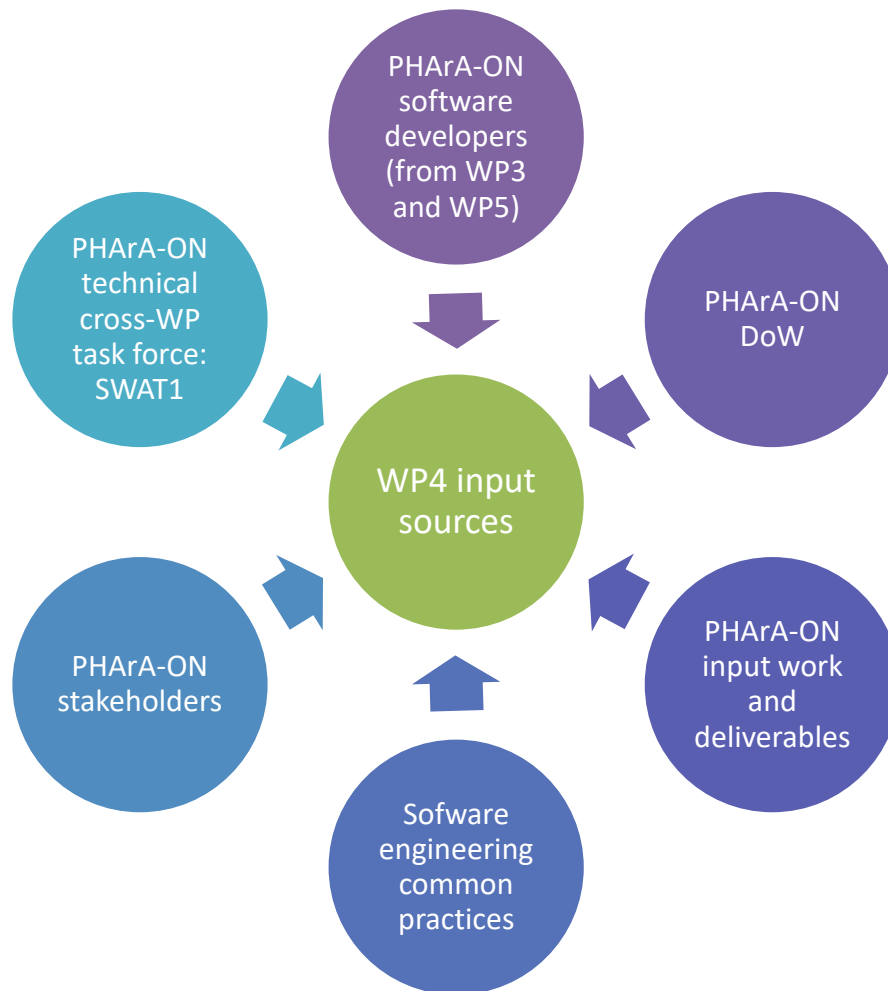


Figure 2.1 Pharaon technical support tools (WP4) inputs sources

2.2 Pharaon DevSecOps phases

The initial work on the overall Pharaon SDLC methodology and processes was carried out in WP4 and described in the initial version of this deliverable (D4.1). The following figure (Figure 2.2) has been presented to give a brief overview of the tools already in use or under close consideration to be used in Pharaon regarding the different phases of the Software Development Life Cycle (SDLC).

Additional details are provided with the work on WP5. As described in D5.1, within the Pharaon project we adopted a SDLC based on the DevSecOps principles and guidelines. In particular, the *Plan*, *Design* and *Code* phases of the SDLC process were managed in WP3 and WP4, while WP5 took care of the

Build, Test, Release, and Deploy (on a staging environment) activities before the deployment of the pilots in production (WP7).

A Continuous Integration and Continuous Delivery (CI/CD) process and infrastructure, along with a Quality Assurance (QA) approach based on multi-level testing, were defined (see D5.1) and adopted to guide integration, releases and testing of the various Pharaon components developed in WP3 and WP4. The results of the adoption of this CI/CD and testing approach are reported in the initial, intermediate, and final Pharaon releases described in D5.1, D5.2 and D5.3, respectively.

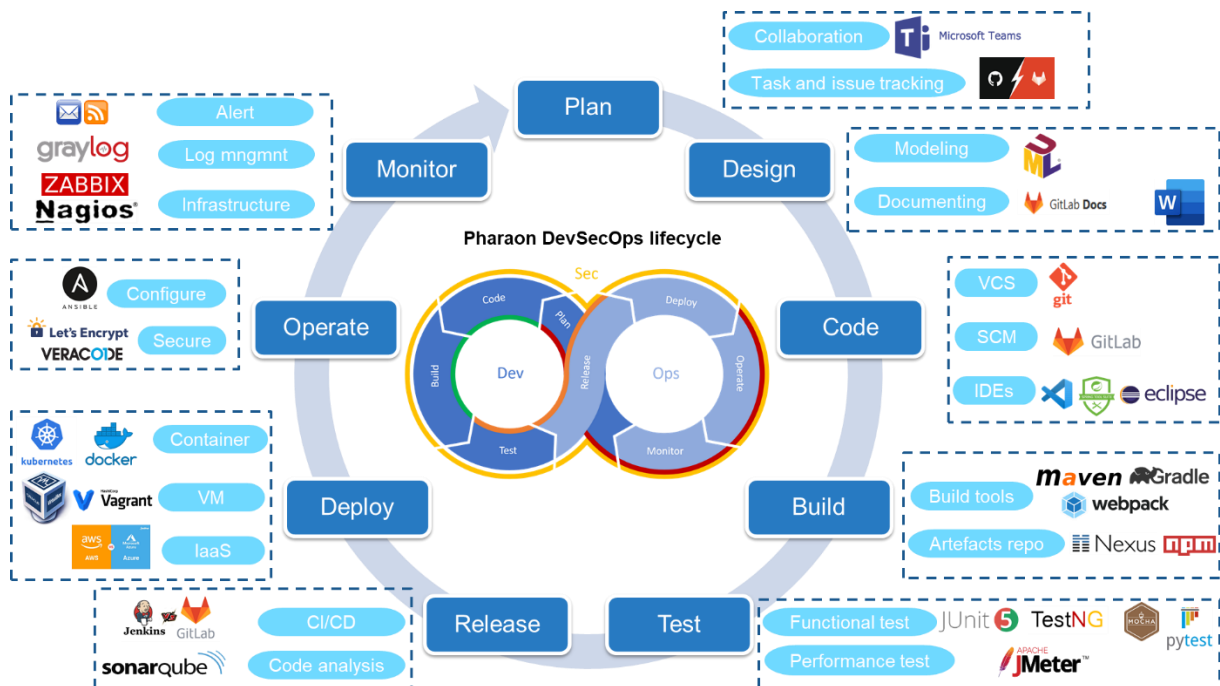


Figure 2.2 Selection of tools mapped to Pharaon DevSecOps lifecycle phases (from D4.1)

2.3 Pharaon technical work organization

An overview of Pharaon technical work organization is illustrated in the following figure (Figure 2.3)

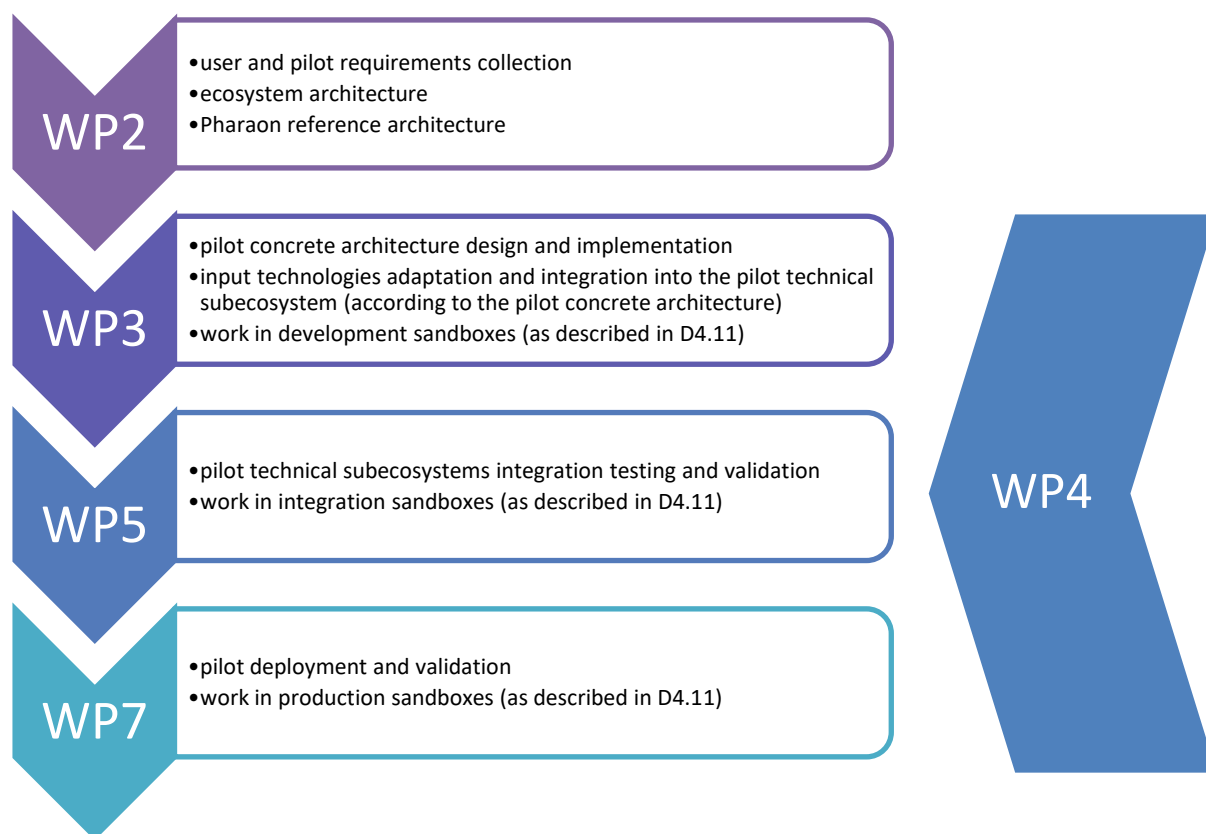


Figure 2.3 Short overview of technical work organization in Pharaon

The following figure (Figure 2.4) illustrates the WP4 support activities in support of the Pharaon technical work (conducted primarily in WP3 and WP5).

