SWForum.eu

European forum of the software research community

Deliverable D3.4

Software Forum Research Roadmap v2

Editor(s):	Juncal Alonso
Responsible Partner:	TECNALIA
Status-Version:	Final - v0.1
Date:	31/12/2022
Distribution level (CO, PU):	PU

Project Number:	GA 957044
Project Title:	SWForum.eu

Title of Deliverable:	D3.4 – SWForum Research Roadmap – v2
Due Date of Delivery to the EC	31.12.2022

Workpackage responsible for the Deliverable:	WP3 - Sustainable SWForum.eu and Research & Innovation Roadmaps
Editor(s):	Juncal Alonso (TECNALIA)
Contributor(s):	Elixabete Ostolaza, Begoña Sanchez, Alejandra Ruiz, Jabier Martinez (TECNALIA)
Reviewer(s):	UOXF
Approved by:	All Partners
Recommended/mandatory readers:	WP2,WP3

Abstract:	There will be three iterations of this roadmap. The first iteration will include the scoring methodology valorising the software technology, digital infrastructures and cybersecurity research and innovation roadmap and the necessary steps for its implementation and the initial set of research priorities identified, including also the input from the cross fertilization workshops, the industry, as well as from the desktop research performed and the landscape reports from task 3.1. This roadmap will be made available for the open (online) consultation. The second iteration will contain the final set of prioritized research areas, as well as a set of recommendations and actions on research priorities for Horizon Europe and Digital Europe, which will be presented in different events and workshops in order to gather more feedback. Interim and working versions of the roadmaps will be presented in the cross-fertilization workshops in order to foster discussions and gather input and feedback from the whole software community. The final version will include the received feedback and will be ready for publication. This is the result of Task 3.2.
Keyword List:	Research roadmap, literature analysis, context analysis, research challenges
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)
Disclaimer	http://creativecommons.org/licenses/by-sa/3.0/

Document Description

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	09.06.2022	First TOC	TECNALIA
v0.2	20.09.2022	Contribution to the different sections by the experts.	TECNALIA
v0.3	27.12.2022	Ready for internal review	TECNALIA
v0.4	05.01.2023	Comments from internal review	UOXF
v1.0	13.01.2023	Ready for submission	TECNALIA

Table of Contents

List of Figures

List of Tables

Terms and abbreviations

AI	Artificial Intelligence
CI	Continuous Integration
CSA	Coordination and Support Action
DAST	Dynamic Application Security Testing
DoA	Description of Action
EC	European Commission
GA	Grant Agreement to the project
KPI	Key Performance Indicator
ML	Machine Learning
OSS	Open-source Software
OSH	Open-source Hardware
SAST	Static Application Security Testing
SDLC	Software Development LifeCycle
SOLC	Software Operation LifeCycle
SW	Software
SWForum.eu	European forum of the software research community

Executive Summary

This document is the second version of a series of three documents, whose aim is to provide policymakers with research priorities identified in the field of software technologies, cybersecurity, and digital infrastructure.

To achieve this, a roadmap methodology was defined in D3.3 - Software Forum Research Roadmap v1 [1], structured in three steps, namely identification of research topics, classification, scoring, consultation, and further analysis. The outcome of these activities will be a set of recommendations to be produced as the final result in D3.5. This document focuses on the activities related to the implementation of the second phase of the methodology: classification and scoring of the identified research topics.

To achieve this a scoring methodology has been defined and it has been applied to classify the identified research topics towards the prioritization of the research challenges towards their alignment with the objectives of SWForum and the policy framework under which this action has been funded. Also, the first set of research and innovation challenges has been reviewed and detailed. As a result, a set of lessons learnt have been derived. These lessons learnt will be the initial basis to derive the final policy recommendations that will be produced in D3.5.

The next version of this document will include a reviewed version of the prioritization base on the open consultation to be performed with the SWForum constituency as well as the final set of policy related recommendations.

1 Introduction

1.1 About this deliverable

This document is the second of a series of three documents that aims to present research challenges in the field of software technologies, including open-source software, cybersecurity, and digital infrastructures.

The goal of this second iteration is to present the scoring methodology, the conclusions, and results of the application of the methodology to the identified research and innovation challenges. It is to be said that the first set of challenges identified in D3.1 has served as input to this deliverable. This initial set of challenges have been refined and detailed ending up with the identification of seven challenges and their related research objectives, research outcomes, limitations of current practices and research sub-topics.

This deliverable also details the scoring methodology used to analyse, classify, and score the proposed challenges against different factors towards the implementation of the SWForum action objectives. The goal is to produce a roadmap for the achievement of these challenges at European level. As a result, the deliverable proposes a set of lessons learnt as inspiring measures and actions to be taken forward.

1.2 Document structure

The document is structured as follows.

Section 1 is the introductory chapter.

Section 2 reviews the road mapping approach and methodology followed for the implementation of the research roadmap and details the "classification & scoring" phase which is the focus of the iteration covered by this deliverable.

Section 3 provides an in-depth analysis of the identified research challenges, reviewing the ones proposed in D3.1 and extending them with new identified challenges.

Section 4 presents the initial findings of the application of the scoring methodology described in section 2 to the set of challenges identified.

Section 5 provides the initial set of lessons learnt and actions to be implemented as a result from the analysis performed in section 4.

Section 6 end ups the deliverable with the main conclusions and next steps.

Appendix 1 shows the complete scoring table which has served as basis for the results in section 4 and lesson learnt in section 5.

2 Road mapping approach and methodology

2.1 Road mapping approach

This report is the second version of the SWForum research roadmap, which previous version was release in December 2021 (D3.3 [1]). The final version of the SWForum research roadmap will be delivered in March 2023 (D3.5). As introduced in D3.3, the main aim of this set of reports is to provide contributions to the European research roadmap in software technologies, including open-source software, digital infrastructure, and cybersecurity, taking into consideration different input sources and incorporating the views from different stakeholders such as the industry, independent users and academia.

The methodology followed in SWForum for the road mapping can be summarised in three steps, as seen in Figure 1 and described below. This methodology has already been used in other projects such as HUB4CLOUD [2] for their main purpose, which demonstrates the adequateness, repeatability, and the scientific soundness of the approach.

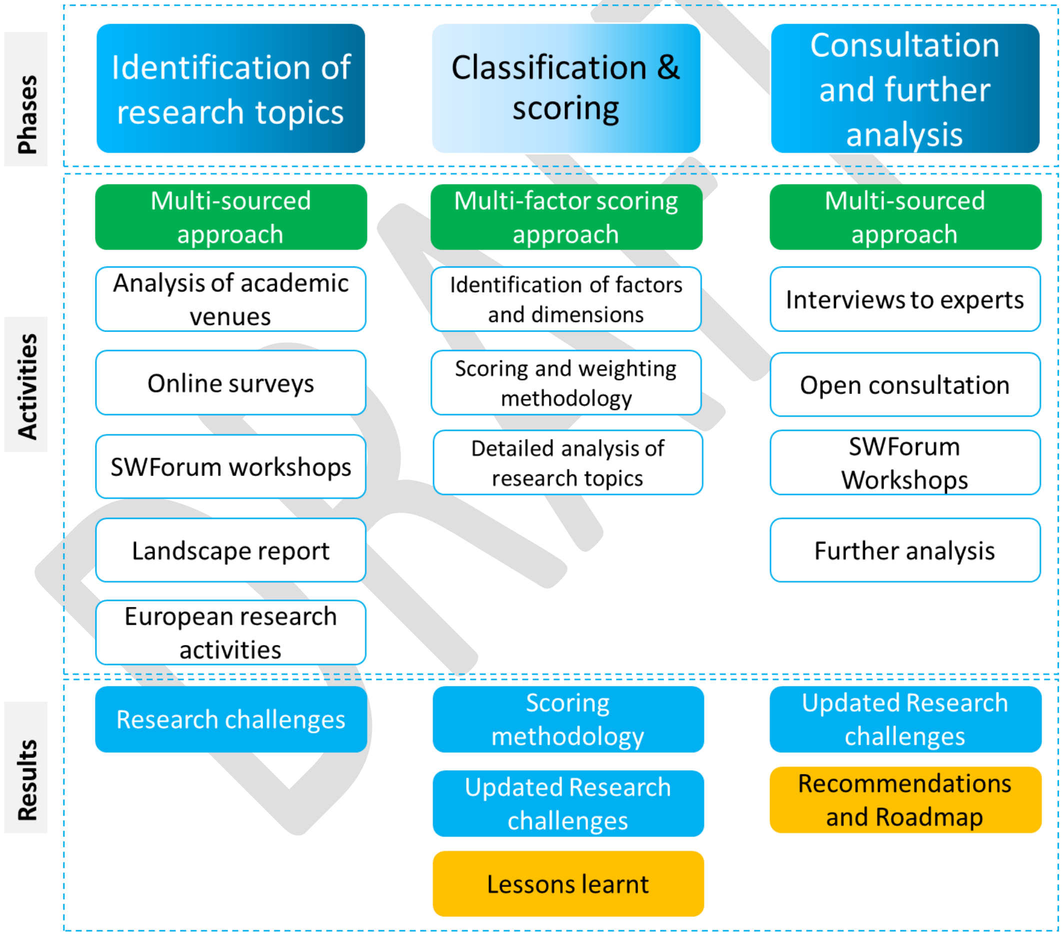
Phases
Activities
Results
Identification of research topics
Classification & scoring
Consultation and further analysis
Multi-sourced approach
Analysis of academic venues
Online surveys
SWForum workshops
Landscape report
European research activities
Multi-factor scoring approach
Identification of factors and dimensions
Scoring and weighting methodology
Detailed analysis of research topics
Multi-sourced approach
Interviews to experts
Open consultation
SWForum Workshops
Further analysis
Research challenges
Scoring methodology
Updated Research challenges
Lessons learnt
Updated Research challenges
Recommendations and Roadmap

Figure 1. Methodology followed in SWForum for the road mapping (adapted and extended from [3]). The current document focuses on the phase "Classification and scoring", hence the different shading in the colours

These three phases were introduced in D3.3, and they cover 1) identification of research topics, 2) classification and scoring and 3) consultation and further analysis.

The focus of this deliverable (D3.4) is Phase 2: Classification and scoring. As shown in figure 2, the activities performed to implement this Phase include:

- Definition of the methodological framework for the scoring approach.
- Identification of the factors, sub-factors, and criteria to assess the identified challenges and related sub-topics.
- Definition of the scoring scale, weights, and process to follow. Actual implementation of the scoring of research challenges by the expert technical team.
- Analysis, graphical representation, and discussion of the results from the scoring including a first set of concluding remarks and lessons learnt.

As shown in figure 2, the main outcome of this Phase 2 includes the scoring methodology, the detailed list of challenges, a preliminary set of concluding remarks, and lessons learnt by analysing the scoring results. The activities described and the outcomes are detailed in the next sections of this deliverable D3.4.

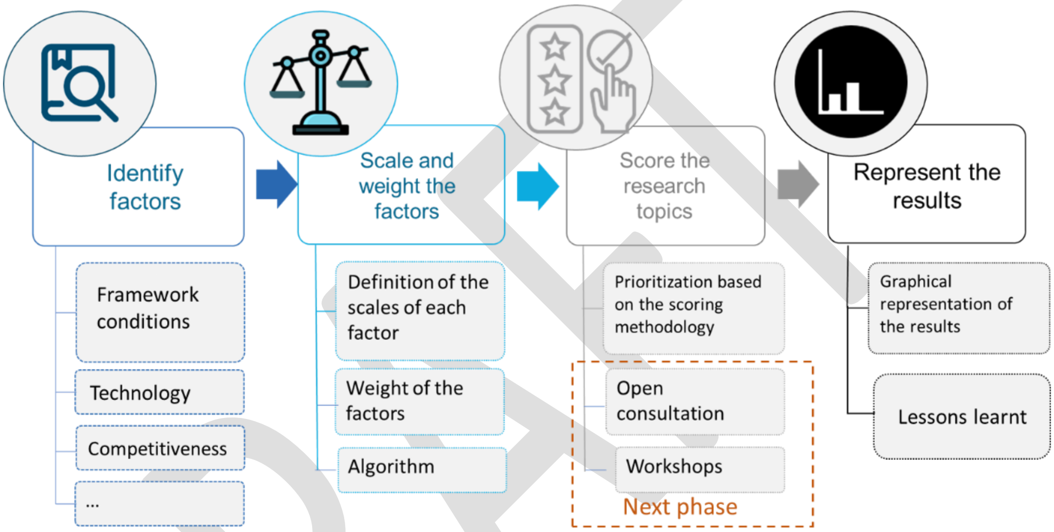
Identify factors
Framework conditions
Technology
Competitiveness
...
Scale and weight the factors
Definition of the scales of each factor
Weight of the factors
Algorithm
Score the research topics
Prioritization based on the scoring methodology
Open consultation
Workshops
Next phase
Represent the results
Graphical representation of the results
Lessons learnt

Figure 2. . Multi-factor scoring methodology phases (adapted and extended from [3])

Phase 3 will be reported in D3.5 and will explain the consultation process preformed to share with the stakeholders through various means such as SWForum workshops, online surveys, interviews, and the final update to the research challenges will be performed and derived into a set of recommended and actions for the European Commission.

2.1.1 Vision and goals

The **vision** is to build "perfect" software systems that are produced and operated "at no cost" as stated in D3.1. "Perfect" means free from faults, secure, resistant, resilient, able to run everywhere, etc. "at no cost" means that the production, deployment, and operation of software systems are optimized and automated with regards of the need of human resources or natural resources (e.g., energy, waste).