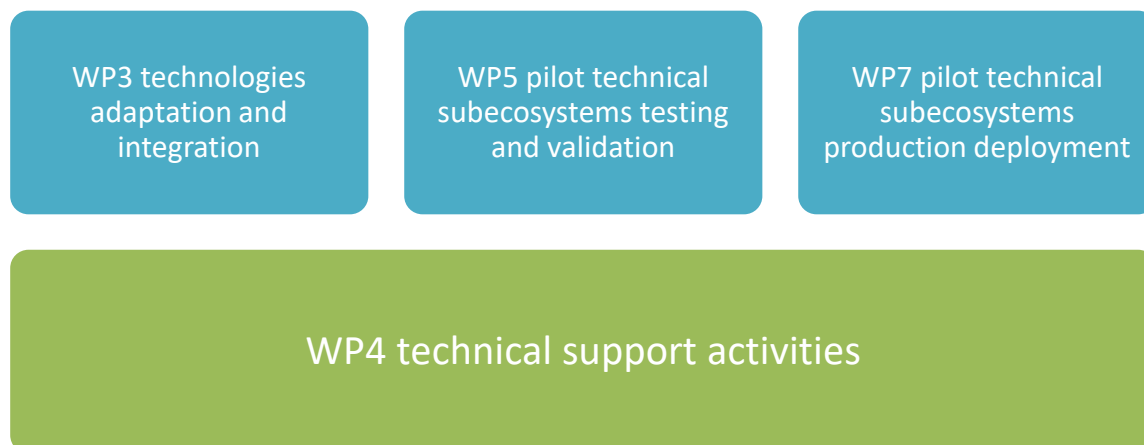


Figure 2.4 WP4 in support of technical work in Pharaon

Mapping of Pharaon sandboxes (together with their characteristics) to Pharaon work packages (i.e. WPs) is given in the following figure (Figure 2.5). More information on Pharaon sandboxes can be found in the initial, intermediate, and final versions of Pharaon in-a-box, described in D4.10, D4.11 and D4.12 (to be released in June 2024), respectively.

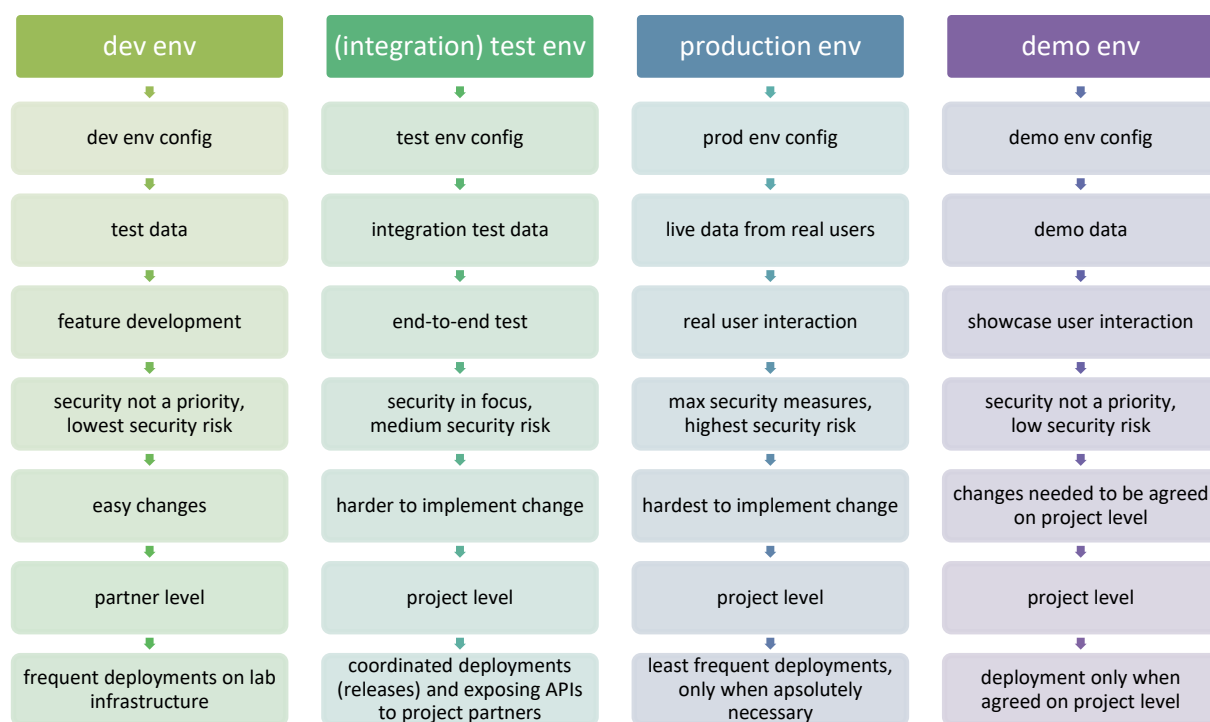


Figure 2.5 Comparison of the Pharaon sandboxes mapped to Pharaon work packages (from D4.11)

3 Pharaon *Technical One-Stop-Shop*

3.1 High-level overview

As described in previous versions of this deliverable, the main outcome of WP4 technical support activities within Pharaon is the realization of the Pharaon *Technical One-Stop-Shop*, as it is considered as the main entry point through which Pharaon personnel, mainly technical people, can access the documentation, guidelines, different repositories, and the most important technical information needed to work on the Pharaon technical ecosystem.

Pharaon *Technical One-Stop-Shop* is hosted on GitLab as a private repository, available to authorized personnel at the following web address: <https://gitlab.com/pharaongroup>. One major change since the version described in D4.2 is Gitlab's decision to offer private Gitlab groups no longer for free. Thus, after analysis in the beginning of 2023 (considering all options and requirements) the Pharaon Core Board (CB) decided to switch to Gitlab Premium SaaS Plan (from August 2023). This change has influenced how Pharaon Gitlab users are managed: from August 2023, all new members need to be approved by Pharaon Technical Manager. Further user and access management is carried out done by the Technical Manager (CETEM) and WP4 leaders (ENT) with granted access to Pharaon *Technical One-Stop-Shop*, which currently includes 75 different Pharaon repositories (<https://gitlab.com/dashboard/projects>).

Updated structure of the *Technical One-Stop-Shop* is illustrated in the following figure (Figure 3.1). It is an evolution of the initial version, presented in the D4.1, and intermediate version, presented in D4.2, where some aspects have already been explained in more detail. Concerning the previous version of the high-level structure of the *Technical One-Stop-Shop* (described in the previous D4.2), there have been no significant changes, however, structure of *Developers' Handbook* was considerably changed as many new Pharaon Technology guides have been created and added (in detailed explained in Section 4). Other parts of the *Technical One-Stop-Shop* have received minor updates.



Figure 3.1 Updated Pharaon *Technical One-Stop-Shop* structure

3.2 Developers' Handbook

As mentioned, *Developers' Handbook* is hosted on GitLab Wiki (<https://gitlab.com/pharaongroup/developers-handbook/-/wikis/home>). The final high-level structure of the *Developers' Handbook* is summarized in the Figure 3.2. The most significant change is addition of Pharaon Technology guides section with three main parts:

- Ecosystem Hub technologies
- Pharaon platforms
- Other Pharaon technologies

Since the *Developers' Handbook* and technology guidelines were the main focus of the work in the last period, more details are provided in Section 4.

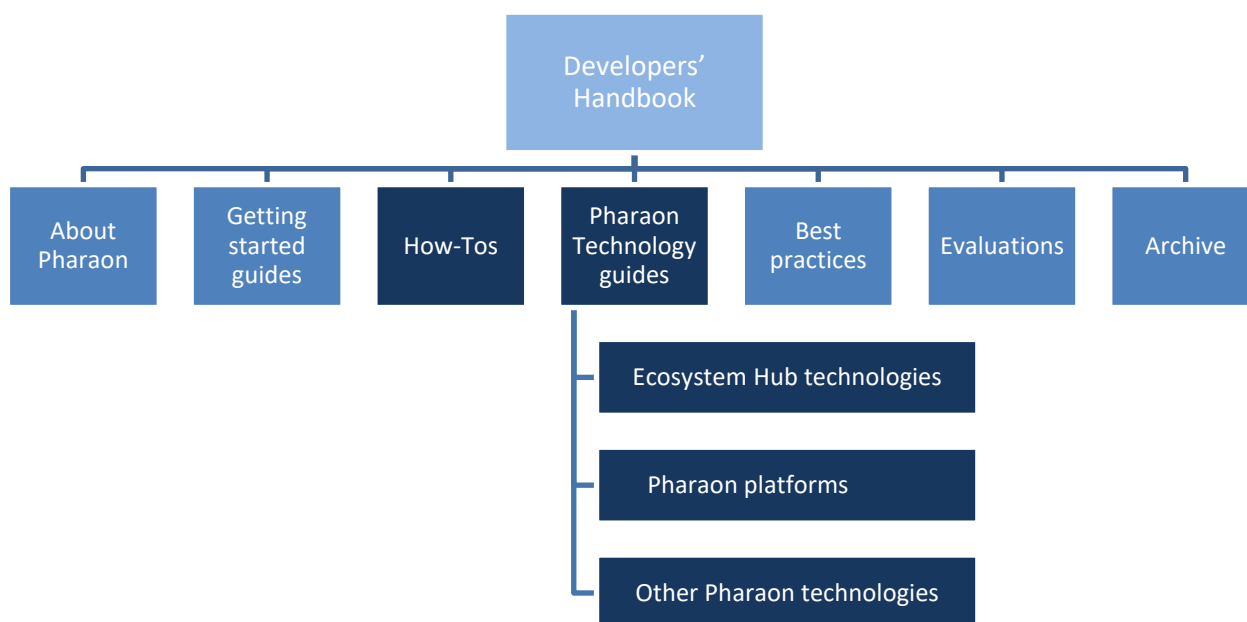


Figure 3.2 Final high-level structure of *Developers' Handbook*

3.3 Code repositories and registries

Hosted on <https://gitlab.com/pharaongroup> in May 2024 there were 75 private Pharaon repositories which contain outputs from WP3, WP4, WP5 and WP7 and were used by developers and technical personnel working on related tasks. Some repositories were also made accessible to participants of Open Call 1.

3.4 Tools

Hosted on <https://gitlab.com/pharaongroup/developers-handbook/-/wikis/home#tools> current tools categories are summarized in the following figure (Figure 3.3).

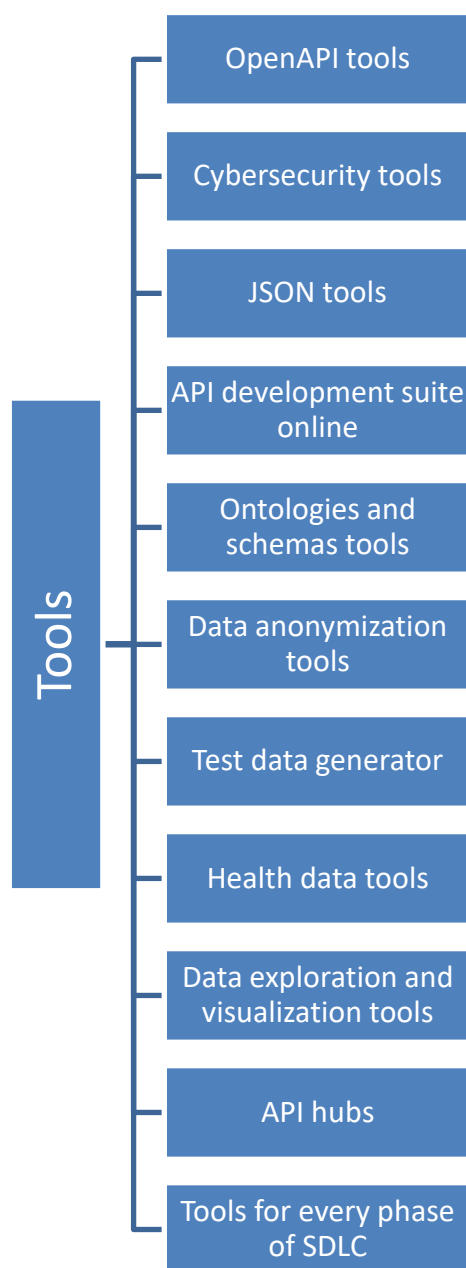


Figure 3.3 Tools overview (from D4.2)

All work regarding the tool survey was described in previous deliverable (D4.2) and focused on free and open-source tools to support different needs of the Pharaon technical work, especially considering the different environments in which the Pharaon technical providers operate.

The survey framework, as shown in the Table 3.1, was used as a template for the tools survey (uploaded in Pharaon GitLab) containing more than 160 tools.

Table 3.1 Template for tools survey (from D4.2)

SDLC phase	Tool category	Tool name	Short Description	License/ Price	Link	Selected for use on the Pharaon project	Comment
------------	---------------	-----------	-------------------	----------------	------	---	---------

Table 3.2 presented the mapping of the Pharaon DevSecOps phase to related WPs, outcomes, and tool categories. This table is given also as a response to one of the fruitful discussions with project reviewers, providing a clearer connection between the Pharaon SDLC and DevSecOps phases and the work done in the WP4 tools survey to support different Pharaon work packages and technical activities.

Table 3.2 Pharaon tools outcomes mapped to DevSecOps phases (from D4.2)

SDLC and DevSecOps phase	Short Description	Related WP	Pharaon outcome	Pharaon tool category
Plan	Involves requirements and feedback gathering from stakeholders and users. It is the first phase of the DevSecOps lifecycle that involves understanding the scope and purpose of the project.	WP1, WP2	Documentation, Wiki	Communication (MS Teams, Zoom) Task and issue tracking (GitLab issues, Excel sheets)
Design	The purpose of the Design Definition process is to provide sufficient detailed data and information about the system and its elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. [ISO/IEC/IEEE 12207:2017]	WP2, WP3	Architecture, program languages, UIs, platforms, Samples, tutorials, discussion forum, email topic-specific threads	Documentation (MS Word, Visio, MS Power Point, UML..)
Code	Related to developing the source code for the system	WP3, WP4	Code standards per programming language, code documentation, code generation, version management Code samples; Pharaon adapters	VSC (Version Control System) SCM (Source Code Management) IDEs (Integrated Development Environments) Source code editors
Build	Related to an automated process which builds the codebase and runs a series of end-to-end, integration and unit tests to identify any regressions.	WP3, WP4	Build management, Continuous Integration	Build automation tool Automate dependency updates Package (Binaries) repository

				General purpose repository
Test	Manual (i.e. UAT) or automated tests. Automated tests might run security scanning against the application, check for changes to the infrastructure and compliance with hardening best-practices, test the performance of the application or run load testing.	WP5	Debug management, Test management, unit test, white box, black box testing, integration test, stress test	Code test frameworks Code quality tools Data generators Load test tools API testing tools Security testing tools Penetration testing
Release	The point at which developers can say a build is ready for deployment into the production environment. By this stage, each code change has passed a series of manual and automated tests, and the operations team can be confident that breaking issues and regressions are unlikely. The organization then manually or automatically deploys any build that makes it to this stage of the pipeline.	WP5	Release management	CI/CD tools
Deploy	In this phase, a build is ready and released into production. Several tools and processes can automate the release process to make releases reliable with no outage window.	WP7	Config management, installation, instantiation, provisioning; device management, service management	Containers Web application servers Configuration management tools
Operate	In the operation phase, the deployed system in the production environment is being managed and making sure that everything is running smoothly. This also includes receiving feedback from users	WP7	Update management; data analysis	Secret Management Service Discovery IAM (Identity and Access Management tools) Antivirus Network analyzers

				Network defense Password security auditing and recovery Web vulnerabilities scanning Network Intrusion detection and prevention
Monitor	The 'final' phase is to monitor the environment by collecting data and providing analytics on user behavior, performance, errors and more.	WP7	Log management, notifications, alerting, 1 st , 2 nd , 3 rd lines of support	Logging tools IT Infrastructure Monitoring

3.5 Pilot entry pages

The work on pilot entry pages was completed and described in the previous deliverable (D4.2) with only minor updates in the final version. As described in D4.2, Pharaon external developers' onboarding followed the process as shown in the following figure (Figure 3.4).

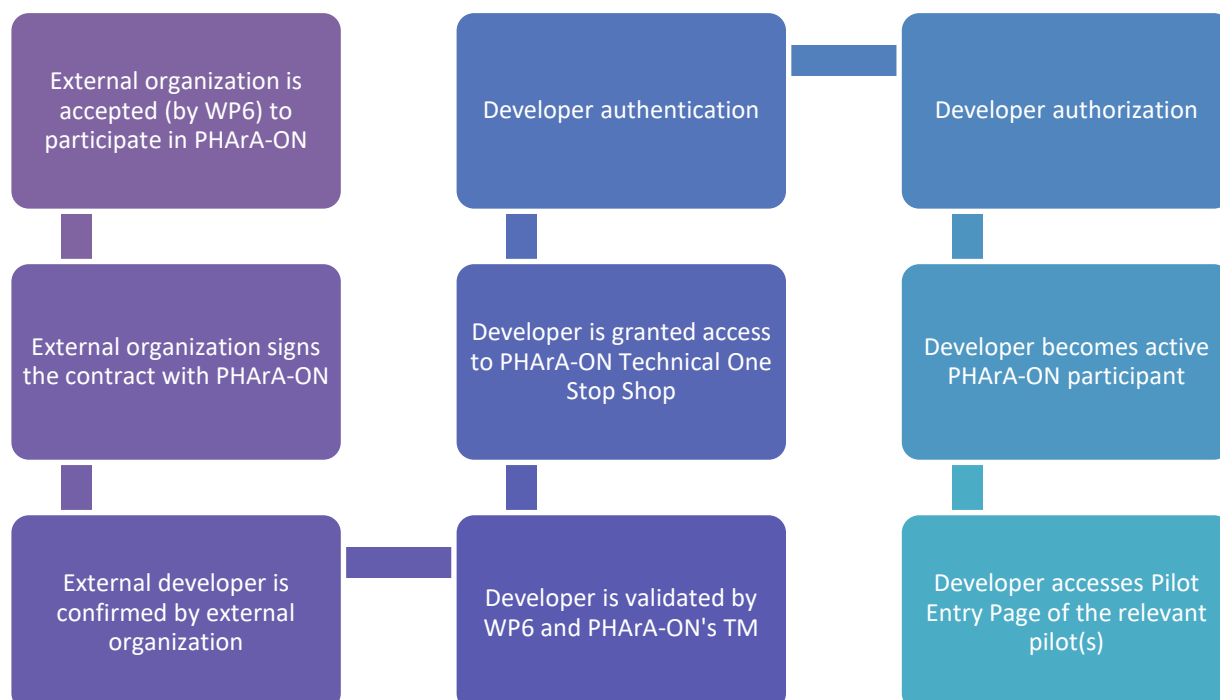


Figure 3.4 Pharaon external developers onboarding process (based on input from D4.2)

To help new partners onboarded through the WP6 Cascade Call familiarise themselves with the pilot, Pilot entry pages have been established as quick and step-by-step guidelines. Each Pilot entry page

summarises essential information and contacts of the Pilot, as well as describing the steps required to complete the onboarding process and take part in the Pharaon pilots. The goal was to improve efficiency, save time, and always have clear, succinct, and up-to-date information about the pilots is always available.

The initial page serves as an overview, acting as a starting point for new parties and providing a summary of the different pilots. This landing page directs the new partner to separate pilot pages using specific URL links, identified in the respective Pilot's section.

The Pilot Entry page environment (Welcome/Landing page and pilot-site specific pages) was set up as a wiki and hosted on the Pharaon private repository (within Pharaon *Technical One-Stop-Shop*) at the following web address: <https://gitlab.com/pharaongroup/Pilot-entry-pages>. The content can be viewed in the table Table 3.3, which provides an overview of all Pharaon Pilot Sites, including some basic information and a step-by-step onboarding process for all Pharaon Pilot Sites.

Table 3.3 Pilot entry pages welcome screen content (from D4.2)

Pilot	Pilot sites	Onboarding Process	Description	More Information
Spain: Andalusia	Andalusia	Onboarding Spain: Andalusia	The Andalusian pilot aims to make technology more accessible and user friendly to older people, building people's confidence on technologies and involving them in the evaluation of products and services, to further improve and adapt them.	https://www.pharaon.eu/about/pharaon-pilots/#andalusia
Spain: Murcia	Murcia	Onboarding Spain: Murcia	The Murcia pilot aims to establish the basis of a new TeleCare Model for the Public Health Care Service of the region starting with patients with chronic heart failure, improving their health and care services and detecting emergency situations to reduce the dependency of older adults.	https://www.pharaon.eu/about/pharaon-pilots/#murcia
Italy	Tuscany, Apulia	Onboarding Italy	The Italian pilots aim at proposing personalized integrated care for frail older adults. In doing so, the pilots emphasize the monitoring of important parameters, the physical and cognitive stimulation, the social interaction, and the work of formal and informal caregivers.	https://www.pharaon.eu/about/pharaon-pilots/#italy
The Netherlands	Many locations across the Netherlands	Onboarding The Netherlands	The Dutch pilot focuses on the PlusBus of the National Foundation for the Elderly. This a mobility service for older adults with almost a hundred locations across the Netherlands. It brings older people together by organizing trips to the supermarket, but also cultural and social activities, such as summer days at the beach and museum visits. Older people using the PlusBus want the service to be enriched with the possibility to stay connected online after an activity, and to remain physically fit to be able to continue to join activities of the PlusBus. The Dutch pilot's aim is to support the users of the PlusBus in these goals of staying connected and staying fit by extending the services of the PlusBus.	https://www.pharaon.eu/about/pharaon-pilots/#netherlands
Portugal	Coimbra, Amadora	Onboarding Portugal	The Portuguese pilot aims to develop and implement citizen-focused solutions and design an integrated care system (planning, integrated infrastructures and processes, knowledge sharing).	https://www.pharaon.eu/about/pharaon-pilots/#portugal
Slovenia	Izola, Ljubljana, Lucija, Koper, Domžale	Onboarding Slovenia	The Slovenian pilot aims at improving well-being through passive monitoring using a variety of sensing devices like wearables, sensors embedded in furniture, and environmental sensors and TV video conferencing to caregivers.	https://www.pharaon.eu/about/pharaon-pilots/#slovenia

The structure of Pilot entry pages follows the Pharaon pilots are detailed at the publicly accessible web address: <https://www.pharaon.eu/about/pharaon-pilots/>). They are organized as follows:

- Onboarding Spain: Andalusia (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-Spain-Andalusia>)
- Onboarding Spain: Murcia (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-Spain-Murcia>)
- Onboarding Italy (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-Italy>)
- Onboarding The Netherlands (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-The-Netherlands>)
- Onboarding Portugal (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-Portugal>)
- Onboarding Slovenia (<https://gitlab.com/pharaongroup/Pilot-entry-pages/-/wikis/Onboarding-process-%7C-Slovenia>)

4 *Developers' Handbook*

The chapter reports details about the final content and structure of the *Developers' Handbook*, tools used, how it evolved and the improvement process that was carried out to provide high-quality guidelines.