The priorities for the research and innovation in each of the knowledge domains can be identified for their contribution to the ideal vision. The challenges to adapt to this purpose are:

- To increase the robustness, security, resilience, and reliability of software products and applications.
- To facilitate the execution over a continuum computing platform.
- To facilitate and automate programming, even by less or non-expert users.
- To allow for the production and exploitation of software almost simultaneously, reducing the delay between production and operation in a continuous life cycle.
- To move towards dynamic, adaptable, and evolutionary forms of execution.
- To adapt systems to the new paradigms that are appearing, such as artificial intelligence and quantum computing.
- European excellence and leadership (TRLs)
- To leverage Open-source based system to the same level of usage, trust and security of proprietary software, with the objective of improving the European sovereignty.

2.2 Scoring methodology

2.2.1 Scoring methodology

2.2.1.1 Methodological framework

A targeted scoring methodology has been prepared by the research team, taking into consideration the project objectives, the expected impacts, and results for the selected challenges. The methodology also includes the perception of the expert researchers on the selected challenges, including the degree of maturity of the technologies/ topics and subtopics as well as the framework conditions regarding Digital strategies and policies at the European Union (EU) level e.g., Digital Decade and Digital compass. These conditions set the framework for the policies and strategies at EU level and consequently, at Member State level.

SWForum.eu intends to raise awareness and strengthen the competitiveness of the European Software Industry including underlying digital infrastructures. Its major objectives are to:

- Promote EU **cross-fertilization** between the areas of software, digital infrastructures, and cybersecurity (Objective 1)
- Create a **self-sustainable forum** of researchers and practitioners in software technologies and related areas, promoting a living forum and Fellowship programme, an online platform as well as a coordinated work to create Research and Innovation Roadmaps (Objective 2)
- **Enhance the visibility** of European-based software technology projects, digital infrastructures and cybersecurity both in the research and in the market domain at an international level e.g., by a taxonomy for the classification of projects, the EU project radar (Objective 3)
- Provide guidance for increasing the **competitiveness of European initiatives** through the definition of a methodological approach to the improvement of their MTRL,

Mentoring, Technology Transfer & Best Practices guiding towards Policy Innovation (Objective 4)

The project has prioritised the following **six challenges**, to identify the main software related topics to be considered in the next years in the context of software technologies, cybersecurity and digital infrastructures as shown in figure 3. All these research and innovation challenges aim at reaching a common vision, to build a "perfect" software system that is produced and operated at no cost.

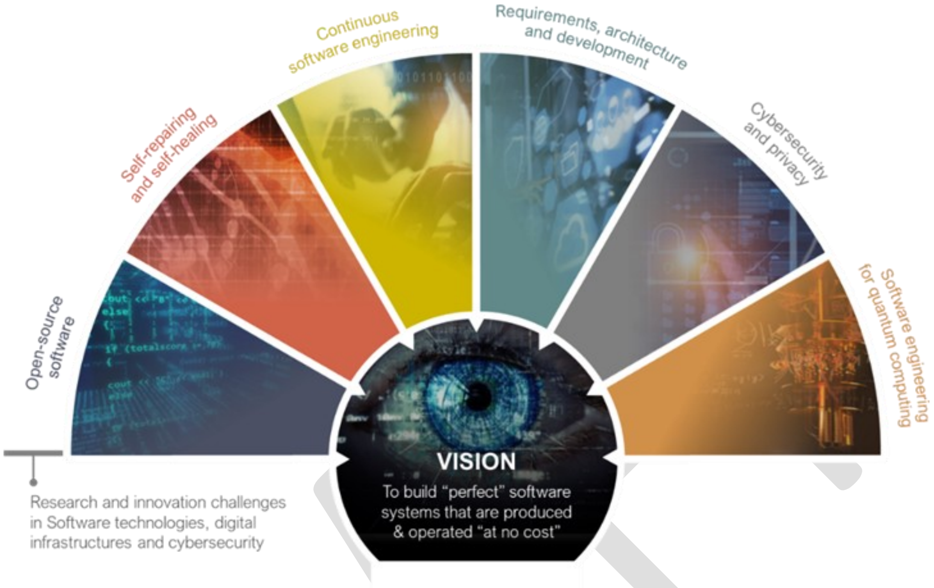
Open-source software
Self-repairing and self-healing
Continuous software engineering
Requirements, architecture and development
Cybersecurity and privacy
Software engineering for quantum computing
VISION
To build "perfect" software systems that are produced & operated "at no cost"
Research and innovation challenges in Software technologies, digital infrastructures and cybersecurity

Figure 3. SWForum.eu challenges. Source. Own elaboration

As **SWForum.eu** is a Coordination and Support Action (CSA), an umbrella project that supports a group of research projects e.g., RIAs and others (see projects in Appendix A). The outcomes from these projects are one of the sources of information to assess the challenges, topics and subtopics of the roadmap against a given number of factors that have been defined to implement the scoring methodology.

A **multi factor and multi scoring methodology** is suggested to evaluate and propose research and innovation topics which are under the umbrella of the projects of this CSA SWForum.eu. The methodology consists of a **matrix** that includes the **six** above mentioned **challenges,** selected in the SWForum.eu project to reach the vision and the **five factors/criteria** defined to target the scope and impact of the SWForum.eu project as presented in figure 4.

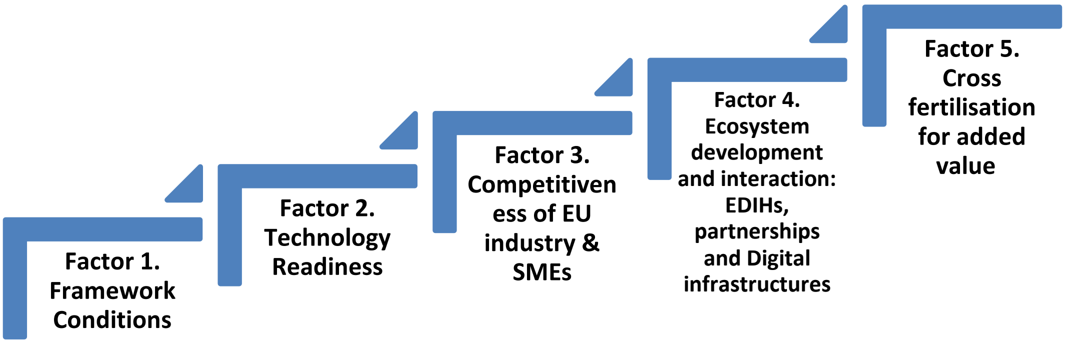
Factor 1. Framework Conditions
Factor 2. Technology Readiness
Factor 3. Competitiveness of EU industry & SMEs
Factor 4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures
Factor 5. Cross fertilisation for added value

Figure 4. SWForum.eu Factors for scoring methodology. Source. Own elaboration

The assessment has been implemented inspired in a **Likert scale type methodology** with a score from 1-5 as shown in Table 1. This methodology has permitted to assess either positively or negatively as well as neutral options. As the technologies subject to this assessment have a novelty dimension regarding their application, some of the assessment parameters are not easy or not possible to assess at this stage, therefore, for these cases a neutral option is very relevant. This Likert type methodology has been detailed per factor in the following section.

Table 1. Likert scale for scoring methodology. Source. Likert scale for scoring methodology [4]

Likert scoring, scale 1-5	
1	Strongly disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly agree

The Methodological process followed for this analysis is represented under the figure 2. The research team has first identified the challenges, topics and subtopics taking into consideration several variables such as, the impacts, limitations to the research, etc. The factors have then been identified along with the criteria, project objectives, expected impacts, added value, etc. This defines the scoring methodology that includes the scoring process as well as the scale and weighting for all those factors.

The research team has worked with the challenges and factors and assessed them following this methodological process. Resulting from this exercise, the values obtained for each sub-topic have been summed up at factor and Challenge level and weighted for each of the challenges based on the weight defined for each factor.

This first **evaluation** has been implemented by a research team in Tecnalia composed of different experts on the topics. After, the **total values per challenge have been weighted**, according to suggested percentages based on an estimation of the relevance of each factor for the achievement of the objectives and impacts of the SWForum.eu project.

These results are planned to be shared with the experts during a **second and third round** of evaluation during different open consultation activities to be performed in the last period of the project. The second round of evaluation will involve a larger group of TECNALIA researchers and the third round of evaluation will open the consultation to external experts from the SWForum.eu constituency.

The final results will be obtained upon the completion of these three rounds.

2.2.1.2 The Factors and criteria to assess the challenges

SWForum.eu project highlights the fact that **software** is encompassing a tremendous amount of diverse application areas, as it runs on top of a large variety of digital infrastructures. Therefore, software introduces significant problems in terms of security and privacy, leading to the need for guaranteeing trustworthiness, self-adaptation, optimisation of behaviour and the like. It is becoming increasingly pervasive and hence, should be combined with other technologies (e.g., cybersecurity, AI) to solve complex problems.

The following five factors have been selected to assess the challenges:

2.2.1.2.1 Factor 1. Framework Conditions

The European Commission has defined a **new framework**[1] with a vision and avenues for Europe's digital transformation by 2030. This refers to the **Digital Decade** as well as the **Digital Compass**.

The Digital Decade aims to empower businesses and people in a human-centred, sustainable, and more prosperous digital future. It is based on the Digital Compass, which sets out digital ambitions for the next decade as shown in the following figures.

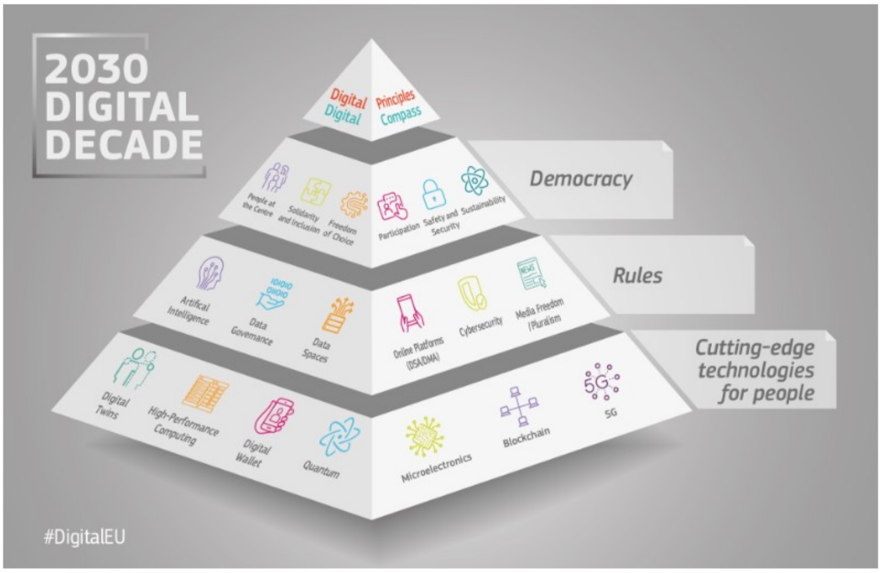

Figure 5. Digital Decade 2030. Source. European Commission

The Digital compass uses the four cardinal points of the compass to identify the main goals:

- A digitally skilled population and highly skilled digital professionals
- Secure and sustainable digital infrastructures

[1] Europe's Digital Decade | Shaping Europe's digital future (europa.eu)

- Digital transformation of businesses
- Digitalisation of public services

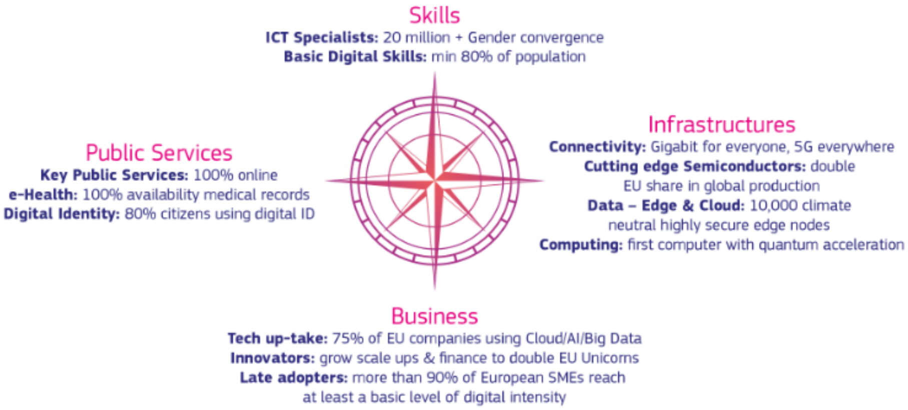
Skills
ICT Specialists: 20 million + Gender convergence
Basic Digital Skills: min 80% of population
Public Services
Key Public Services: 100% online
e-Health: 100% availability medical records
Digital Identity: 80% citizens using digital ID
Infrastructures
Connectivity: Gigabit for everyone, 5G everywhere
Cutting edge Semiconductors: double EU share in global production
Data – Edge & Cloud: 10,000 climate neutral highly secure edge nodes
Computing: first computer with quantum acceleration
Business
Tech up-take: 75% of EU companies using Cloud/AI/Big Data
Innovators: grow scale ups & finance to double EU Unicorns
Late adopters: more than 90% of European SMEs reach at least a basic level of digital intensity

Figure 6. Digital Decade 2030. Source. European Commission

Cloud computing, artificial intelligence, digital identities, data, and connectivity are key policy areas to ensure that the set goals are reached, being all these part of the software ecosystem.

To ensure that these targets are met, cooperation with **Member States** is needed for each of the targets. EU level and national trajectories need to be planned and implemented together with strategic roadmaps to adapt EU actions to national levels.