4.1 Software documentation tools and technology

This section describes the technology and tools used to deliver the handbook, namely Gitlab Wiki and other tools/technologies used to create Pharaon software documentation and guidelines. There are many types of software documentation that are often classified as external and internal software documentation. The external software documentation usually refers to end-user documentation (note that end-users in this context are Pharaon developers that will use developed technology and tools to produce integrated services for Pharaon end users – caregivers and older adults), API documentation and just-in-time documentation. Examples of internal software documentation include administrative, requirements, standards, design, and similar documentation used internally within an organisation or a project. Different specialised tools may be used to produce different types of documentation. The focus of this task (T4.1) and all work described in related T4.1 deliverables refers to developers' documentation that provides instructions to developers for building the software and guides them through the development/integration process. A software documentation tool can be defined as any technology that assists developers and other stakeholders in authoring (creating, updating), and delivering documentation for software projects. There are numerous benefits to documenting software, benefiting both authors (creators) and users (developers in our case). For authors it increases efficiency and the quality of created content, improves technology usability, fosters collaboration and knowledge sharing, while for users (developers) it facilitates faster onboarding and understanding, enhances maintenance and support, and in general improves and speeds up technology adaptation and integration.

As for the software documentation tool, there are several key features that every documentation tool should support. This includes:

- **Support for Markdown and HTML** – these are two of the most popular programming languages for creating well-structured and visually appealing technical guidelines,
- **Access control** – allows control/management of access on who can read and create certain parts of documentation, usually using roles and permissions,
- **Feedback support** – it is very important that software documentation tool support easy feedback provision from users,
- **Seamless article redirection and linking support** to redirect users from one article to another,
- **Automatic creation of table of contents** to simplify navigation,
- **Backup and versioning** support.

By leveraging these features any project can enhance its documentation processes to efficiently deliver high-quality content.

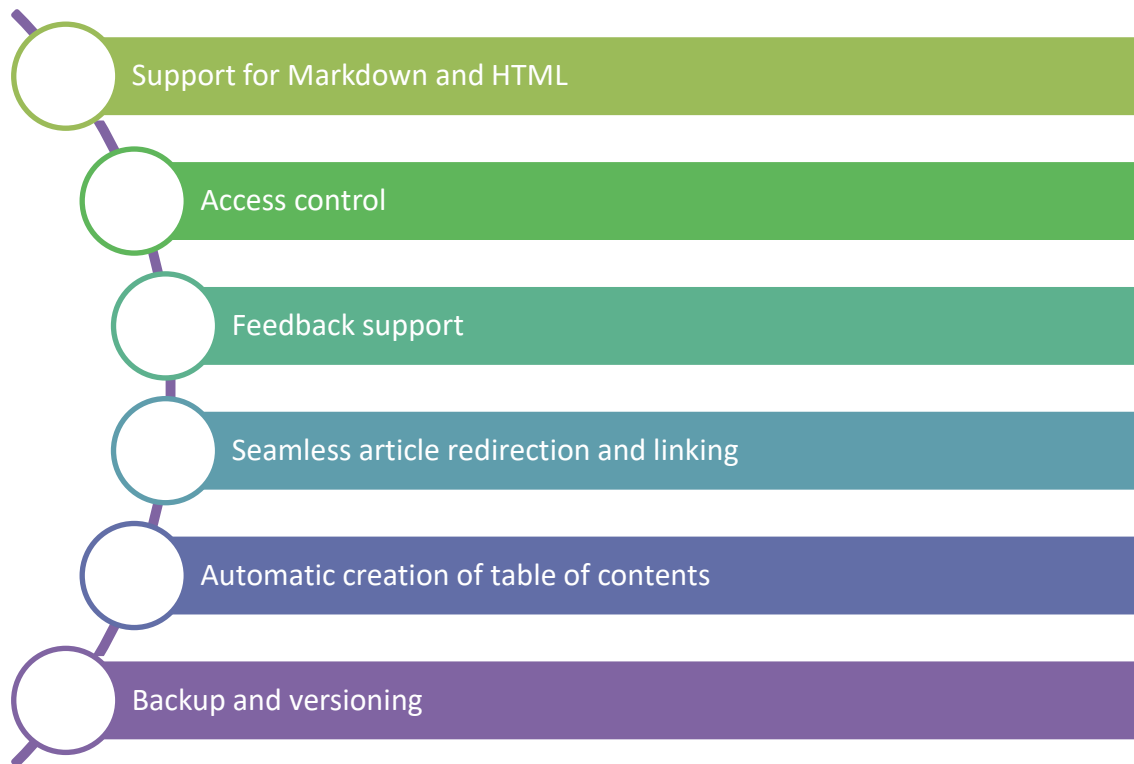


Figure 4.1 Key features of software documentation tools

4.1.1 Gitlab Wiki

Gitlab Wiki allows developers to collaborate on creating documentation and share information. It facilitates tracking revision history and help structure information into folders for better organization. Gitlab Wiki as tool for *Developers' Handbook* has been selected at very beginning of the project and the task T4.1, and rationality for using explained in first iteration of this deliverable, in D4.1 Developer guidelines & templates – first (released in March 2021). As described in the mentioned deliverable, wikis are often used as a tool for documentation and are especially convenient for technical content that is to be provided in Pharaon *Developers' Handbook*. The advantages are as follows:

- more focused on collaboratively improving each topic,
- more open to every developer on the project,
- focus on content and it is easy to find and update information,
- available together with code repositories.

Beside the advantages mentioned in D4.1, Gitlab Wiki supports multiple markup formats, including used Markdown, Rdoc, AsciiDoc and others. Within Pharaon, Markdown format was used which enabled versatile portability of the developed content as guidelines can be easily converted to HTML, MS Word, PDF, or any other format. In addition, Markdown is recognised for its universal application, ease of use and readability.

4.1.2 Docusaurus

Docusaurus² is an open-source static site generator that is designed for content-focused projects. Similar to Gitlab Wiki, it supports writing content in Markdown language. In addition, it has support for MDX, which allows creating more dynamic and interactive documentation by embedding React components directly in Markdown content. It provides a framework and set of tools specifically designed for creating documentation sites.

The main goal of Docusaurus is to streamline the documentation process for developers, enabling them to focus more on content creation and simplifying the complexities of website development.

The principal features of Docusaurus are:

- **Markdown Support:** Documentation content is written in Markdown, a lightweight markup language that is easy to write and read. This enables developers to focus on content creation without worrying about HTML formatting.
- **Customization:** While Docusaurus provides default themes and layouts out of the box, it also offers extensive customization options. Developers can tailor the look and feel of their documentation site to match their brand or project requirements.
- **Versioning:** Docusaurus includes built-in support for versioning documentation. This allows users to navigate between different versions of the documentation.
- **Search Functionality:** Docusaurus comes with a built-in search feature that enables users to find information within the documentation site.
- **Localization:** Docusaurus supports localization, allowing developers to translate their documentation into multiple languages.
- **Sidebar Navigation:** Docusaurus generates a sidebar navigation menu based on the directory structure of the documentation files. This provides users with easy access to different sections of the documentation.

4.2 Structure and content

The final structure of Pharaon *Developers' Handbook* is illustrated in Figure 4.2. The focus of work in the final version of this deliverable was completing all guidelines about Pharaon technology, including Ecosystem Hub and platforms.

In comparison to the structure of the *Developers' Handbook* described in the previous version of deliverable (in D4.2), there have been several changes. Main change is of course changed structure of Pharaon Technology guides section which contains all guidelines for different technology used in Pharaon project and for the integration using Ecosystem Hub. This section was divided to three main parts:

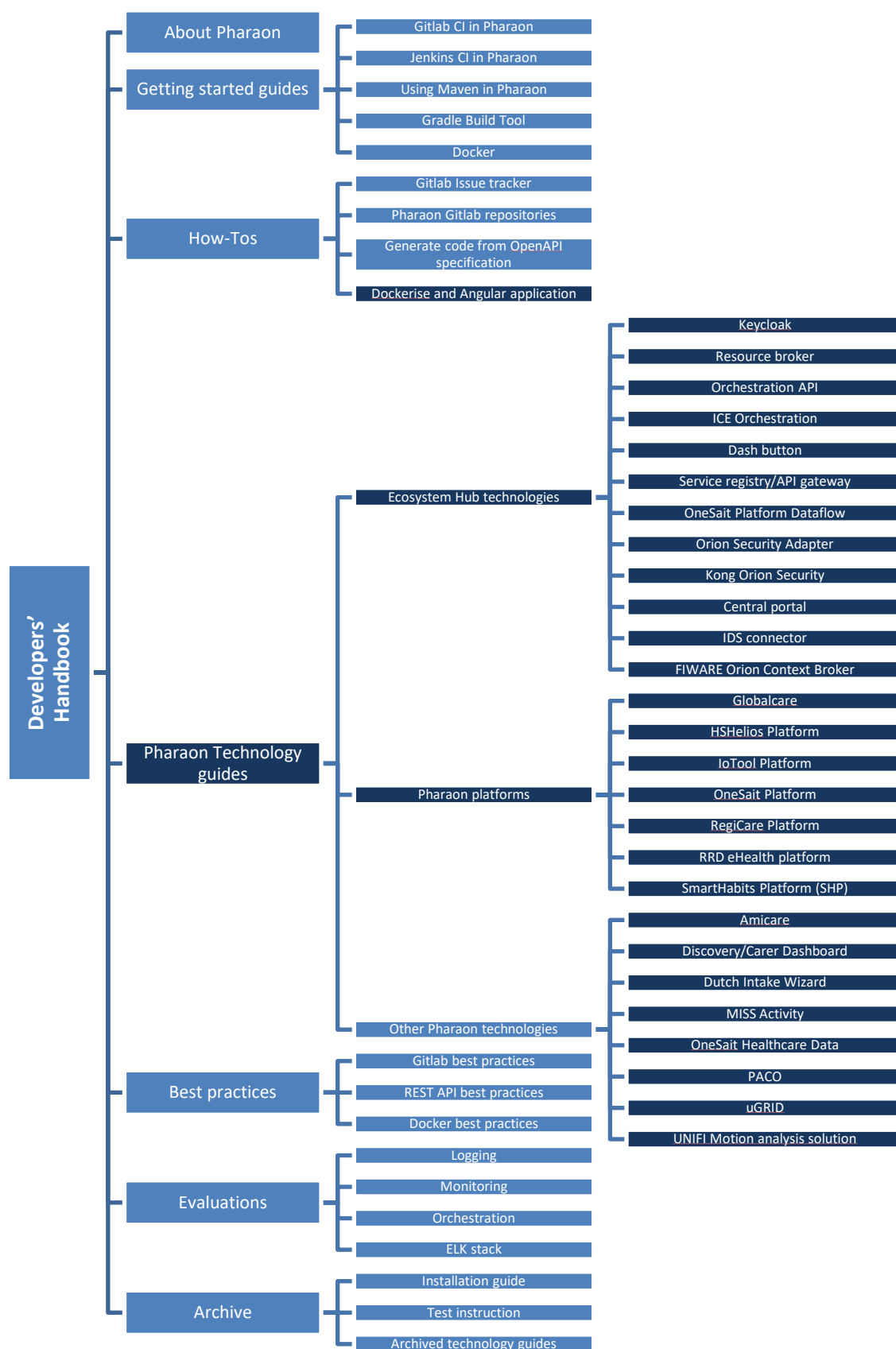
1. **Ecosystem Hub technologies** – provides general information about Ecosystem Hub and how to integrate technical components in order to comply with Pharaon integration standards and contains guidelines about all Ecosystem Hub technologies, including Keycloak, Resource broker, Orchestration API, ICE Orchestration, Dash button, Service registry/API gateway,

² <https://docusaurus.io/>

OneSait Platform Dataflow, Orion Security Adapter, Kong Orion Security, Central portal, IDS connector, and FIWARE Orion Context Broker.

2. Pharaon platforms – provides guidelines about using Pharaon platform technologies such as OneSait Platform, Globalcare, HSHelios Platform, IoTool Platform, RegiCare Platform, RRD eHealth platform and SmartHabits Platform (SHP).
3. Other Pharaon technologies – provide all other Pharaon technologies guidelines such as Amicare, Discovery/Carer Dashboard, Dutch Intake Wizard, MISS Activity, PACO, uGRID and UNIFI Motion analysis solution.

Another structural change is the addition of the archive section. This section contains all guidelines created in the beginning of the project that are not primarily used anymore, however they can still be useful in certain cases. For example, installation instruction for some technology has changed since in the beginning some technology required direct installation and manual configuration of all components on virtual/real machines. However, within the scope of task T4.4 and the introduction of concept of Pharaon sandboxes, the deployment and installation of these technologies has been significantly improved and simplified using docker and related technology. More info can be found in deliverable D4.11 (final version D4.12 expected to be submitted in June 2024). In addition, some initial testing instructions that were provided for certain technology are obsolete, and more general end-to-end integration test instruction were provided later within the scope of work on WP5 Technology Ecosystem Integration and described in related deliverables.

Figure 4.2 Final structure of *Developers' Handbook*

4.3 Timeline

This section presents the timeline of major *Developers' Handbook* releases and presents how handbook content evolved over time from first release, described in D4.1, to the final release presented in this deliverable. The main timeline overview is presented in Figure 4.3.

The first version of the *Developers' Handbook* was presented in D4.1 and formally released in March 2021. The focus of the work reported in that deliverable was on initial considerations and information of the initial work done with regards to the Technology Support Tools and more concretely with regards to the process and tools related to Software Developing Life Cycle (SDLC) within Pharaon and established of Technical One-Stop-Shop. The initial version of the *Developers' Handbook* contained instructions, guidelines, recommended open-source tools and best practices needed by Pharaon developers to get started with technical (adaptations and integration) work on Pharaon project. At the beginning of the projects the technology development was only initial phase thus it was important to support developers with clear process description, available support tools and guidelines about basic development technologies, as documented in D4.1.

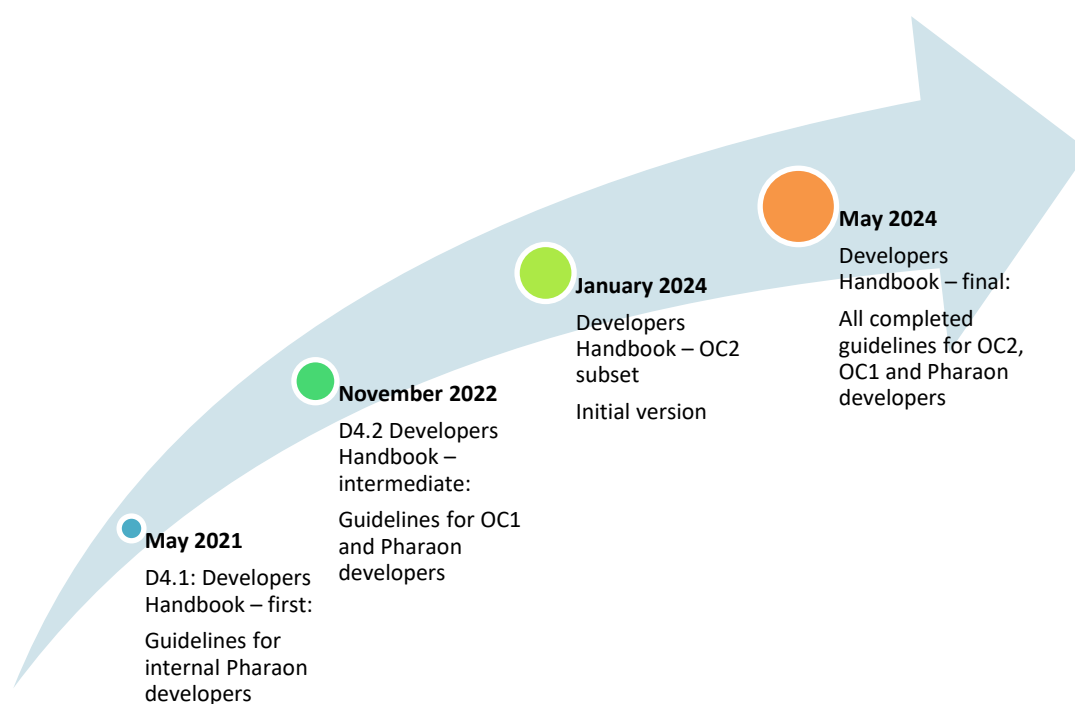


Figure 4.3 Developers' Handbook timeline overview

The second, intermediate release of *Developers' Handbook* was presented in D4.2 in November 2022. The work continued to focus on sharing relevant knowledge, common practices, processes, useful tools, and Pharaon-specific processes and key information to facilitate and further Pharaon technical work. Main difference in comparison with the first version was guidelines and documentation relevant for participants of 1st Open Call (described within the work of WP6) and the introduction of the Pilot Entry Pages. All parts of the *Technical One-Stop-Shop* have been extended with new content and the first version of several Pharaon technology guidelines about platform technologies have been added. The structure of intermediate version of *Developers' Handbook* with highlighted changes to the previous version is shown on Figure 4.4 (from D4.2). The light blue boxes highlight the new contents

included in comparison to first release, the aquamarine boxes highlight the handbook's sections that were reported as "in progress" in D4.1 and were completed in D4.2.

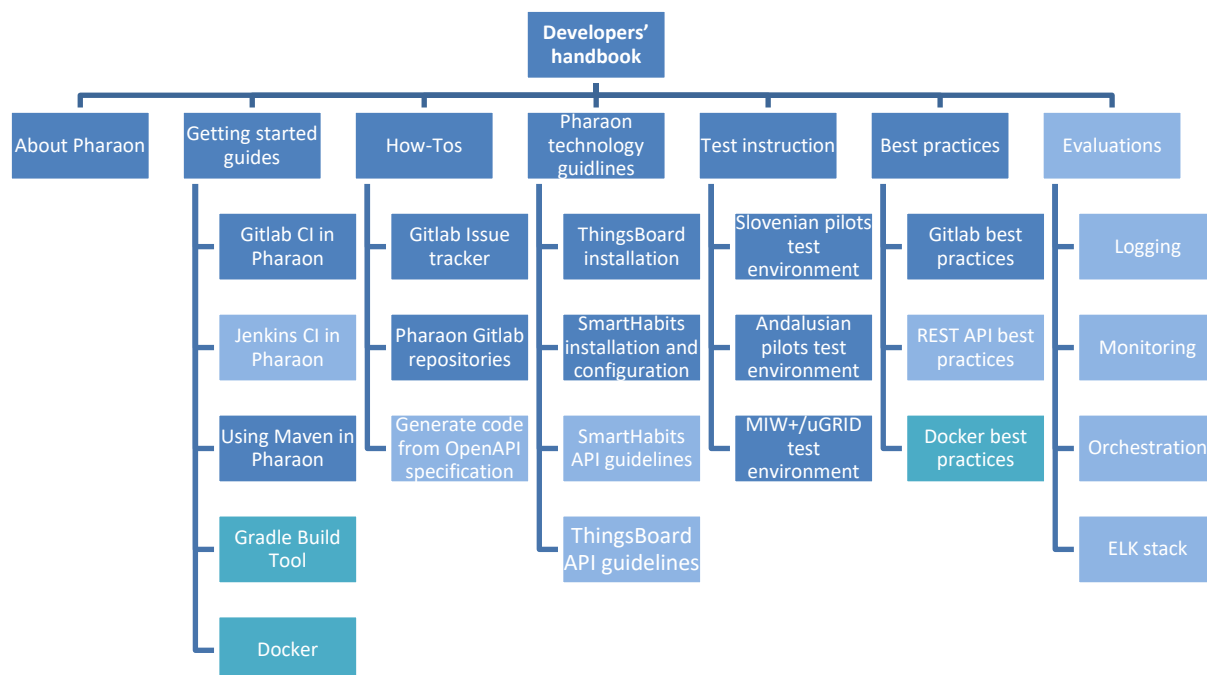


Figure 4.4 Developers' Handbook structure in intermediate version from D4.2.

This final version, described in this deliverable, has been updated to include all the remaining technology guidelines, in particularly Ecosystem Hub technology guidelines that describe how to integrate technical components in order to comply with Pharaon integration standards. The final version also added several new and improved platforms and other technology guidelines, among others, based on internal guidelines improvement process, described in next section. The structure of the final version of Pharaon *Developers' Handbook* was described in previous section and presented on Figure 4.2.