These elements are therefore, considered in the SWForum.eu scoring methodology. The main aim under this factor 1 is to assess the challenges by means of the following three criteria:

- **Digital Decade 2030.** Potential alignment of the challenge with the Digital Decade principles
- **Digital Compass.** Potential contribution of the challenge to the priorities: skills, public services, Business, infrastructures
- Existence of policies/strategies at **member state level** and/or potential alignment of the challenges with the **national roadmaps.** (This will be assessed in the second round, as there is not enough information available at this stage.)

Factor 1 allows to ensure that the challenges are fully aligned with the European Commission strategies and initiatives and, as a result, with the Member States ones.

2.2.1.2.2 Factor 2. Technology Readiness

The main objective behind this factor is to provide an **expert assessment** on the degree of the technology development and readiness to market for the given challenge, topic, and subtopic. This factor will be assessed against the following two criteria:

- The European based **software technology projects[2] and their outcomes** under the umbrella of the SWForum.eu CSA, previously classified per Challenge based on their

[2] See considered projects in Appendix 1

support or relationship to that challenge. Projects have been classified as fully supporting a challenge (green cells) or partially supporting a challenge (yellow cells).

- The **Best practices** in relation to the challenges. The desk research conducted under each of the challenges, has led to the identification of documents, articles and especially best cases and practices. This has served as a basis for the assessment of this criteria.

For the next evaluation round with experts, the results from the SWForum.eu **Market & Technology Readiness Level** (MTRL) [5] applied to the different project outcomes could be cross-checked with these evaluation results. The MTRL instrument provides industry stakeholders with a compass to guide their innovations towards market uptake. The MTRL is used to evaluate how close to the market the projects are. The MTRL is introduced as a complementary methodology to "Technological Readiness Level" (TRL) to assess the projects outcomes. The radar can be consulted at SWForum.eu Radar | Live radar maps, including the projects in terms of their position in the taxonomy, their maturity, and their MTRL score.

This factor and criteria respond to the scope of **Objective 3** (as described in section 2.1.1)**,** that intends to enhance the visibility of European based software technology projects, digital infrastructures, and cybersecurity both in the research and in the market domain at an international level, as well as **Objective 4**, aiming at providing guidance for increasing the **competitiveness of European initiatives** through their MTRL, Mentoring, Technology Transfer & Best Practices guiding towards Policy Innovation.

2.2.1.2.3 Factor 3. Competitiveness of EU industry & SMEs

It is well known that SMEs are the locomotive of European industry, and this is truer than ever in software related industries, where innovation has consistently been spearheaded by the SMEs in areas ranging from machine vision to new DevOps lifecycle processes. A prime example of that is the agile movement, which was driven by SMEs instead of the larger actors.

This factor is assessed taking into consideration the following criteria:

- The **projects** under the umbrella of the SWForum.eu CSA, previously classified in clusters according to the challenges. In this case, the values have been assigned considering if projects are targeting companies, as they can specially provide funding opportunities e.g., financial support to third parties, vouchers, etc. as well as if companies are part of the consortiums.

This factor is aligned with the main objective of the project, which is to raise awareness and strengthen the competitiveness of the European Software Industry as well as the project expected impacts on industry and the business community.

2.2.1.2.4 Factor 4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures

Factor 4 intends to assess the potential degree of development of the software ecosystem for the given challenge. By the software ecosystem we understand stakeholders cooperating e.g., scientific researchers, providers, developers, operators, policy makers relevant to software technologies, digital infrastructures, and cybersecurity, etc. representing the industry, the government, the universities as well as citizens.

EDIHs are a priority to implement the Digital Programme. They are one-stop shops that support companies dynamically to respond to the digital challenges and become more competitive. EDIHs provide access to technical expertise and experimentation for companies as well as the possibility to 'test before invest' using digital technologies. EDIHs are thus, a key relevant player for the ecosystem development and interaction regarding software. Figure 7 presents the main services provided by an EDIH.

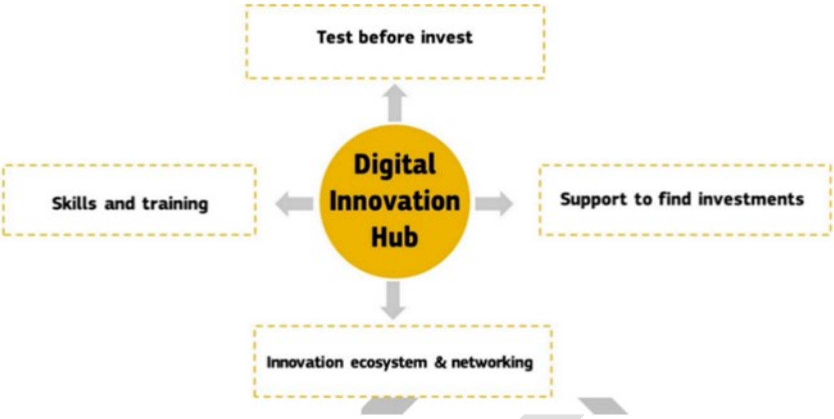
Test before invest
Skills and training
Digital Innovation Hub
Support to find investments
Innovation ecosystem & networking

Figure 7. European Digital Innovation Hub (EDIH). Source. European Commission

There are at least **two strategic platforms/partnerships** at EU level that can also play a role in relation to this project: **ECSO**- European Cyber Security Organisation and **ADRA**- AI Data Robotics Partnerships.

- ECSO – European Cyber Security Organisation (ecs-org.eu). It aims at fostering cooperation between public and private actors at early stages of the Research and Innovation process to allow access to innovative and trustworthy European solutions (ICT products, services, and software). It also aims to stimulate cybersecurity industry, to allow industry to elicit future requirements from end-users, as well as essential sectors. It includes a wide range of actors, from innovative SMEs to producers of components and equipment, critical infrastructure operators and research institutes, brought together under the umbrella of ECSO.
- **ADRA.** Home – Ai Data Robotics Partnership (ai-data-robotics-partnership.eu)**.** The AI, Data and Robotics Partnership is one of the European Partnerships in digital, industry, and space in Horizon Europe. To deliver the greatest benefit to Europe from AI, Data, and Robotics, this Partnership drives innovation, acceptance, and uptake of these technologies.

Regarding Digital infrastructures and Digital capacities, this is an initial list that can be of relevance to SWForum.eu:

- Testing and Experimentation Facilities (manufacturing, Edge AI, health, agri-food, smart communities)- TEFs: Testing and Experimentation Facilities under the Digital Europe Programme | Shaping Europe's digital future (europa.eu)
- Data Spaces Common Support Centre.
- AI on demand platform.
- The European Cybersecurity Competence Centre and Network The European Cybersecurity Competence Centre and Network is now ready to take off | Shaping Europe's digital future (europa.eu).
- The EU Blockchain service Infrastructure EBSI4be.
- European Alliance for Industrial Data, Edge and Cloud

- Open Forum Europe

The Factor has been assessed against the following criteria:

- **Connection to EU digital infrastructures, platforms and EDIHs**. If the projects under the umbrella of SWForum.eu are connected to any of these infrastructures' platforms/partnerships and EDIHs including other relevant ones.
- **Ecosystem development and integration**. Degree of established ecosystem of software engineering, security and digital infrastructure practitioners working together on the particular challenge, topic/subtopic.
- The potentiality for a **self-sustainable forum** of researchers and practitioners in software technologies and related areas. it could be the case that these forums already exist for a given challenge, if not the value responds to the potential of a self-sustainable forum.

This factor is fully in line with **Objective 2**: Create a self-sustainable forum of researchers and practitioners in software technologies and related areas and **Objective 3**, enhancing the visibility of European-based software technology projects, digital infrastructures, and cybersecurity.

2.2.1.2.5 Factor 5. Cross fertilisation for added value

This factor intends to assess the degree and potentiality of cross-fertilisation for the given challenge and generate added value for industry between the areas of software, digital infrastructures, and cybersecurity.

Cross fertilisation in SWForum is understood as facilitating the collaboration of different stakeholders coming from different communities and expertise such as the industry, the public sector or the academia.

The main criteria to assess the challenges are:

- The **projects** under the umbrella of the SWForum.eu CSA, previously classified in clusters according to the challenges have been checked in terms of technologies to assess their potential for cross-fertilisation.

This is fully in line with **Objective 1** to promote EU cross-fertilization between the areas of software, digital infrastructures, and cybersecurity.

2.2.1.3 The scoring scale and process

The scoring process is structured, as already mentioned, in a matrix that sets the **6 challenges** including their respective topics and subtopics as well as the **5 Factors** with the given criteria. Following the 1-5 scale the research team has carefully assessed and prioritised these choices in a logical and objective way. The research team has therefore expressed their level of agreement following these 1- 5 options.

The scoring ponderation has been as follows per factor:

- **Factor 1: Framework Conditions** for the given Challenge. Framework Conditions are assessed following two Criteria:
  - **C1.1. Digital Decade 2030.** The potential alignment of the challenge, topic, and subtopics with the Digital Decade principles is assessed.
  - **C1.2. Digital Compass.** The potential contribution of the challenge, topic, and subtopics to the priorities – skills, public services, business, infrastructures regarding the Digital Compass principles – are assessed and scored.

- An additional criterion has been suggested for the expert's validation, as there is not enough information currently on the existence of policies/strategies at Member State level for all the given challenges.

As for these two criteria, the assessment and scoring follow a similar Likert scale approach as detailed in Table 2.

Table 2. Scale for scoring methodology for factor 1

Scoring values 1-5 **Factor 1. Framework Conditions** **Criteria for assessment:** **C1.1. Alignment with Digital Decade 2030.** **C1.2. Potential contribution to the priorities: skills, public services, Business, infrastructures Digital Compass.**	
1	C1.1. Challenge, topic, and subtopic are not aligned at all with the Digital Decade. C1.2. Challenge, topic and subtopic are not contributing at all to the Digital Compass priorities.
2	C1.1. Challenge, topic and subtopic are not aligned enough with the Digital Decade. C1.2. Challenge, topic and subtopic not contributing enough to the Digital Compass priorities.
3	C1.1. Digital Decade is well-known and respected, but this is not having an impact on the challenge, topic and subtopic. C1.2. Digital Compass priorities are well-known and respected, but this is not having an impact on the challenge, topic and subtopic.
4	C1.1. It is important to know about Digital Decade and respect the principles, and this is having an impact on the challenge, topic and subtopics. C1.2. It is important to know about Digital Compass and respect the principles, and this is having an impact on the challenge, topic and subtopics.
5	C1.1. Challenge, topic and subtopic are fully aligned with the Digital Decade. C1.2. Challenge, topic and subtopic fully aligned with the priorities of the Digital Compass.

- **Factor 2. Technology Readiness** for the given challenge. There are two criteria to assess this second factor: one in relation to EU based software technology projects and the second criteria on the existence of best practices.
  - **C2.1**. The first criteria assessed the **projects** under the umbrella of the SWForum.eu, previously clustered per challenge. The scoring has been made based on existence of European based software technology projects, digital infrastructures, and cybersecurity both in the research and in the market domain at EU level per challenge, topic and subtopics.
  - **C2.2.** As for the second criteria on **Best Practices**, only two options have been considered for the assessment – the existence of reference documents to better understand the challenge, topic, and subtopic or not.

More detail is presented in Table 3. Note that two additional criteria have also been considered of interest to assess this factor with the experts in the open consultation process:

- Existence of Research and Innovation Roadmaps
- MTRL of analysed projects

Table 3. Scale for scoring methodology for factor 2

Scoring values 1 or 5 **Factor 2. Technology Readiness** **Criteria for assessment:** **C2.1. Existence of European based software technology projects, digital infrastructures, and cybersecurity both in the research and in the market domain at EU level.** **C2.2. Existence of Best Practices.**	
1	C2.1. For a given challenge, there is no knowledge on the existence of projects and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics. C2.2. If there are no Best Practices available.
2	C2.1. For a given challenge, there are plans to launch projects and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics.
3	C2.1. For a given challenge, there are no projects and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics.
4	C2.1. For a given challenge, there is one project and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics.
5	C2.1. For a given challenge, there is more than one project and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics. C2.2. If there are Best Practices available.

Source. Own elaboration

- **Factor 3. Competitiveness of EU industry & SMEs** for the given Challenge. There is one criterion to assess this factor:
  - **C3.1.** The influence of the technologies to strengthen the competitiveness of the European Software Industry - including the underlying digital infrastructures together with the needed security mechanisms. This has been checked in all the projects clustered per challenge as detailed in the following Table

Table 4. Scale for scoring methodology for factor 3

Scoring values 1-5 **Factor 3. Competitiveness of EU industry & SMEs.** **Criteria for assessment:** **C 3.1. Influence of the technologies to strengthen the competitiveness of the European Software Ind digital infrastructures together with the needed of security mechanisms.**	
1	C3.1. For a given Challenge, companies are not targeted on the associated projects under the umbrella of SWForum.eu.
2	C3.1. For a given Challenge, companies that are targeted on the associated projects under the umbrella of SWForum.eu are big companies.

3	C 3.1. For a given Challenge, companies involved on the associated projects under the umbrella of SWForum.eu are big companies but the project results target both SMEs and big companies.
4	C3.1. For a given Challenge, there is at least one project under the umbrella of SWForum.eu associated to the topic/subtopic with use cases where SME companies are involved, or in the case that the consortium is formed by big companies, the results address SMEs.
5	C3.1. For a given Challenge, there is more than one project under the umbrella of SWForum.eu associated to the topic/subtopic with use cases where companies are involved, or in the case that the consortium is formed by big companies, the results address SMEs.

- **Factor 4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures** for a given challenge. The three criteria used to assess this challenge are the following:

  - **C4.1.** It assesses the existence of digital infrastructures, platforms, EDIHs, etc. related to the challenge, topic, and subtopic.
  - **C4.2**. It assesses the degree of established ecosystem of software engineering, security, and digital infrastructure practitioners. Here there are only two options – yes or no.
  - **C4.3**. It assesses the potential to create a self-sustainable forum of researchers and practitioners in software technologies and related areas (Researchers, industry representatives, and end users) interested in the future of EU research and innovation actions in the context of software technologies, with an attractive "pull" effect on potential participants in these activities and events and maintain a high level of excellence in the interactions within the Forum. Here there are only two options of assessment – yes or no.

Table 5. Scale for scoring methodology for factor 4

Scoring values 1-5 **Factor 4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures.** **Criteria for assessment:** **C4.1. Connection to digital infrastructures, platforms, EDIHs, etc.** **C4.2. Degree of established ecosystem of software engineering, security and digital infrastructure practitioners.** **C4.3. Potentiality to create a self-sustainable forum of researchers and practitioners in the software technologies and related areas.**	
1	C4.1. There is not enough knowledge on the existence of relevant digital infrastructures, platforms and EDIHs to assess the Challenge, topic/subtopic. C4.2. The ecosystem does not exist. C4.3. There is not enough potentiality to create a self-sustainable forum.
2	C4.1. The challenge, topic/subtopic does not connect with relevant digital infrastructures, platforms and EDIHs.

3	C4.1. For a given challenge, there is no project under the umbrella of SWForum.eu associated to the topic/subtopic that connects with relevant digital infrastructures, platforms and EDIHs.
4	C4.1. For a given challenge, there is one project under the umbrella of SWForum.eu associated to the topic/subtopic that connects with relevant digital infrastructures, platforms and EDIHs.
5	C4.1. For a given challenge, there is more than one project under the umbrella of SWForum.eu associated to the topic/subtopic that connects with relevant digital infrastructures, platforms and EDIHs. C4.2. The ecosystem exists. C4.3. A self-sustainable forum exists.

- **Factor 5. Cross fertilisation for added value for a given challenge.** There is one criterion to assess this factor:
  - **C5.1**: Potentiality of the technology to cross-fertilise with others in view of generating added value for industry between the areas of software, digital infrastructures, and cybersecurity. For the projects grouped under each challenge the assessment given is 5 when cross-fertilisation is targeted by the projects and 1 when there is not such cross fertilisation.

Table 6. Scale for scoring methodology for factor 5

Scoring values 1 or 5 **Factor 5. Cross fertilisation for added value.** **C5.1. Criteria for assessment: Potentiality of the technology to cross-fertilise with others in view of generating added value for industry between the areas of software, digital infrastructures, and cybersecurity**	
1	C5.1. For a given challenge, there are no projects under the umbrella of SWForum.eu associated to the topic/subtopic that cross-fertilise with others in view of generating added value for industry between the areas of software, digital infrastructures, and cybersecurity.
5	C5.1. For a given challenge, there are projects under the umbrella of SWForum.eu associated to the topic/subtopic that cross-fertilise with others in view of generating added value for industry between the areas of software, digital infrastructures, and cybersecurity.

Table 7 presents the template used for the scoring of the different challenges. The complete table with all the scores can be found in Appendix B.

Table 7. Scoring Matrix

Factors (F) and Criteria (C)/ Challenges (Ch)	**F1. Framework conditions**	**F2. Technology Readiness**	**F3. Competitiveness of EU industry & SMEs**	**F4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures**	**F5. Cross fertilisation for added value**
	Criteria **C1.1. Digital Decade 2030.** Potential alignment of the challenge with the Digital Decade principles. **C1.2. Digital Compass.** Potential contribution of the challenge to the priorities: skills, public services, Business, infrastructures.	**Criteria** **C2.1. Projects**: European based software technology projects, digital infrastructures, and cybersecurity both in the research and in the market domain at EU level per Challenge, topic, and subtopics. **C2.2. Best practices.** if Best Practices have been identified or not.	**Criteria** **C3.1. Projects**: Influence of the technologies to strengthen the competitiveness of the European Software Industry - including the underlying digital infrastructures together with the needed security mechanisms.	**Criteria** **C4.1. Connection** to digital infrastructures, platforms, EDIHs, etc. **C4.2. Degree of established ecosystem** of software engineering, security, and digital infrastructure practitioners. **C4.3. Potentiality to create a self-sustainable forum** of researchers and practitioners in software technologies and related areas.	**Criteria** **C5.1. The projects** under the umbrella of the SWForum.eu CSA, previously classified in clusters according to the challenges will be checked in terms of **technologies** to assess their potentiality for cross-fertilisation.
(Scale from 1-5) Weight	(10%)	(20%)	(20%)	(30%)	(20%)
Challenge 1. Open-source software.	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5
Challenge 2: Self-repairing and self-healing: Defect prediction and fault localization using artificial intelligence.	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5
Challenge 3: Continuous software engineering	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5
Challenge 4. Requirements, Architecture and development.	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5
Challenge 5. Cybersecurity and Privacy.	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5
Challenge 6. Quantum software engineering.	C1.1: Values from 1-5 C1.2: Values from 1-5	C2.1. Values from 1-5 C2.2. Values 1 or 5	C3.1. Values from 1-5	C4.1. Values from 1-5 C4.2. Values 1 or 5 C4.3. Values 1 or 5	C5.1. Values 1 or 5

3 Research and innovation challenges in Software technologies, digital infrastructures, and cybersecurity

In D3.3 six initial challenges were identified based on the initial research performed, namely:

1. Open-source software
2. Self-repairing and self-healing: Defect prediction and fault localization using artificial intelligence
3. Continuous software engineering
4. Requirements, Architecture and development
5. Cybersecurity and privacy
6. Specific technology domains

During this iteration an internal review of the topics have been performed by experts in Tecnalia and SWForum. As a result, challenge 6 has evolved to a new one entitled "Software Engineering for Quantum computing" based on latest studies and research performed.

Furthermore, the identified challenges have been elaborated and described including the following topics:

- Research challenge description: Detailed description of the research challenge.
- Research objectives and outcomes: Research objectives and outcomes to be overcome in the next years in the context of open-source software.
- Limitation of current practice.
- Research sub-topics to address the limitations of the current practice and to achieve the proposed goals and objectives.

3.1.1 Challenge 1: Open-source software

3.1.1.1 Research challenge description

Open-source software (OSS) has become a reliable alternative to proprietary software. It is embedded and used in our daily systems as well as in the industrial sector. There are several possibilities that open-source offers to the users. Being free from licensing costs and supported by shared Research and Development and programmers, OSS allows smaller players, with limited financial capacity, to enter the market within home technology services, for which proprietary licence prices have kept profit margins low. Even more, OSS is in essence close to a public service (as seen by the European Commission in open-source software strategy 2020 – 2023[3]): it is public code, which makes it a good use of public money and prevents vendor lock-in, eases the use and reuse of software solutions, everyone can benefit as it is easy and free to modify it. It also helps to solve complex technological problems in a collaborative manner. "Think Open" is the new mindset change promoted by the European commission around OSS. Therefore, and as already introduced in D3.3, OSS has impact at different levels, addressing society in general, industry, and economic development:

- Societal impacts of OSS:
  - Free software broadens access to employment by providing a range of possible knowledge and the means to acquire those skills.
  - Influence of OSS on security in general, and safety.

https://commission.europa.eu/about-european-commission/departments-and-executive-agencies/informatics/open-source-software-strategy_en

    - Security and safety as public goods, OSS is perceived to become relevant in environmental science, but also disaster impact assessment, and energy efficiency.
    - Legislation awareness.
- Business impact: In addition to the impacts on the whole economy and the companies, OSS adoption could increase companies' profitability, savings related to the development of software, improved productivity, shorten time to market and enhance innovation capability.
- Impact in economic development:
    - Cost synergy between OSS and proprietary software in product development and marketing.
    - Competition effects of OSS can have impacts on the price and quality of proprietary software.
    - Common goal is cost reduction, as well as to reduce energy consumption.

3.1.1.2 Research objectives and outcomes

Based on the analysed OSS needs in Europe we have defined a set of research objectives and outcomes to be overcome in the next years in the context of open-source software.

GOAL: Open-source for advanced technologies, quantum, data, AI and machine learning training methods and models to support the European technological sovereignty in some critical technology areas through the usage and knowledge of OSS.
OUTCOME: OSS solutions and development approaches for Artificial Intelligence, Quantum computing, Data management and sharing.

GOAL: Increase the trustworthiness of OSS to make sure that open-source components used in our applications are free from vulnerabilities.
OUTCOME: Automated continuous security testing mechanisms, practices, and approaches for Open-Source Software.

GOAL: Research and proposal of open standardised practices to develop, implement, test and validate OSS along the different phases of the SDLC and SOLC.
OUTCOME: Open standards and legislation adapted to needs and idiosyncrasy of OSS.

GOAL: Investigate and invest on research activities for Open Hardware to create a network of excellence of Open-Source Hardware increasing the collaboration and cross-fertilisation of different stakeholders (researchers, practitioners, industry).
OUTCOME: Mature Open-source Hardware European community

3.1.1.3 Limitations of current practice

- Lack of trained and skilled people.
- Some OSS projects are hardly maintained (more contributors less work).
- License compatibility and integration between proprietary and open-source software
- Evaluation of open-source components to be adopted is difficult (different dimensions to be analysed: i.e., functionality, license, etc.) and this usually increases the reluctance of the adoption of open-source solution.

- Trustworthiness in OSS and OSH has been recognized as one of the major concerns for the adoption of open-source solutions One of these concerns is related to the ability of chips or software components to incorporate backdoors or malware embedded by a bad actor in the supply chain. Also the difficulties on identifying the dependencies of different software pieces (i.e. libraries) affects the trustworthiness of OSS and OSH.
- Lack OSS coding standards. Company policies towards open-source embrace the use of and contribution to software that is crucial for its products. Raising awareness of Open-source possibilities has been identified as an effective policy for increasing economic growth. Open-source assets facilitate making an efficient use of resources and increase industrial competitiveness. But companies also need to adopt standards and thus OSS solutions need to be compliant /part of that standardization process.

3.1.1.4 Proposed research sub-topics

To address the limitations of the current practice and to achieve the proposed goals and objectives, the research sub-topics can be decomposed as follows:

Table 8. Open-Source Software Research sub-topics

Open-Source Software
Open-Source Software for Artificial Intelligence: AI (Artificial Intelligence) has acquired a strategic stature within the vision of European Software Industries, and a multi-pronged set of objectives is evolving based upon not only technological considerations but also policy-oriented considerations such as ethical issues in AI. OSS has the potential to make significant contributions to each of these objectives. Technologically, much of current AI research, and especially the sub-discipline of Machine Learning (ML), makes use of large, mature open-source components (TensorFlow, etc.). ML also makes heavy use of large datasets for training of the neural networks, and here the principles of Open Data can promote the widespread availability of datasets in all sectors of ML application, ranging from language processing to machine vision. Other challenges to be explored for the potential contribution of open approaches personified by OSS are Explainability/Transparency of ML reasoning (especially in critical applications) and Ethics, such as various forms of bias (gender, race, economic). The use of OSS to make available open platforms / testbeds / facilities for testing ML applications is also an important challenge to explore.
Open-Source Hardware and Open-Source Processors: In the last 20 years, Open-source has become pervasive in all the ICT industry. Open-source Software needs to be considered while designing the business strategy of any company in the ICT sector. On the other hand, Open-source hardware is in terms of adoption in a similar position as OSS was a decade from now. The success from Open-source Software has come with the availability of a flexible Open stack from the kernel to the application level. These stacks ensure Digital autonomy to the EU in the layers it covers. The extension of the stack below the kernel in the HTC sector would carry the Digital autonomy to layers where Europe is today not independent. The adoption of the Open-source approach by the processor level will open o asset of threats and opportunities that need to be overcome.
Open-Source Software for Quantum Computing: Due to its incipient stage of development, Quantum Computing is hybrid by nature. It offers a specialized form of computing power which needs to co-exist with classical architectures in the form of hybrid computing approaches. Sustainable Quantum Computing strategies are based on the combination of classical architectures with traditional hardware that access quantum devices as needed. Therefore, interoperability and reusability are key aspects to take into consideration. A hybrid quantum stack, especially one that relies on both cloud and on- premises / private cloud

resources, will require management and orchestration to ensure that programs, experiments, processes, and technologies run smoothly and are interoperable.

In this context, OSS solutions can help to reach this level of compatibility and interoperability. Initial Quantum Computing challenges and how to address them leveraging on OSS practices and solutions, specially focused on:

- OSS middleware for Quantum Computing.
- OSS based interoperable components and libraries for Quantum Computing (i.e., Pennylane libraries)

OSS sustainability and interoperability with privative software: Due to the characteristics of the Open-source communities and projects developed the long-term sustainability is essential to the digital universe. Also, the heterogenous and more and more complex software ecosystems lead with the need of enabling the coexistence of both types of software pieces, proprietary and open solutions. To enable this, several issues need to be overcome.

The means and techniques to ease the selection of OSS components to be incorporated, the methodological approach to incorporate these into the traditional Software Development Life Cycle (SDLC) and Software Operation Lifecycle (SOLC**)**, Continuous integration (CI) and testing, intelligent software package management systems that will enhance robustness and security in software ecosystems are some of the issues to be overcome **[6]**. Integrated Development Environment (IDE) that combines information from different sources through formal and semi-formal models to deliver software project intelligence to shorten the learning curve of software programmers and maintainers and increase their productivity **[7]**.

Trusted and secure Open-Source Software and Open-Source Hardware: To increase adoption and fully embrace Open-source solutions in the European Software industry mechanisms to increase the trustworthiness of OS solutions need to be investigated.