As illustrated on Figure 4.3, within the work on this deliverable, to support external developers participating in 2nd Open Call projects, the subset of developers' guidelines relevant for external developers has been prepared and shared with participants ahead of final release plan. The first version of *Developers' Handbook* subset for OC2 participants was released in January 2024, before planned OC2 implementation and integration phase and with coordination and help from WP6 leading partners. The structure is presented on Figure 4.5.

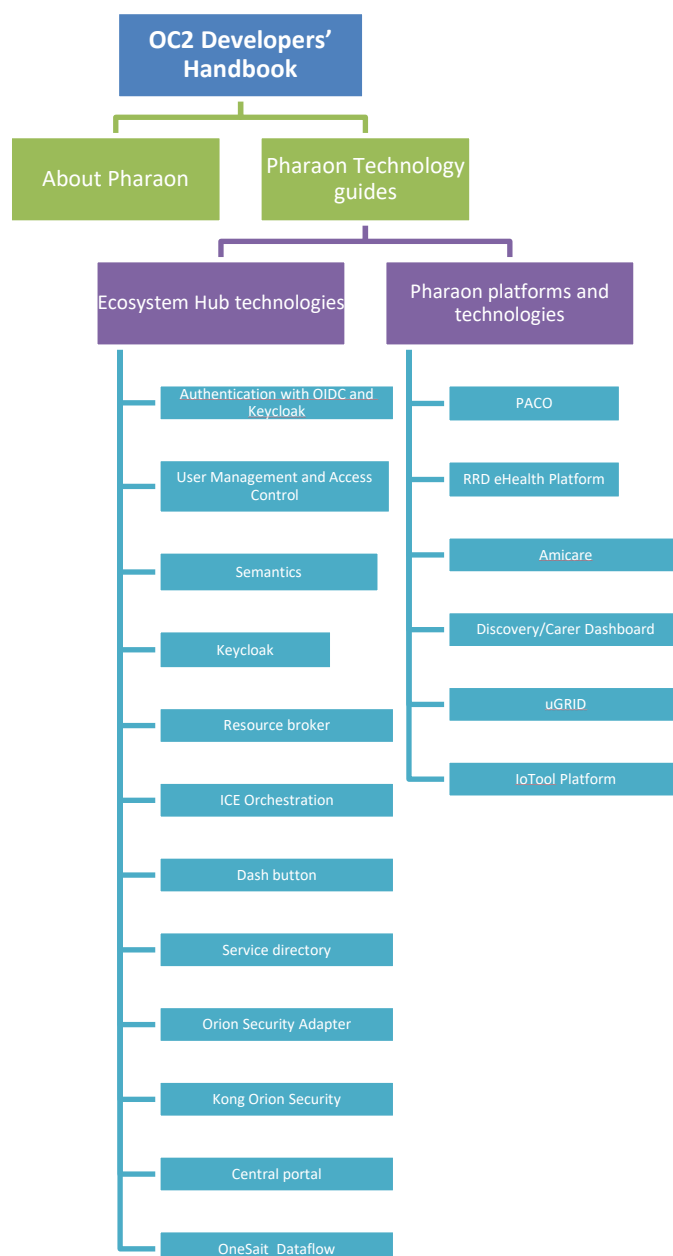


Figure 4.5 Structure of *Developers' Handbook* subset for OC2

4.4 Guidelines improvement process

This chapter describes the Pharaon internal *Developers' Handbook* improvement process that was crucial for creating effective and high-quality guidelines. It was performed continuously during the development of guidelines and involved multiple stages with the main objective of enhancing the accuracy and quality of the guidelines. The process included, among other things, analysing the current status of the guidelines, identifying gaps, defining internal review guidelines, selecting internal reviewers, managing the reviews, collecting feedback and final guidelines improvements.

When it comes to developers' documentation, internal review demands intense focus going beyond mere error detection of grammar and typos. It ensures accuracy, identifies issues, gaps, helps achieve

project technical objectives, saves costs, and mitigates risks. Thorough guidelines review offered the following benefits:

- improved accuracy helps detect errors early, saving time for developers start using the guidelines,
- enhanced completeness by identifying gaps in the descriptions,
- increased clarity and consistency, making the guidelines are easier to interpret and follow,
- saves time and cost since internal reviews can identify duplicate or unnecessary information,
- ensures all guidelines are aligned in terms of basic content and provided information.

4.4.1 Overall iterative improvement process

The overall iterative *Developers' Handbook* improvement process is illustrated with Figure 4.6. Once a need for a guideline is identified, the assigned partner starts creating content. It may use instructions for adding specific technology guidelines to Pharaon Developer's Handbook³, and other input materials. After self-evaluation of the created content, and necessary corrections, the author publishes an initial version of the guideline. The guideline is then evaluated through an internal review process, which is described in more detail in 4.4.2. Following content improvements based on feedback from internal review, author publishes revised version that is used by the developers (internal or external) for development of integrated solutions. During this phase, additional formal or informal feedback may be received, based on which revised version of the guideline may be published.

³ <https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Instructions>

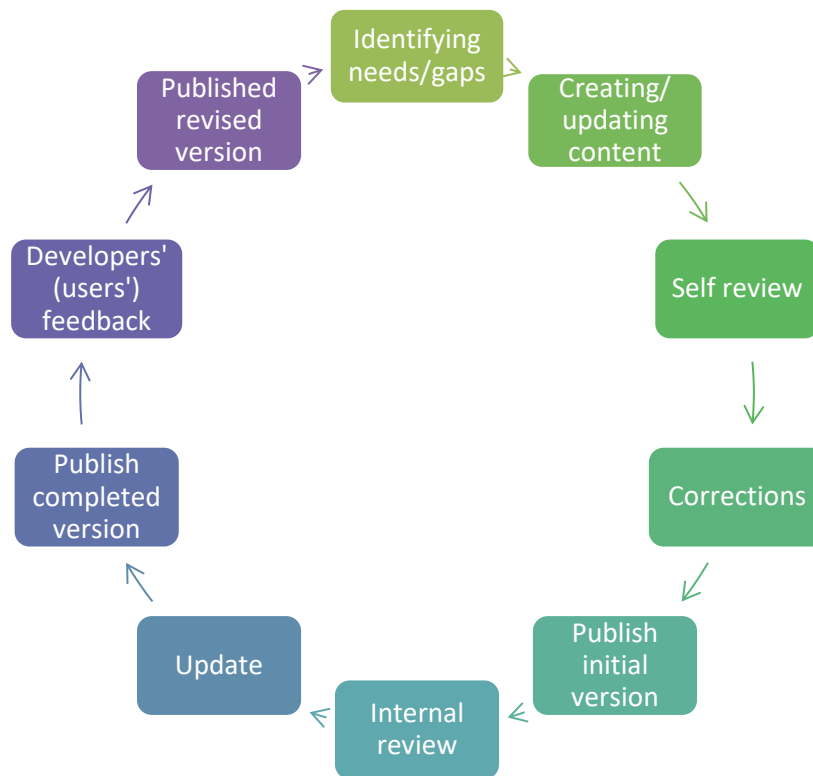


Figure 4.6 Overall iterative guidelines improvement process

4.4.2 Technology guidelines internal review process

Although all guidelines developed and documented within *Developers' Handbook* followed the improvement process depicted in Figure 4.6., special focus was put on the quality and accuracy of Pharaon technology guidelines. Several categories of guidelines have been produced, including various how-to, best practices, and getting started with technology and tools guides. The main and most important outcome are the Pharaon technology guidelines, which describe how to start using and integrating Pharaon technology to develop integrated services. For this reason, it was important to produce effective and high-quality guidelines with the help of the technology guidelines internal review process, described in this chapter.

The technology guidelines internal review process is described with the Figure 4.7. The status and gap analysis are very important, considering that guidelines development and improvements was an iterative process that followed the development of technology. It was important to understand not only the status of guidelines, but also the status and planning of technology development to identify any misalignment (i.e. the guideline describes usage of older/obsolete API version). Secondly, it was very important to define basic guideline template and instructions to align all guidelines. Mentioned instructions describe how to add guideline to *Developers' Handbook* Gitlab Wiki, how to structure it (basic about section, access information, code and configuration examples, links to external resources), how to link it in main guideline repository and where to find more information about used GitLab Flavored Markdown language. Based on this assigned partner created/update the content of the technology guideline.

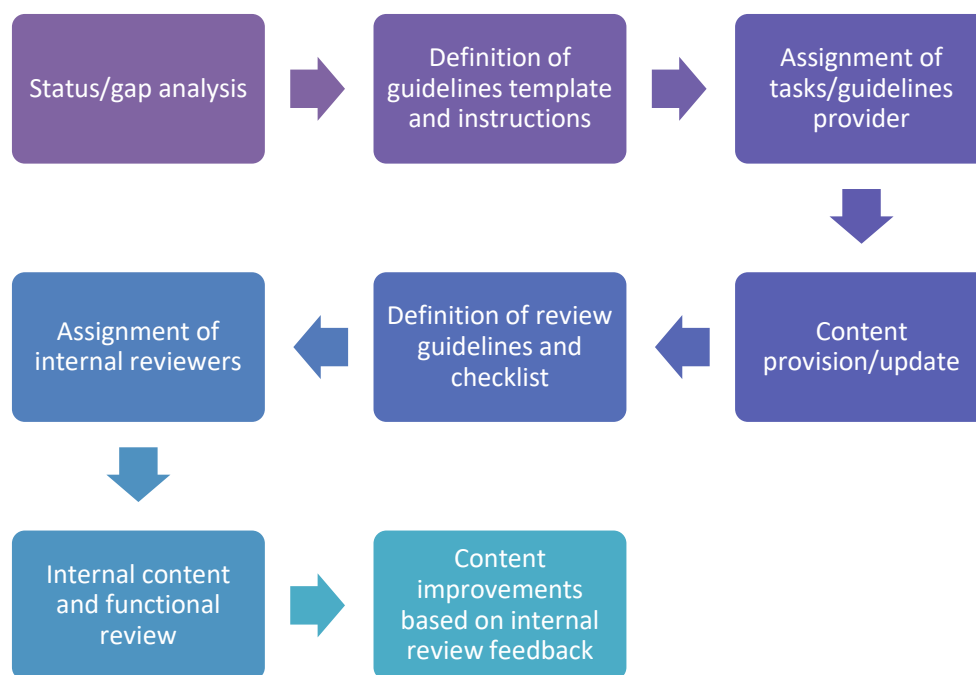


Figure 4.7 Technology guidelines internal review process

In parallel with the guideline content creation and update, the key was to define concrete internal review guidelines and checklist that will help internal reviews in the evaluation process. The structure of internal review checklist and evaluation form is illustrated with Figure 4.8. The actual form was implemented using MS Excel and provided to partners via project common document repository (Collabto). There are five main parts:

1. Basic information section provides details on review guideline number (for reference), guideline name, Pharaon partner(s) that created/authored the guideline, and direct link to the guideline on Pharaon *Developers' Handbook* Wiki pages.
2. The About Reviewer part provides details about the assigned partner and the date of the actual review.
3. The guideline review basic checklist section is a list of main aspects of the guidelines to which reviewer needs to pay attention to. This includes checking if the content is clear (in term of right language, grammar, and abbreviations), whether there is introduction section with basic information about the technology, does it have links to external content for more details and trying if all links work. Furthermore, it was important to check if related technology (software/service) is accessible for testing/developing (i.e. can it be downloaded or accessed in the test environment and if all access information is provided). The guidelines should also include code/configuration examples for getting started with development/integration. Finally, it needs to be assessed to ensure it is suitable and not too short.
4. The most important section is the overall review, with final review (for simplicity of the internal review process only acceptable and need improvements assessment were defined). This section also includes final comments and suggestions for the authors.
5. The final section addresses the improvements status, indicating whether the feedback and recommendations have been completed or are still ongoing.