To this end, specific relevant topics need to be considered such as Cybersecurity Certification , Security Assessment for OSS and OSH, who is the author and final responsible of OSS building blocks and therefore who is responsible if there is a security issue / breach**,** OSS and backdoors (purposeful embedded security “holes”), ECSO “made in Europe” label (also companies have to “certify” that there are no “backdoors”), Security standards and certification for OSS

Furthermore, with thousands of OS applications, tools and libraries in use, the failure of an small set of code can have severe consequences for the parent software system. Evolution and maintenance of all the software including OS and proprietary tools is crucial, specially, of critical software. Means to identify, manage, and maintain critical software is crucial to increase the trustworthiness of Open solutions.

Open-source for the Computing Continuum: The adoption of cloud computing by the industry can accelerate the commoditization of open-source software. Several OSS solutions exist to solve specific challenges that can be found in the cloud continuum services stack. Cloud vendors, especially the large ones, are monetizing OSS by integrating it into their own cloud services proprietary derivatives created by them but few release them as open-source or they do it under a very complicated licensing model. End-to-end integration of the whole cloud stack and easier exit cloud switching options horizontally (between cloud services of the same type) and vertically (between layers of the cloud stack) are amongst the most critical challenges that need to be solved for the cloud continuum to be successful. This can only be achieved by making open-source technologies available by each cloud and edge service provider, allowing for portability, and simplifying the switch among cloud continuum services. This calls for communities that wish to create open technologies which are part of larger ecosystems, and which together aim to create full stack solutions.

3.1.2 Challenge 2: Self-repairing and self-healing: Defect prediction and fault localization using artificial intelligence

3.1.2.1 Research challenge description

Bugs are prevalent in software. Software, also mature commercial software, is released and deployed on a regular basis with defects, known and unknown. Many of these defects, including security-related ones, remain unresolved for long periods of time. The business impact of these defects is huge. These bugs can happen for many reasons: faults introduced by missing code, code smells, etc.

To prevent these defects, there is the need to develop tools, methods, and algorithms that would allow an automatic program repair (self-repairing), that is, a software able to translate a specification into a machine-executable activity that would automatically generate a fix for that fault (self-healing). This could be achieved by the development of means for obtaining a semantic representation of programs automatically from the source code, where deep learning algorithms could be later applied. Other options could include the analysis of execution traces of a program running on the test cases in a test suite and defining defect localisation algorithms, as well as machine learning algorithms and techniques that automatically learn and exploit properties of correct code.

It is internationally recognized [8] that many of the automated approaches developed during the previous decade, including model-based software engineering, DevSecOps tools, defect and vulnerability analysis, automated bug fixing, modern code review, and value stream management tools, had the objective of improving software development efficiency and quality. Despite these advances in automation, failures, software security and quality issues, and overspending continue to be the norm.

In order to tackle these challenges, Artificial Intelligence based technologies could be applied with the objective of enhancing the intelligence of the systems to increase their self-repairing and self-healing capabilities. For example, many systems would benefit from the development of tools to help developers avoid, detect, and fix defects as they develop software. A range of techniques that includes safer programming languages, better-designed frameworks, cheap and easy automatically generated tests, and tools that recommend bug fixes will collectively provide better results than relying on any one technique alone. In the next decade, AI approaches will provide an opportunity to rethink how we achieve programming goals, by providing improved capabilities for the elimination of trivial and repetitive mistakes that later become hard to detect and fix.

3.1.2.2 Research objectives and outcomes

Based on the analysed needs, we have defined a set of research objectives and outcomes to be overcome in the next years in the context of open-source software.

GOAL: Increase the resilience, correctness, sustainability, and adaptiveness of software systems through the usage of AI techniques, to improve the efficiency and effectiveness of software engineers in detecting defects and malware in the software and decreasing the human error factor from the software systems.
OUTCOME 1: AI based methods and algorithms to detect faults and no compliances.
OUTCOME 2: AI based trustable defects detection techniques and approaches

GOAL: Improve the adaptiveness of the code to changing environments.

OUTCOME: AI based self-repairing mechanisms and approaches to be adopted and included by software developments in their developments.

GOAL: Assure a sustainable consumption of computing resources.
OUTCOME: AI algorithms **for scheduling and capacity planning.**

GOAL: Achieve the continuous monitoring and sustainment of software systems.
OUTCOME: Trustable monitoring mechanisms for distributed systems.

3.1.2.3 Limitations of current practice

- Centralised monitoring techniques and approaches need to be evolved to monitoring approaches for distributed systems, specific solutions for managing and storing all the runtime data need to be proposed.
- Time spent designing and testing systems continues to be cut short when schedule challenges hit, further jeopardising the quality of the systems developed.
- AI algorithms to detect code smells both at Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) levels need to be proposed.
- Enactment of context sensitive self-healing approaches and techniques need to be investigated and techniques to embed this in the code needs to be proposed.

3.1.2.4 Proposed research sub-topics

To address the limitations of the current practice and to achieve the proposed goals and objectives, the research sub-topics can be decomposed as follows:

Table 9. Self-repairing and self-healing *Research sub-topics*

Self-repairing and self-healing: Defect prediction and fault localisation using artificial intelligence
AI enabled code repair; code error detections, predictions and solution proposals . As the rate of code development rises it is more and more difficult to assess the correctness and trustworthiness of such code using traditional methods based on testing. AI enabled systems for assessing and detecting errors and classifying source code as correct (error-free) or incorrect is required to accompany and help the developers and maintainers of software intensive systems. Furthermore, as the complexity of the code increases the dependency upon other software components is also exponentially scaling. Thus, AI supported models, approaches and methods need to be investigated to assess and detect various source code errors (logic, syntax, semantic, runtime, etc.).
AI enabled execution configuration proposal and supervision. The code used to create set-up and configure the resources aims at automatizing the tasks of DevOps teams with the objective of gaining efficiency and reliability in the creation and management of the infrastructural elements to be used for the execution of the application-based software. The automation level can be improved with the incorporation of AI paradigms to problem modelling, as large-scale optimization or multi-objective problems, characteristics that are typically present in the DevOps environments. With the help of AI, DevOps teams can test, deploy, release, and monitor software more efficiently. AI can also improve automation and support the decision making by quickly identifying issues and mitigation actions. Therefore, several processes from the applications runtime lifecycle can be improved by the in-

corporation of AI techniques, such as data stream analysis, concept drift and anomaly detection.
AI enabled cybersecurity threats detection. Artificial intelligence is becoming a must have when talking about the improvement of the effectiveness of the information security teams. As AI techniques and approaches evolve the impact of the application of AI to the detection and prevention of cyber security threats will continue to increase. New approaches from the static (SAST) and dynamic (DAST) perspectives need to be investigated to increase the ability to analyze and mitigate threats more quickly. . Furthermore, artificial intelligence may assist in the discovery and prioritization of risks and their impact and help on the proposal of the mitigation actions to be proposed, as well as the identification of malware assaults before they occur (prediction) . As a result, even with the possible drawbacks, artificial intelligence will aid to advance cybersecurity and assist businesses in developing a stronger security posture.
AI enabled automatic evolution and adaptation through the Computing Continuum. AI can also contribute to increase the efficiency of teams in charge of SOLC, through the support to strategic decision-making by automating it and reducing the time needed to make decisions. This is of special relevance when talking about critical systems or software intensive systems, where decisions need to be taken at run time in order to prevent the systems from failures that could affect the business continuity. Specially in the computing continuum, with a large variety of elements and variables AI can rise relevant benefits for the automation and decision-making process. Therefore, decentralized monitoring models, multi-objective-based algorithms, dynamic models, federated machine learning approaches need to be investigated to provide self-repairing and self-healing capabilities to software intensive systems and realize the cognitive computing continuum concept.

3.1.3 Challenge 3: Continuous software engineering

3.1.3.1 Research challenge description

Traditional software engineering processes are being revisited through the perspective of new ways to approach the software life-cycle. In particular, there are pressing needs to achieve higher levels of productivity, quality and improved response to changes. This agility is expected to find its cornerstone in a continuous software engineering approach. While several agile methods, DevOps methods, or continuous integration are practices that are already becoming widespread (if not mainstream in many application domains) there are still many challenges to achieve continuous software engineering in its envisioned final shape. The vision is related to minimise the disconnections between disciplines and activities in a way that the frictions and obstacles that hinder the full potential of continuity are removed. Companies approaching this ideal agility will open the door to unprecedent levels of competitiveness.

Delivering the continuous agenda poses several significant challenges which need to be addressed. Continuous concept must be perceived when one considers approaches as Enterprise Agile, DevOps, Leann, Beyond Budgeting, and other similar concepts in Lean Thinking. These philosophies require a holistic and integrated approach across all the activities in the software development lifecycle. As well, it is necessary to be highlighted the need for tighter connection between the various phases of business strategy, development, and execution.

Continuous Software Engineering includes DevOps, Agile software development, continuous deployment, continuous delivery, continuous testing, and continuous integration techniques pertain to the continuous delivery model. Continuous integration refers to the process of adding new code commit to source code. Continuous delivery builds on continuous integration and each code commit is automatically tested at the time it is added. Continuous testing adds manual testing to the continuous delivery model, and continuous deployment adds more automation to the software development process. Besides, the challenge to research more on how to adopt, improve, and implement these approaches, assurance, and re-assurance of systems (e.g., compliance to standards, regulations etc.) has been identified as a major concern preventing to fully leverage the benefits of the continuity of the previous phases.

3.1.3.2 Research objectives and outcomes

Based on the analysed needs we have defined a set of research objectives and outcomes to be overcome in the next years in the context of continuous software engineering.

GOAL: Smart reuse approaches to speed up and improve the quality of assurance and re-assurance processes (e.g., for (re-)certification) for software-intensive systems
OUTCOME: Traditional reuse approaches of assurance assets such as assurance patterns or templates need to be extended with smart capabilities that increase the automation level of their creation or adaptation to an evolved system. This way, assurance patterns' configuration could be automatically recommended or applied, or assurance models can be automatically created or composed based on information from the system or similar systems. For the case of re-assurance, an impact analysis of the change in the system is needed prior to tackle its implications in the assurance assets. AI can be used to refine and optimise the assurance assets.

GOAL: Methods and smart tools for effective and agile co-engineering between experts of different disciplines in software-intensive systems development life-cycles
OUTCOME: Disconnections between disciplines and activities can be minimised leading to earlier identification of issues. Also, co-engineering processes can be dynamically identified, so interaction points can be treated effectively and duplicate work is avoided. This is useful for highly specialised domains of knowledge, skills, and compliance needs, such as the disconnections between safety and security engineering teams, or the software and hardware teams just to give an example. In practice, it is applicable also for the responsible teams of the different system development life-cycle.

GOAL: Smart tools to leverage systems usage by humans as well as runtime information into continuous software engineering.
OUTCOME: Monitoring system usage and automatically understanding the user's interaction and feedback can support decision-making during the continuous development life-cycle for future updates of the system.

GOAL: Customization and iterative optimization of the continuous software engineering paradigm
OUTCOME: Methodological adaptations to specific scenarios (e.g., multi-cloud, quantum, AI engineering) and quantitative measurements and criteria for the optimization of the process.

3.1.3.3 Limitations of current practice

General limitations related to the continuous software engineering challenge are:

- Assurance and re-assurance processes are major bottlenecks to achieve the desired agility.
- Different engineering disciplines are highly specialised and the interaction among them is most of the times delayed in time until they finish their independent analysis, or they

have already created their expected assets. There are missed opportunities to detect inconsistencies or issues earlier that will reduce the overall effort or cost. On one hand, this separation is an organisational and methodological issue, but on the other hand, there is a lack of tools to help identify the suitability of these interaction points.

- Gathering feedback about system usage is challenging. For instance, logging runtime information, or getting and interpreting information from users can require a significant effort.
- Generally, AI support for software engineering practices is still not widely accepted and used. In this sense, its integration into continuous software engineering is still in its infancy despite that the potential benefits it can bring for competitiveness.
- DevOps paradigms need guidelines and automatic recommendation approaches for their adaptation to different organizational and technological scenarios.
- Continuous Software Engineering processes do not use to have quantitative analyses towards its iterative improvement.

3.1.3.4 Proposed research sub-topics

In to address the limitations of current practice and to achieve the proposed goals and objectives, the research sub-topics can be decomposed as follows:

Table 10. Continuous software engineering Research sub-topics

Continuous software engineering
Smart (re-)assurance. In a continuous software engineering where the evolution is a highly desired property, the creation or update of assurance assets (e.g., evidence management with respect to requirements or test scenarios, or showing compliance to certain regulations, certifications, standards etc.) can represent a major bottleneck if no smart approaches are in place to help in their construction and safe evolution.
Co-engineering. Highly specialized teams need to effectively communicate their decisions and the constraints that they might impose to other teams. This need to be in an agile way and as soon as possible to avoid the propagation of issues to the next software engineering phases. Interdisciplinary efforts are challenging because most of the time they do not fully understand each other, so automation and innovative techniques should be developed for the interaction points and interference analysis of teams of different phases (requirements, design, development, operation, testing) or quality assurance aspects (safety, security, privacy, performance expert teams). When humans are part of the decision-making process, effective co-engineering is needed for the continuous software engineering vision.
Leveraging feedback and runtime information. Monitoring of runtime information has been proven useful for systems' self-adaptation and failure diagnostics during the operation phase. However, leveraging this information for early phases of software engineering such as requirements and design in a continuous software engineering approach to update the system in a more agile way is a promising research topic. Similarly, integrating users' feedback and increasing the automation in users feedback analysis in a more seamless way in the continuous software engineering approach (e.g., changes in requirements, design, configuration) is desired.
Optimized DevOps. We should assume that there is no "one-size-fits-all" solution for DevOps or continuous software engineering. Each application scenario has its own peculiarities that need to be acknowledged (e.g., multi-cloud, organizational resources etc.). Besides these application contexts, several internal or external factors to an organization (e.g., dynamic ecosystems) can make that the methodology that was more appropriate is no longer the optimal one. Quantitative approaches should monitor the process and suggest optimizations if found.

3.1.4 Challenge 4: Requirements, architecture and development

3.1.4.1 Research challenge description

Requirement engineering is a cornerstone of the software development lifecycle. Requirements are expressed in natural language, which often leads to misconceptions, especially when trying to create the conceptual and architectural models of said requirements. The use of Natural Language Processing and heuristics could help in deriving conceptual models ensuring thereof a traceability between requirements and the next phases in the development life cycle.

Systems nowadays are getting more complex. Microservices are becoming more popular in certain domains such as the cloud, edge, and distributed computing in general. The migration from traditional architectural models to a more loosely coupled ones, based on microservices pose several challenges, such as how to achieve a proper communication or in performance related issues.

The usage of deep learning in programming languages, abstractions, semantic representations of syntax of programming languages and (supervised) machine learning algorithms could benefit the quality of the code finally delivered, for instance, by facilitating to localise and resolve code smells and faults. The need to release code faster often implies to suffer from technical debt in the mid and long term. Technical debt is caused when code delivery is promoted over the quality of the code, which later will have to be refactored. However technical debt occurs in all phases of the SDLC and different actions and strategies need to be performed in order to alleviate the consequences of such decisions. Some of these can include automation of tests, defect prediction (see challenge 2, section 3.1.3), or the application of continuous engineering practices but there are more.

Due to the disruption of AI in the last years that will impact even more in new coming years, this challenge is linked to the following two disciplines [8]

1. AI-Augmented Software Development. AI augmented software development (that is, AI applied to SDLC) will allow software engineers to easily express the changes they care about, including requirement and design trade-offs and different solution options, and then, trust that automation will correctly resolve most, if not all, of the details at the programming language level. For example, many systems would benefit from the development of tools to help developers avoid, detect, and fix defects as they develop software. A range of techniques that includes safer programming languages, better-designed frameworks, cheap and easy automatically generated tests, and tools that recommend fixes will collectively provide better results than relying on any one technique alone. In the next decade, AI approaches will provide an opportunity to rethink how we achieve programming goals, by providing improved capabilities for the elimination of trivial and repetitive mistakes that later become hard to detect and fix.
   These advances will inevitably drive a re-envisioning of the software development process, with increased intelligence and support to developers. Taking advantage of the data generated through the software development lifecycle will be a beneficial and natural by-product of the process. Consequently, this research area asks the question: What will AI-augmented software development look like in the future?

2. Engineering AI-Enabled Software Systems or AI Engineering. The systems of the future—from smart cities and buildings to defence and transportation systems, to healthcare—will likely incorporate AI elements. Advances in Machine Learning (ML) and the increasing availability of computational power are already resulting in huge investments in systems that aspire to exploit AI. AI-enabled systems, software-reliant systems that

> include data and components that implement AI algorithms mimicking learning and problem solving, have inherently different characteristics than software systems that do not use AI components. These differences are driving academia, industry, and governments to explore the creation of a new discipline of engineering called AI Engineering [9] [8] [10].
> However, AI-enabled systems are, above all, software systems. The development and sustainment of these systems have many parallels with building, deploying, and sustaining software systems. Research programs in software engineering will need to focus on the challenges that AI elements bring to software analysis, design, construction, deployment, maintenance, and evolution.

This challenge will cover AI Engineering as well as AI augmented software development.

Finally, new paradigms such as quantum computing will affect the way in which software is developed, where abstractions for modelling, designing and building. Quantum applications will play a prominent role. For further information on Quantum computing and Software Engineering, see *Challenge 6 - Software Engineering for Quantum computing* below.

Impacts of this challenge:

Societal impact: better quality products and services.

Business impact: better quality products and services that meet customers' needs and behaviours, shorten time to market, cost reduction, faster return on investment (ROI).

Technology impact: shorten development cycles, more maintainable software, trustworthy software

3.1.4.2 Research objectives and outcomes

GOAL: Apply NLP and heuristics to automatically derived or assist programmers in the design of conceptual and architectural models as well as with the traceability of the next SDLC phases.

OUTCOME: Facilitate developers the building of conceptual and architectural models derived from system requirements and specifications as well as assist them in the automatic testing of the next SDLC phases.

GOAL: Improve proper communication and performance issues in loosely coupled microservice based architectures and facilitate the migration from tightly coupled traditional architectures to new microservices based architectures.

OUTCOME: Better performance with proper communication in microservices based architectures.

AI augmented software development discipline:

GOAL: Automating AI relevant software engineering tasks and accelerating the development of reliable automation for engineering applying innovative techniques such as AI (ML/DP/NLP mainly) or the opportunities that quantum computing technologies offer.

OUTCOME: Enable the design, development, and deployment of reliable software by further shifting the attention of humans to the conceptual tasks that computers are not good at and eliminating human error from tasks where computers can help.

Engineering AI-Enabled Software Systems/AI Engineering discipline:

GOAL: Explore what existing software engineering practices can reliably support the development of AI systems and what new software engineering research challenges need to be solved in order to reliably construct AI enabled software systems.
OUTCOME: Best practices and techniques, tools and processes based on it to develop AI components and AI-Enabled Software Systems.

3.1.4.3 Limitations of current practice

General limitations related to requirements, architecture and development overall challenge are:

- How NLP and heuristics apply to requirements engineering & conceptual models.
- Loosely coupled architectures: How to achieve Proper Communication and Performance related issues.
- How to support the migration of traditional architectures to new loosely coupled architectures.
- How to get quality code applying DL (in Abstraction, semantics and syntax of programming languages and supervised ML).

Limitations related to *AI-Augmented Software Development* sub-challenge:

- Developers are expected to be experts in many topics (requirements, architecture, design, programming languages, analytic models, a dizzying array of technologies and frameworks, quality attributes, testing approaches, platforms, and much more). The current software development processes are not the most appropriate to orchestrate these activities and the artifacts created along the way.
- Streamlining artifacts created during the SDLC is a resource intensive challenge. During SDLC many artifacts from requirements specification, design documents, test cases, etc are produced.
- Lack of appropriate tools to manage complex systems (in terms of size, distribution, concurrency, etc).
- Formal methods and model-based approaches are not good to scale beyond limited aspects of the system.
- Time spent designing and testing system takes still too much effort.
- System sustainability and evolution, especially in legacy systems, is highly effort-demanded.
- Conformance to quality standards and intended architectures are not guaranteed as part of the SW development framework or tool chain.

Limitations related to *Engineering AI-Enabled Software Systems* discipline or problems faced applying current engineering practices to the development of AI-enabled systems:

- SW Deployment process (such as agile approach): they do not match with the development of ML/DL models or components.
- Train ML models maybe expensive: challenge focus on self-supervised systems and new techniques needed to address what elements of the system can be self-supervised, how self-supervised elements can work together with self-adaptation elements (in particular with self-adaptive software systems), and the resulting challenges for system monitoring and observability (among other things).
- Platforms that support the integration of ML models to the systems. Traditional techniques are used for it (e.g., MLOps, which applied DevSecOps principles to ML

components). New tools, metrics and analysis to provide relevant information to developers need to be developed.

- Systems that contain AI components cannot be reliable tested. Need of testing and analysis techniques to support testing of AI components and AI-enabled systems.

3.1.4.4 Proposed research sub-topics

Table 11. AI-Augmented Software Development research challenge sub-topics

AI-Augmented Software Development **[8]**
Re-envisioned software development lifecycle. See the progress when AI augmented is applied to the SDLC and focus on clarifying the different roles that humans and AI-augmented tools perform, ranging from AI as a trustworthy assistant to AI completely replacing some tasks.
Identify new forms of evidence of quality. AI generates metadata to efficiently verify or validate code and generate traceable evidence with code.
Automate design, evolution, and analysis tools. To assist developers with evolution and refactoring tasks.
Scale auto code generation and repair. Augment with AI techniques, model-based techniques and formal methods to increase the scope and scale of their applicability.
Collect evidence demonstrating developers' acceptance and efficacity of AI assistance.

The above sub-topics are complemented and detailed by the following ones:

- **NLP/ML/DL applied to system requirements and specification.**
- **Scalable, secure and privacy safe, performant and standarised IoT reference architectures & compliant platforms.** IoT platforms compliant with an ever-improved reference architecture that overcomes the current IoT platform limitations. Apply new paradigms (AI, Quantum, etc.) so that IoT systems can gain in adaptability, task automatization, performance, etc.
  - This sub-topic includes methodologies and toolchains to program IoT and Big-data frameworks as a single system and tune the whole infrastructure towards defining its performance, energy efficiency, security, reliability, and dependability requirements.
- **Apply new paradigms** (e.g., NLP, ML) to define/improve tools and techniques **to migrate traditional architectures to loosely coupled architectures based on microservices**.
- **New and adaptation of existing OT deployment techniques and tools (DevOps, DataOps, DevSecOps,etc) of complex systems and System of Systems**.
- **New Modelling techniques & tools and adaptation of existing modelling techniques & tools to be applied to SDLC and AI-enhanced systems.** Traditional model-driven engineering, model-driven software development, model transformation, etc. should be re-vised to create new modelling techniques and tools and adapt existing ones. On the other hand, AI techniques allow to combine different modeling techniques to get adaptable software systems that operate as human-independent as possible and based on the programmers' intent properties.
- **AI-based automatic alignment between design and code**. ML and search based.
- **Design and analysis methods for AI enabled systems. Uncertainty management**.
- **Predicting program properties based on neuronal networks**. For instance, stablishing a correspondence between code snippets and labels in a natural and effective manner. Based on code paths and other program tokens, derive the semantics associated to each program and the main features of the referred code (frequency of use in operation, etc.).

- **DL/ML/NLP applied to code clone detection, code smell and other programming assistance**. Comparing with the traditional techniques applied to code smell, code clone, etc, DL/ML/NLP already showed promising improvements to add semantic analysis to existing tools (e.g., CCleaner [11] , Code2vec [12]). Further research on applying DL/ML/NLP techniques to existing tools and new tools applying DL/ML/NLP techniques is required.
- **AI based coding combining multiple software engineering artifacts**. NLP and boosting algorithms that turn weak learners into strong learners improving the traceability in the SDLC.
- **AI-based developer tools & AI-assisted development workflow** (In what roles do humans and AI perform most effectively as part of an overall team that produces software of sufficient quality?).
- **AI-based code generation approaches** to take advantage of commonly repeated applications & automated code repair AI, and in particular ML, is good at recognising patterns in huge amounts of data. Success will depend on the ability to identify small, scalable portions of auto code generation and repair problems
- **NLP/ML/DL applied to Open-Source Development environments to facilitate life to developers, programmers, maintainers of open software code** (e.g., project DECODER).

Table 12. Engineering AI-Enabled Software Systems sub-topics

AI-Enabled Software Systems **[8]**
AI-enabled system specification methods. Methods for specifying AI enabled system behavior need to be developed.
Testing practices for AI-enabled systems. Unit, integration, and regression testing practices for AI-enabled systems need to be well understood**.**
Design and analysis methods for AI-enabled system. Key AI-enabled system quality attribute concerns, including explainability, monitorability, reliability, and trust, will need to be supported by architectural patterns, tactics, and analysis methods**.**
Data management in support of AI-enabled systems. Understanding the impact of data on system behavior, data architecting, and change management needs to be well supported by analysis and conformance tools.
Uncertainty management methods. There need to be techniques to model, analyse, and design for uncertainty.
Continuous monitoring and sustainment. AI-systems need to be effectively monitored, self-healed, evolved, and sustained.

3.1.5 Challenge 5: Cybersecurity and privacy

3.1.5.1 Research challenge description

A complex issue, securing all parts still does not imply that the whole system is secure. Security is a responsibility of all, not only technical but also organisational. This problem is increasing as the Edge is becoming more popular as edge devices are more prone to be attacked. A common security framework in Europe that will enable the possibility to automate Security tests might ease the pain. Crucial in the software systems is to assess risks in network and software design, risks in information processing, transmission, and storage; detect, prevent and respond to attacks or system failures; and regularly test and monitor the effectiveness of key systems and procedures. So, to do software security better it is inevitable to "shift left" - conduct security testing from the beginning and throughout the software development life cycle (SDLC). The modelling and verification of task allocation and authorisation constraints have also gained significant interest, particularly in the information systems security field. Task allocation and authorisation constraints represent an important aspect in compliance requirements.

Moreover, a continuous analysis of third-party libraries used for the development of software systems to detect security and privacy issue becomes now more necessary than ever. The use of machine learning techniques for malware detection to detect malicious libraries and applications in large and complex systems, as well as large datasets need to be further evaluated.

Another important issue is related to data transparency[4] and sharing data with others, and more specifically, in terms of formats, interoperability, quality, reliability and licensing.

Impacts of this challenge:

Societal impact: legislation awareness (e.g., GDPR, Cybersecurity Act), more trustworthy software, improved data protection (GDPR), more trust and privacy compliance (GDPR).

Business impact: increased productivity and effectiveness.

Technology impact: standardisation, increased security and safety.

3.1.5.2 *Research objectives and outcomes*

GOAL: Define a European security framework to automate security tests in Sw systems. Ensure that reference architectures for complex systems such as IoT platforms, Data Spaces, etc. are compliant with the European security framework. Deploy the European security framework through DIHs and recently launched EDIHs at European level and adapt the European security framework at national level if required.
OUTCOME: Agree a European security framework and development of compliant tools to automate security tests in software systems or platforms. Reference architectures of complex systems should integrate automate security tests based on the European security framework.