After assignment of internal reviewers to specific guideline review, and performing actual evaluation, review checklist and providing suggestions with final assessment, using received feedback the author performed final improvements to the content.

After assignment of internal reviewers to certain guideline review, and performing actual evaluation, review checklist and providing suggestions with final assessment, using received feedback the author performed final improvements to the content.

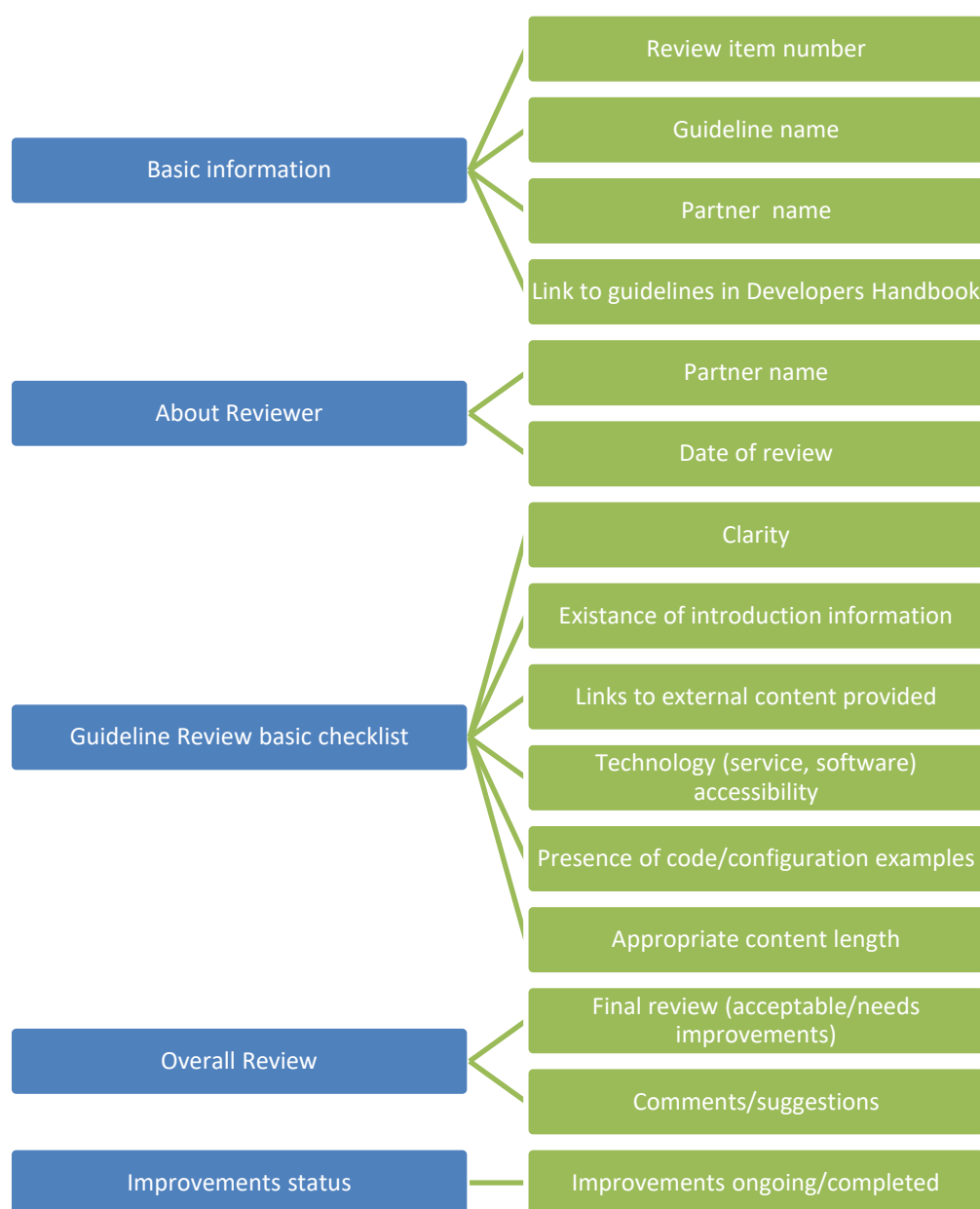


Figure 4.8 The structure of internal review checklist and evaluation form

4.4.3 The results

The results of the technology guidelines internal review were added to mentioned MS Excel provided via project Collabto. The majority of internal reviews of technology guidelines were performed during February and March 2024. In summary, out of 28 technology guidelines that were in the initial review process iteration, the majority were assessed as acceptable with only minor suggestions for improvements, while only 3 were assessed as needing improvements. The first results of guideline review checklist are depicted in the Figure 4.9. The snapshot shows an overview information with main checklist results but does not contain all details collected in the evaluation form (i.e. comment/suggestion column with all feedback information for each reviewed guideline is not shown).

#	Guideline		Internal Reviewer		Guideline Review checklist						Final Review [acceptable/needs improvements]
	Guideline name	Partner	Partner name	Date of internal review	Is it clear? (language, grammar, abbreviations)	Does it have About section with basic information about the technology?	Does it have links to external content for more details? Do all links work?	Is the related technology (software/service) accessible for testing/developing? I.e. can you download it or can you access the service in test environment?	Does it have code/configuration examples for getting started with development/ integration?	Is the content length suitable? (or is it too short)	
1	Ecosystem Hub overview	RRD	SCMA	21/02/2024	Y	Y	N/A	N/A	N/A	Y	Acceptable
2	Hub: Authentication with OIDC and Keycloak	RRD	SCMA	21/02/2024	Y	Y	N/A	Y	Y	Y	Acceptable
3	Hub: User Management and Access Control	RRD	ENG	23/02/2024	Y	Y	N	N/A	Y	Y	Acceptable
4	Hub: Semantics	RRD	ENG	23/02/2024	Y	Y	N	N/A	Y	Y	Acceptable
6	Resource broker	RRD	ENG	08/03/2024	Y	Y	Y	Y	Y	Y	Acceptable
8	ICE Orchestration	ICE	ENT	26-02-2024	Y	Y	Y	Y	N/A	Y	Acceptable
9	Dash button	ICE	CETEM	28-02-2024	Y	Y	N	Y	Y	Y	Acceptable
10	Service directory	ICE	MIWEnergia	20-02-2024	Y	Y	Y	Y	Y	Y	Acceptable
11	OneSalt Platform Dataflow	INDRA	ENT	26-02-2024	Y	Y	Y	Y	N	N	Acceptable
14	Central Portal	CETEM	MIWEnergia	20-02-2024	Y	Y	N	N	N	Y	Acceptable
15	FIWARE Orion Context Broker	ENG	ICE	27-02-2024	Y	Y	Y	Y	N*	Y	Acceptable
16	Globalcare	GlinTT	ASC	12-03-2024	Y	Y	Y	N	N	Y	Needs improvements
17	HS.Hellos platform	GlinTT	ASC	12-03-2024	Y	Y	Y/N	Y	N	Y	Needs improvements
18	OneSalt Platform	INDRA	ENT	26-02-2024	Y	Y	Y	Y	N	N	Acceptable
19	Regicare Platform	ADS	CETEM	28-02-2024	Y	Y	Y	Y	Y	Y	Acceptable
20	RRD eHealth platform	RRD	GLINTT	26-02-2024	Y	Y	Y	Y	Y	Y	Acceptable
22	Amicare	CETEM	GLINTT	26-02-2024	Y	Y	Y	N	N	Y	Needs improvements
26	PACO	RRD	ICE	27-02-2024	Y	Y	Y	Y	Y	Y	Acceptable
27	uGrid	MIWEnergia	ICE	27-02-2024	Y	Y	Y	Y	N	Y	Acceptable

Figure 4.9. Snapshot of internal review checklist results in first iteration

5 Conclusion

One of the crucial goals of this task was to provide the appropriate technical personnel with the knowledge, instructions, and guidelines related to extensive and frequently heterogeneous technological activities in a project. In order to promote general knowledge sharing and the adoption of best practices, the effort was focused on identifying and sharing the most relevant information about adopting and developing Pharaon technologies and recent software engineering trends. One of the main tasks was organizing all the relevant information so that it could be easily accessible and particularly helpful for the Pharaon internal and external developers' daily hands-on work. This was achieved with established Pharaon Technical One-Stop-Shop, main entry point through which developers can access all the documentation, guidelines, different repositories, recommended open-source tools, best practices and the most important technical information needed to work on the Pharaon technical aspects.

The work described in this final deliverable was related to realization of all remaining guidelines about Pharaon technology, creating a subset of *Developers' Handbook* for participants of Open Call 2, and on aligning and enhancing all content by the newly implemented internal guideline improvement procedure. This was essential for producing high-quality and effective guidelines for developers.

Appendix A: Technology Guidelines reference list

The table below lists all technology guidelines that have been created and finalized within the work on this version of deliverable (D4.3), completed in period from November 2022 until May 2024. All other guidelines, getting started guides, how-tos, best practices, tools survey, pilot entry pages and other content that was mostly completed before November 2022 is documented in the previous version of this deliverable, in D4.1 and D4.2.