GOAL: Apply new techniques and paradigms (e.g., AI, Quantum Computing) to overcome current limitation of risk assessment tools and techniques and adapt risk assessment methodologies to new software systems such as AI-enhanced systems and to distributed and loosely coupled systems communication networks.
OUTCOME: Risk assessments tools, techniques and methodologies enriched with the application of new techniques and paradigms link to disruptive technologies such as AI, Quantum computing and so on.

GOAL: Apply AI (e.g., ML, DL) to detect, prevent and respond to attacks or system failures in an autonomous way and create share data spaces at European level with relevant data supporting these tasks (detection, prevention, respond and recovery). Apply AI to assist or automate continuous analysis of third-party libraries as well as complex systems to detect security & privacy issues. Apply AI to test and monitor efficiency of key software systems and procedures.
OUTCOME: Solutions at European level and link to worldwide solutions to share identified attacks on time. Facilitate the access of software systems all over the world to these attacks.

[4] Data transparency is both "**the ability to easily access and work with data no matter where they are located or what application created them**" and "the assurance that data being reported are accurate and are coming from the official source."

Enriched existing detection, prevention and respond to attack tools with the possibilities offered by AI application.

GOAL: Modelling and verification of task allocation and authorisation constraints.
OUTCOME: Task allocation and authorization modelling and testing tools and techniques.

3.1.5.3 Limitations of current practice

How to perform in an automatic way security test in software systems? Specially in complex systems that involve edge computing, IoT, cloud solutions, microservices, etc.

There is a need to test in an automatic way the security of software systems: in particular, in the last SDLC phases (verification & validation phases) and in operation (OT).

Security issues at organisational and technical level should be solved for complex systems and data privacy and security ensured in ecosystem infrastructures (data spaces, IoT platforms, etc).

Existing security solutions should improve based on AI technologies and new security tools developed taking advantage of the new paradigms (e.g., AI, Quantum Computing).

AI technologies can also be applied to overcome current limitations and facilitate a vulnerability free code, to improve security risk management solutions, improve network security and so on.

How to define security requirements in formal languages that will automate or assist the generation of secure code in software systems? A development effort should be devoted to formalising the security needs and constraints of software systems and to complete model driven development solutions based on them.

3.1.5.4 Proposed research sub-topics

Table 13. Cybersecurity and privacy Research sub-topics

Cybersecurity and privacy
European reference security frameworks and European security framework compliant platforms (or toward an European security framework): interoperable, trustworthy and adaptive embedded HW/SW platform architecture; Safety and security new paradigms for Software updates (Over The Air Software Updates (OTASU)) in MCCPS (Mixed-Criticality Cyber-Physical Systems) reference architecture based platforms; Safety and security in IoT based on edge -fog-cloud computing environment; security requirements in Sw systems and the maintenance of security requirements in their corresponding micro-services based systems; Big-data security frameworks (for instance, based on the Big Data Security Onion Model of Defence).
Automatically express and manage security requirements in an effective and unambiguous way, such that both engineers and stakeholders have a common understanding of their content. Once these security requirements are unambiguously specified and decomposed, one needs to verify the compliance of the realizations to required security behavior by formal verification and testing for both protection and prevention means.
Security attributes are necessary to be addressed at design level. Since DevOps is promoting frequent software deliveries, verification methods artifacts should be updated in a timely fashion to cope with the pace of the process.
Task allocation and authorization constraints
Holistic design methods and architectures that guarantee non-functional properties "by construction" throughout all phases of the software and system development lifecycle (SDLC).

New and advanced architectures, technologies and methodologies to protect sensitive data in computing continuum (from cloud data centers through fog nodes to end devices).
Vulnerability free code correctors. Develop or improve tools applying AI techniques to generate vulnerability free code patches that can be applied on large scale Sw systems.
Safety verification tools to assess safety in an automatic or semi-automatic way.
Test and monitor efficiency of key systems and procedures.
AI to Detect, prevent and respond to system failure. Tools and techniques applying new technologies such as IA and Quantum Computing to attach detection and system recovery for complex systems.
Continuous analysis of 3rd party libraries to detect security & privacy issues.
Tools to assess security risk assessment in containerised clouds. The main cloud services are offering nowadays the alternative of Containers as a Service (CaaS). To **derive the appropriate policies and methodologies to manage resources in such computing environment several modelling and simulation tools are under development. Their aim is allowing modelling and simulation of containerised clouds** to **optimise the resource management,** battery **consumption, scalability, etc. Those tools should also allow to assess security risks associated to each alternative containerized solution.**
Security in APPs and services (Sw Engineering in SDLC and OT - APPs in operation). Based on App store analysis and user submitted content (information/datasets mined from Apple, Windows, Blackberries, etc.) improve security in APPs and platforms using APPs. The security issues under study are Faults, malware, permissions, plagiarism, privacy and vulnerability. Therefore, the two main line of research in secure APPs are: 1.- Vulnerabilities in APPs and How to write secure Software for APPs. Improve and develop tools to identify vulnerabilities in APPs as well as identify best practices to write secure sw for APPs. 2.- Malicious APPS. Improve and develop tools to identify malicious software. Even if nowadays most of the tools are based on static analysis, dynamic analysis techniques are also interesting to explore. NLP applied to user submitted content (e.g., user-reviews) can be an interesting approach to identify security requirements.
Compliant management frameworks at organisation level. Business Processes (BP) are impacted by industry regulations. Therefore, BP and Compliance Management Frameworks are essential to mitigate litigations risks and even criminal penalties. Compliance should be addressed from the design phase and, formal languages to specify compliance request are the first step. They are usually based on formal reasoning and verification techniques. Tools based on formal compliance request languages, verification techniques grounded on temporal logic, based on semantic repositories and covering as many industries regulation as possible should be part of the future research. In addition, it is desirable that the tools are infrastructure independent.
(Others) Security in MDE (Model Driven Engineering): Intent properties related with security issues and model transformation.
(Cloud) Secure Posture Management: Automatic identification of cloud security issues and compliance risks (enabling continuously certification and policy enforcement).
Development kit for the automatization of IaC-Infrastructure as Code secure solutions for cloud paradigms (i.e., containers) & associated modelling language (e.g., DOML for Piacere project). The solution should be as independent as possible from the different platforms.
Secure Adaptive edge/cloud compute & network continuum over a heterogeneous sparse edge infrastructure to support nextgen applications.

3.1.6 Challenge 6: Software Engineering for Quantum computing

3.1.6.1 Research challenge description

Zhao inspired on classical software engineering defined Quantum Software Engineering as: "Quantum software engineering is the use of sound engineering principles for the development, operation, and maintenance of quantum software and the associated document to obtain economically quantum software that is reliable and works efficiently on quantum computers."[5]

Currently Quantum software is going through an explosion phase, multiple quantum software applications are being researched and experimented with in different sectors and businesses, and at the same time numerous challenges that quantum software development is facing are being identified and will face in the years to come. This situation is being possible thanks to the rapid development of quantum hardware, as well as the possibility that researchers and professionals have of accessing quantum computers through QaaS (Quantum as a Service) services. It is therefore time to pay attention to the research and development of quantum software engineering to solve the challenges that are being identified and thus take advantage of the benefits of quantum computing.

At the same time, the signatories of the Talavera Manifesto [6]recognised at the time of its publication in the year 2020, the rapid increase in awareness of the need for applications based on quantum computing for the resolution of business challenges that are complex to solve with traditional software. In addition, they also observed a great interest in producing quantum software in an industrial and controlled way. As indicated, the need to apply and adapt the knowledge and experience available in relation to software engineering is key to managing the construction of quantum software in an industrial and controlled manner.

3.1.6.2 Research objectives and outcomes

Based on the analysed needs we have defined a set of research objectives and outcomes to be overcome in the next years in the context of open-source software.

GOAL: Develop standards for software development processes in hybrid IT-system. Define some best practices for Quantum Software design. Take into account the lessons learned for the classical software design.

OUTCOME: Agree and standardise on best practices for a systematic design of quantum software. Build an integrated workflow for the software life cycle with hybrid quantum systems **[13]**

GOAL: Agree on a standardised way to define the different levels of the software structure when using hybrid quantum solutions. Improve availability of European quantum hardware in the cloud **[13]**

OUTCOME: Standardise an intermediate representation framework that works across multiple technologies. Integrate quantum computers with HPC systems. **[13]**
Proposed a standardised quantum software architecture interoperable among the different quantum platforms which offers QaaS.

GOAL: Identify methods and tools for validation and verification of quantum software which is uncertain by nature. Ensure the quality of the quantum software solutions.

[5] Quantum software Engineering. Landscapes and horizons. Jianjun Zhao.

[6] The Talavera Manifesto for Quantum Software Engineering and Programming.

OUTCOME: Propose methods and tools to be able to validate and verify quantum software ensuring the quality on its designs and implementations.

3.1.6.3 Limitations of current practice

Currently the design of quantum software requires a deep understanding of quantum computing concepts and operators. Quantum models and methods are difficult to understand by most professionals and academics responsible for distributed systems analysis, architectural design, software development and testing.

Currently, for the implementation of quantum software, approaches such as QaaS (Quantum computing as a Service) are proposed, however, when implementing it, the impossibility of abstracting the service from the architecture in which they are executed is evident. Not only that, but there are proven deficiencies in the real abstractions to express or conceive quantum service architectures, to which must be added a lack of support infrastructure for the execution of quantum services.

The different quantum hardware (QPU) requires specific programming instructions for each device. There is no standard or de facto programming platform or interoperability requirements. Current tools are designed by and for quantum computing experts, focused on the development of specific quantum algorithms, without taking into account the possible use/creation of computational blocks for the construction of potentially reusable hybrid systems.

3.1.6.4 Proposed research sub-topics

In to address the limitations of current practice and to achieve the proposed goals and objectives, the research sub-topics can be decomposed as follows:

Table 1. Quantum software Engineering research subtopics

Quantum software Engineering
Quantum software viability studies. The nature of quantum software is completely different from the nature of classical software, so in order to carry out a feasibility study, it is necessary that the work teams that carry out this study have knowledge of multiple aspects: • The business problem that you want to solve. • The quantum algorithms available for this problem. • The quantum elements to be developed as well as the transformations that need to be carried out on them. • The structure of breakdown of tasks that are necessary to carry out both to carry out an initial pilot and for a quantum software development, implementation, and maintenance project. • The necessary knowledge and skills
Quantum software architecture - **Middleware.** There are several problems that we find when integrating quantum algorithm approaches on quantum processors and how far they are from integrating with modern computing methods and paradigms. On the one hand, currently programming models are very close to physical computing systems, so the differences between the quantum circuit model or the adiabatic model imply differences in the construction of programs and instruction sets**.** There is a need for abstraction of the different hardware technologies so we can focused on the algorithm itself. - **QaaS:** As quantum computers overcome the scalability and reliability limitations of qubits, several vendors are offering computing services in a 'quantum as a service'

model so that researchers can either test new algorithms or build end applications on hybrid systems. We refer to hybrid systems as those in which a final application runs on a classical system, but to solve certain calculations or solve specific problems, so-called quantum computers are used.

Quantum DevOps: Currently, quantum systems are integrated with high-performance computing (HPC) systems following an asymmetric multiprocessor model. This implies that the two systems that are going to cooperate with each other must be programmed independently. Similarly to DevOps discipline, a similar discipline needs to be defined, adapting the lessons learned from classical DevOps to the quantum solutions deployment.

Quantum testing: The testing process on classical software is done in a deterministic environment, while quantum environments require working with uncertainties. Conventional testing techniques are based on measuring variables of running programs. In a quantum computing context, this intervention would mean the collapse of the quantum state, so they are not appropriate techniques for quantum software.

Some of the proposals for solutions to face the present challenges for this discipline this discipline are:

- **Quantum tomography.** Reconstruction of quantum states. This approach involves generating the full spectrum of probabilities of the quantum state we want to measure. The repeated reconstruction of quantum states is known as quantum state tomography and provides an estimate of the quantum state of interest. The tomography process is a very long process, since it involves collecting all the possible options of the quantum state.
- **Adaptation of Classical V&V methods to quantum.** Adapt some of the already well-known classical verifiers to be applied in quantum hybrid solutions. In most of this cases implies to check the quantum part of the system as a blackbox**.**

Quantum software interoperability: Currently, for the implementation of quantum software, approaches such as QaaS (Quantum computing as a Service) are proposed, however, when implementing it, the impossibility of abstracting the service from the architecture in which they are executed becomes clear. The different quantum hardware (QPU) requires specific programming instructions for each device. There is no standard or de facto programming platform. There is a need for interoperability among the different quantum computing platforms so the algorithm implementations in software can be reusable.

4 Results from the analysis and prioritisation of research and innovation challenges

The scoring Methodology shows the following aggregated results per challenge and factor as presented in Table 14.