Table 4 Technology Guidelines reference list table

#	Technology Guideline name	Short description	Partner name	Link to guidelines in Pharaon Gitlab Wiki	Link to additional guidelines and documentation ⁴
1	Ecosystem Hub overview	This document describes how to integrate technical components in order to comply with Pharaon integration standards. Components are integrated with the help of a set of integration standards and tools, which we call the Pharaon Hub. It is built on top of widely used interoperability standards: OpenAPI and OpenID Connect (OIDC).	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Ecosystem-Hub	
2	Hub: Authentication with OIDC and Keycloak	Single sign-on (SSO) is performed via Keycloak OIDC. In Pharaon approach, OIDC tokens are used for direct exchange between technologies (e.g. via API endpoints), and technologies are responsible for their own (data) security.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Ecosystem-Hub/Authentication-with-OIDC-and-Keycloak	
3	Hub: User Management and Access Control	Explains how to handle user administration and inter-user access control, that is, which user can access what from which other users.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Ecosystem-Hub/User-Management-and-Access-Control	
4	Hub: Semantics	In Pharaon we have a data-centric approach, which means data is designed to outlast particular data providers or users, by making the data and definitions persistent and adding sufficient meaning to the data. In the Pharaon Hub, shared data definitions can be semantically annotated and stored in the common definitions facility. The definitions are JSON schemas for use with OpenAPI accessible via the \$ref reference construct. In addition to data, our approach also enables APIs and services to be annotated with semantics.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Ecosystem-Hub/Semantics	

⁴ Additional guidelines or documentation deployed on partner's host server

5	Keycloak	Keycloak is the Identity and Access management system used within the Pharaon project to provide authentication and authorization mechanisms for components and pilots.	ENG	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Keycloak	
6	Resource broker	The resource broker provides search functionality to the APIs registered in the API registry.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/RRD-Resource-Broker	https://dev.senseact.com/pharaon-resourcebroker/api-docs/
7	Orchestration-API	The Orchestration APIs are a collection of APIs that are utilised by the Orchestration component. The Accounts API provides information on user accounts and organisations, the Config API organises the configuration, the Designer API which supports the Camunda process modeller and the Process Engine API offer endpoints that are utilised throughout the Orchestration component.	ICE	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Orchestration-API	https://process-engine-api.pharaon.icelab.cloud/api/
8	ICE Orchestration	The ICE Orchestration tool is a web application that allows users to create processes using a modified Camunda interface, create forms for custom input data, manage services using the service registry, a control panel to manage the processes, and task manager to allow users, managers and admins of an organisation to assign, complete and manage tasks assigned via the processes.	ICE	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/ICE-Orchestration	
9	Dash button	The Dash Button is a web component that allows component owners to manage end user logins with a common button.	ICE	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Dash-Button	
10	Service registry/API gateway	The Service Directory is a repository of OpenAPI compliant services with the capability of users to add, edit, remove or simply see the services available to them. It also contains the Service Directory API which is the back-end functionality. This is paired with the API Gateway, which provides authentication support to bridge the Camunda functionality of Orchestration with any other service.	ICE	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Service-directory	
11	OneSait Platform Dataflow	The Onesait Platform Dataflow system is a robust and efficient engine designed to facilitate the seamless streaming of data in real-time. Utilizing the Data Collector component enables users to effectively route and manage data within their data streams.	INDRA	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/onesait-platform-dataflow	https://dataflow-docs.onesaitplatform.com/

12	Orion Security Adapter	The Orion Security Adapter service is a proxy security layer to access protected REST APIs of FIWARE Orion Context Broker. Any users, identified by the token, can access data based on the fiware resources (entities or subscriptions) and verb (GET, POST, etc.). This service also includes the group membership mechanism where only users belonging to this group can access the data. Both token and group membership are retrieved from Keycloak.	ENG	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/Orion-Security-Adapter
13	Kong Orion Security	Kong Orion Security service is a proxy security layer to access protected REST APIs of FIWARE Orion Context Broker. Any users, identified by the token, can access the fiware resources (entities or subscriptions) based on their roles.	ENG	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/Kong-Orion-Security
14	Central Portal	The Central Portal (CP) serves as a unified access point to the Pharaon Project, offering information about each pilot, along with details about their platforms and technologies. Within the CP, users can also view showcase demos (with no real data) or videos demonstrating the functionalities and capabilities of each Pharaon component.	CETEM	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/central-portal
15	IDS connector	IDS connector is one the main components of data space. With IDS connectors, businesses can achieve secure and trustworthy data exchange, have granular control over data sharing, and seamlessly connect diverse data sources. The standardized protocols further ensure a consistent and efficient data-sharing experience. IDS connectors are essential for bridging various platforms and systems, and are instrumental in facilitating communication between different data ecosystems, ensuring smooth cross-platform data exchange.	ENT	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/IDS
15	FIWARE Orion Context Broker	FIWARE Orion Context Broker is a generic enabler provided by the FIWARE platform to manage data context. There are no Orion REST APIs accessible directly, but within the Pharaon project, two security services have been developed to provide proxy security layers to protect Orion REST APIs.	ENG	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/FIWARE-Orion-Context-Broker
16	Globalcare	Globalcare is a Hospital software designed and developed by Glintt. It consists of 4 product lines: HMS - Hospital Management System; Clinical; Pharma & Logistics; and BI&A – Business Intelligence & Analytics. These products cover virtually all hospital activities and meet the needs of all hospital teams. Able to be integrated and communicate with other systems, GLOBALCARE can be provided as a complete software package or as a modular solution, according to business needs.	Glintt	https://gitlab.com/pharaongroup/developer-s-handbook/-/wikis/Pharaon-technology-guides/Globalcare

17	HS.Helios platform	HS.HELIOS is a proprietary integration system that promotes interoperability, by regulating and analyzing the exchange of messages between different health information systems that are currently deployed at a Health institution.	Glantt	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/HSHELIOS-platform	
18	OneSait Platform	Onesait Platform is an open technology platform developed by Minsait. Onesait Platform enables the development of all types of solutions by abstracting the complexity of the underlying technical layers. It helps us to simplify and standardize developments seeking independence in lower technical decisions and flexibility in the face of technological changes: allowing to work with several clouds in multi-cloud formats; making cloud and on-premise architectures compatible; and managing the complexity of technologies and protocols.	INDRA	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/onesait-platform	https://onesaitplatform.com/space/DOCT
19	IoTool Platform	IoTool is comprehensive, one-stop IoT Solution incorporating Client / Dashboard / Gateway / Cloud / API. The Android device (smartphone, tablet or TV box), ESP32 / nRF microcontroller, WiFi device or MQTT client is used as a gateway to sending and receiving data to and from the IoTool cloud or other clouds (IBM Watson, Amazon, ThingsBoard and similar).	SENLAB	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/IoTool-Platform	https://docs.senlab.io
19	Regicare Platform	RegiCare is an information system built around the primary processes of welfare organisations. It's tailored around registration, simplification and standardisation of administrative and production processes. RegiCare's purpose: to provide more insight into the impact of the work so that the added value to society is even more visible.	ADS	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/RegiCare	https://regicare.demo.adsysco.nl/docs/index.html
20	RRD eHealth platform	The RRD eHealth platform provides a back-end for general data storage and an app for collecting sensor data, administering questionnaires, and coaching. It is used as the back-end for PACO.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/RRD-eHealth-Platform	https://www.senseact.com/docs/api/
21	ThingsBoard API Guidelines	The ThingsBoard Community Edition (CE) is an 3rd party open-source IoT platform for data collection, processing, visualization, and device management. The guidelines provide brief examples on how to use and get test data from ENT's instance of ThingsBoard CE in test/staging environment.	ENT	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/ThingsBoard-API-guidelines	
22	SmartHabits Platform	Part of intelligent privacy-aware home care assistance solution that is, within the scope of Pharaon project, used for interoperability between different Pharaon technologies, and for processing and detection of anomalies in sensor data (unusual value in sensor data, an outlier) using machine-learning technology.	ENT	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/SmartHabits-Platform	https://test.smarthabits.com.hr/api-gateway/swagger-ui/index.html

23	Amicare	AmiCare is a non-invasive system that monitors older-adults' daily habits and greatly contributes to the peace of mind of relatives and carers thanks to the power of IoT	CETEM	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Amicare	https://amicare-swagger-bucket2.s3.eu-central-1.amazonaws.com/index.html
24	Carer Dashboard / Discovery	The Carer Dashboard is a tool designed for formal and informal caregivers of older adults. It serves to visualize sensor data, manage user profiles, and handle sensor-related information. The primary components include the Carer Dashboard Frontend, Carer Dashboard Backend, Discovery, and its Data Leechers. Discovery, initially the main component, encountered stability issues and was replaced during the project. The backend now manages inhabitant information, connects inhabitants with sensors, and handles data from SmartHabits.	ASC	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Carer-Dashboard	
25	Dutch Intake Wizard	Web application that guides end users from the Dutch pilot through the intake by means of a dialogue.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Dutch-Intake-Wizard	
26	MISS Activity	Miss Activity is an easy-to-use activity monitor for the elderly. The MISS Activity is unique because: A)The specially adapted algorithm is tailored to the slow gait of the target group. B) The user interface method of use has been developed with and for the target group. C)The MISS Activity can be carried in the pocket. The activity meter validly measures the number of steps and active minutes. D) The data is displayed in a simple way in the app. In the app, the user can see the data of that day and the distribution of the number of steps and active minutes spread over the day. In the same app, the participant can indicate the personal step goal and active minute goal.	MAIN	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Miss-Activity	
27	PACO	Persuasive E-health Agents for Coaching Older adults: Coaching tool to help adults toward a dietary and healthier behaviour change.	RRD	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/PACO	https://dev.senseeact.com/servlets/paco/senseeact/swagger-ui/index.html
28	uGrid	uGRID is an energy management software developed by MIWenergia with simple solutions that helps you control and monitor your energy consumption and achieve the maximum possible sustainability. In order to know the data in real time, a series of equipment and sensors are deployed in the internal network of the facility. The measurement collects accurate information on consumption in real time and is displayed through the online platform	MIWenergia	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/uGRID	https://api-pharaon.miwenergia.com/swagger/index.html

29	UNIFI Motion analysis solution	The solution is used for the detection of the movements made by an elderly person and for an assessment of their motor skills. The technology is based on a miniature sensor node (MEMS IMU) integrated into a Micro Control Unit (MCU) containing the Bluetooth Low Energy (BLE) module.	UNIFI	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/UNIFI-Motion-analysis
30	Sensors data integration in Fiware Unifi	The sensors, integrated into the IORT Scenario, were developed within the Pharaon Italian Pilot to address crucial needs identified by participants, including older adults, informal, and formal caregivers. These needs include: Developing an advanced monitoring Health Service for neurodegenerative diseases. Promoting cognitive and physical stimulation. Integrated with Fiware, the sensors enable data visualization and user management, enhancing the overall functionality and usability of the system.	UNIFI	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Sensors-data-integration-in-Fiware-Unifi
31	Exercise recognition camera UNIFI	The Physical Exercises recognition camera, integrated into the broader IORT Scenario, comprises RGB data for human activity recognition. Participants performed 10 physical exercises for 2 minutes each, with a camera positioned 3 meters away.	UNIFI	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Exercise-recognition-camera-Unifi
32	Instructions for adding technical guidelines	Instructions for adding specific technology guidelines to Pharaon Developer's Handbook	ENT	https://gitlab.com/pharaongroup/developers-handbook/-/wikis/Pharaon-technology-guides/Instructions