Table 14. Scoring results

Factors (F) and Challenges (Ch)	F1. Framework conditions	F2. Technology Readiness	F3. Competitiveness of EU industry & SMEs	F4. Ecosystem development and interaction: EDIHs, partnerships and Digital infrastructures	F5. Cross fertilisation for added value	TOTAL WEIGTED
Challenge 1. Open-source software	4,4	2,2	3,2	1,2	1,6	2,2
Challenge 2: Self-repairing and self-healing: Defect prediction and fault localization using artificial intelligence	4	2,4	3	1,3	1	2,1
Challenge 3: Continuous software engineering	3	2,5	3,2	2,6	2,4	1,4
Challenge 4. Requirements, Architecture and development	3,8	3,4	3,1	4,1	3,6	3,5
Challenge 5. Cybersecurity and Privacy	3,8	3,4	3,2	3,6	1,9	3,2
Challenge 6. Quantum software engineering	2,7	1,1	2,2	1	0,8	1,4

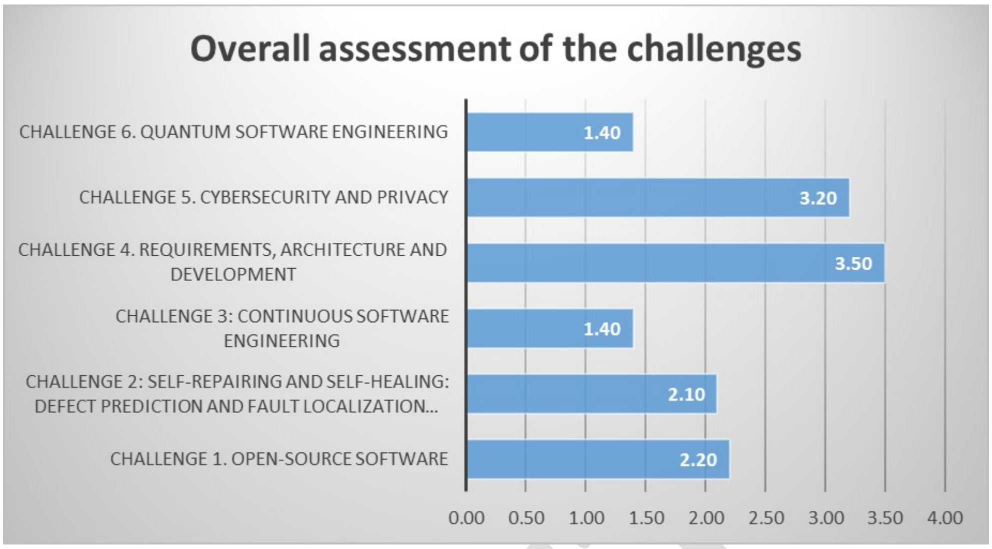

Figure 8. Overall assessment of the challenges.

Figure 2 provides an overview of the overall assessment of the six challenges identified, towards the selected factors. This is the overall weighted assessment as explained in section 2.2.1.3. The figure shows at general level how the different challenges are aligned towards the support of the SWForum objectives and impacts, and which ones have greater opportunity to advance towards the selected factors. To this respect, we can make a distinction between **more mature challenges**, (**CH4 and CH5**) , **intermediate maturity** challenges (**CH1 and CH2**) **and less mature challenges (CH3 and CH6)**. This information provides a general overview of research challenges already supporting the SWForum.eu objectives and the ones that need larger development for this. In the following figures, the analysis per Challenge is shown, including the individual assessment of each challenge towards the specific factors.

- **Challenge 1- Open-Source Software**. Framework conditions, both Digital Decade and Digital Compass are well known and are having an impact on the challenge, topic and subtopics. Open-Source Software is the most aligned challenge with the Framework Conditions of all the considered for the analysis. With respect to the ecosystem, it is true that there exists an ecosystem of Open-source developers, and this is inherent to the open-source concept and the peculiarity of the approach. Nevertheless, the autonomous and diverse characteristics of the ecosystem has ended up in a set of distributed communities that conform the ecosystem and that usually are not grouped under a formal common platform or initiative. With respect to the technology readiness and existence of best practices again, due to the nature of the movement, there is space to work on specific best practices or technology guides covering the whole stack from the physical level (i.e. open-source processors) to the upper layer (i.e., practices and guidelines for specific software technologies like AI, Quantum, Cloud Continuum).

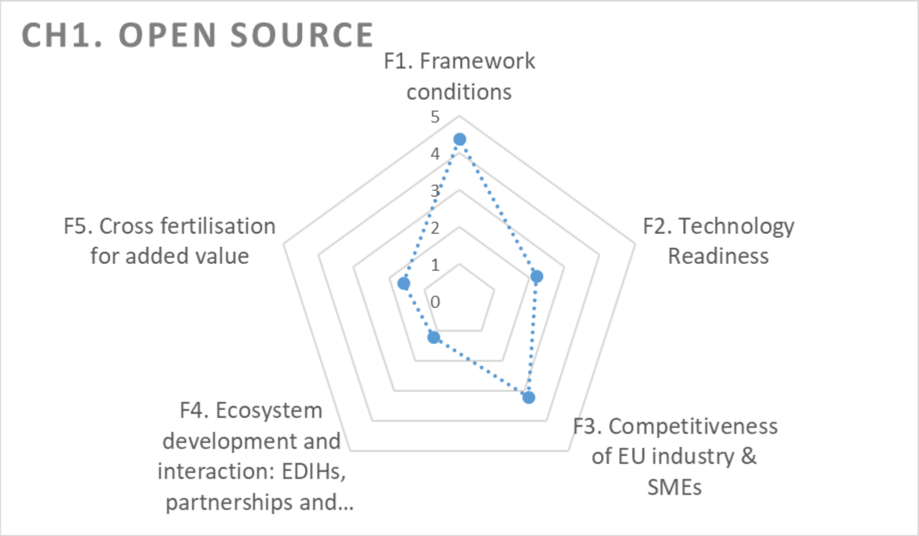

Figure 9. Assessment of CH1 towards the selected factors.

- **For Challenge 2- Self-repairing and self-healing**. Framework conditions, both Digital Decade and Digital Compass are well known and are having an impact on the challenge, topic and subtopics but not fully aligned yet. In the case of CH2, two factors have been assessed with lower value: F4. Ecosystem development and F5. Cross fertilization. In this case, both are related as specific hub and forums need to be developed in specific context of self-repairing software could be developed. The concept of cognitive software can be achieved through the incorporation of AI techniques into the SLDC and SOLC to provide advanced and intelligent capabilities to the software so that it can be more robust, fault tolerant and trustworthy.

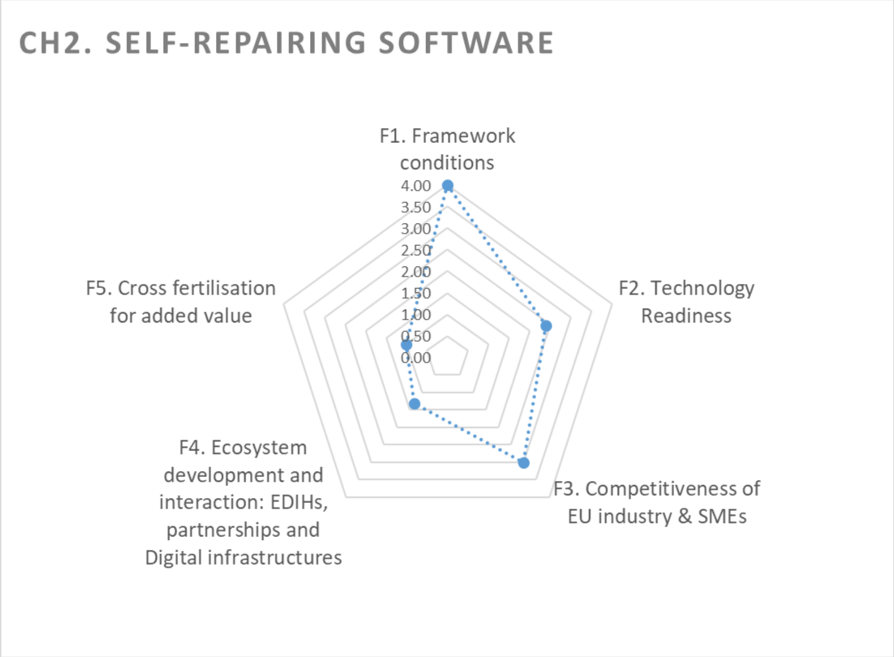

Figure 10. Assessment of CH2 towards the selected factors.

- **For Challenge 3- Continuous software engineering.** Framework conditions, both Digital Decade and Digital Compass are known and respected, but this is not having an impact on the challenge, topic, and subtopics, so it is not fully aligned yet with the Framework conditions. Continuous software engineering is a known discipline into which a lot of effort has been put in the last years. It includes Continuous Development, Continuous Integration, and other related practices that have already been incorporated in the industry. However, these practices need to be revised to be incorporated and adapted to new computing paradigms such as DevOps.

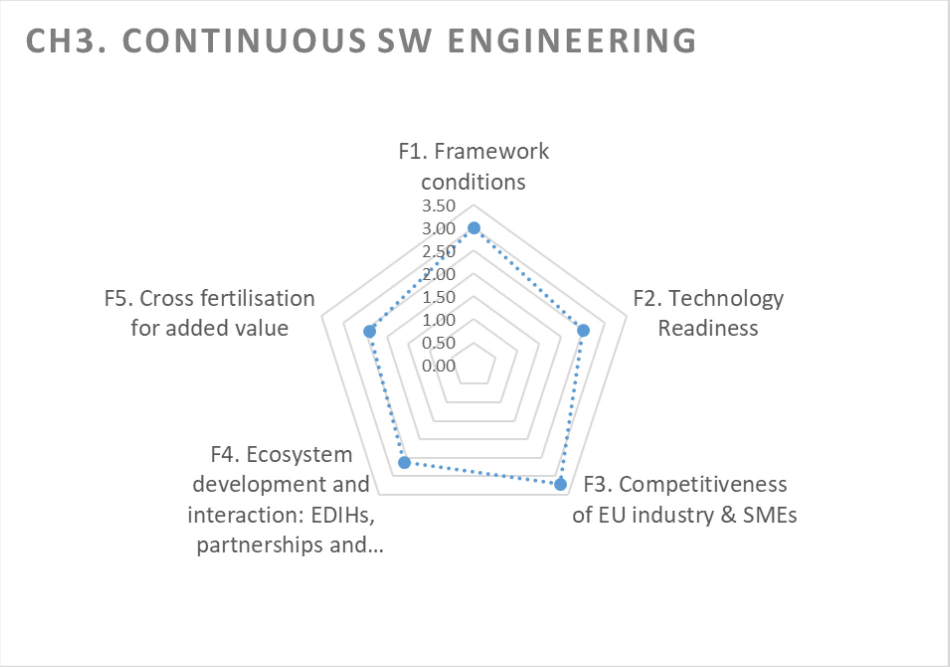

Figure 11. Assessment of CH3 towards the selected factors.

- **For Challenge 4- Requirements, Architecture and development**. Framework conditions, both Digital Decade and Digital Compass are well-known and respected, but this is not having an impact on the challenge, topic and subtopics, so it is not fully aligned yet with the Framework conditions. Similarly, to CH3, CH4 is a well-known Software Engineering discipline which is reflected in the high scores achieved for every factor. Nevertheless, and due to the increased complexity of the software systems, AI techniques can and should be investigated to be incorporated to the early phases of the SDLC, i.e. requirements elicitation, architecture definition and actual development. To this respect, incorporation of techniques and methods from the research to the industry and more concrete alignment with SMEs needs is needed.

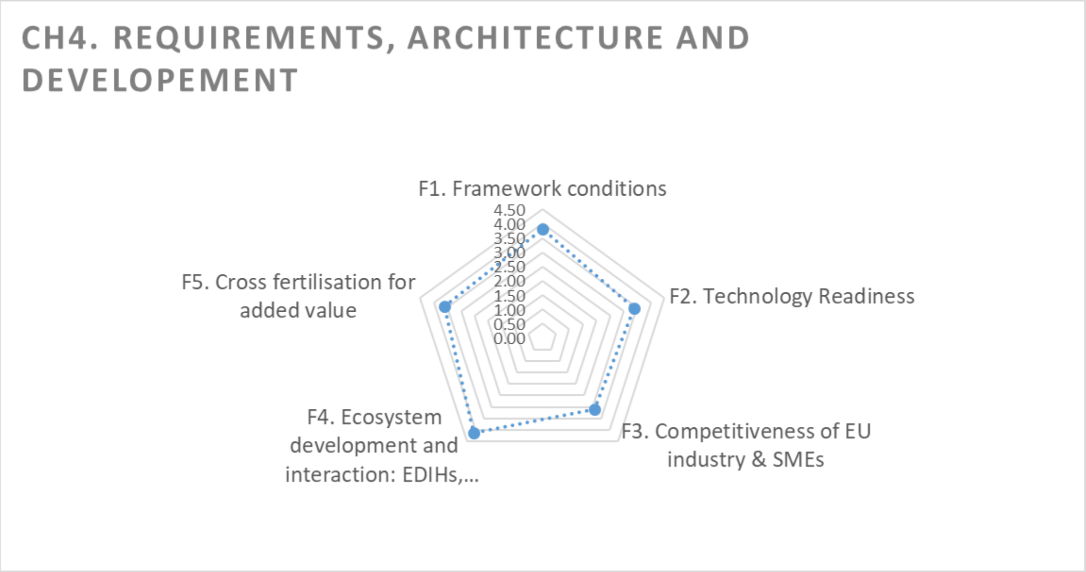

Figure 12. Assessment of CH4 towards the selected factors.

- **For Challenge 5- Cybersecurity and Privacy.** Framework conditions, both Digital Decade and Digital Compass are well-known and respected, but this is not having an impact on the challenge, topic, and subtopics, yet there is not an impact on the challenge topics and subtopics. However, the score is close to achieving it. Although the values are low and there are no projects and or infrastructures under the umbrella of SWForum.eu related to the topics and subtopics nor Good Practices, this is the Challenge which shows better position regarding the technology readiness. Cross fertilization can be developed in the context of Cybersecurity as this topic is transversal to several domains and technologies and currently existing platforms are much more focused and could be extended to cross fertilization with other technologies such as AI or Quantum, as it has begun to happen with other technology topics such as secure data management or Cloud Computing cybersecurity.

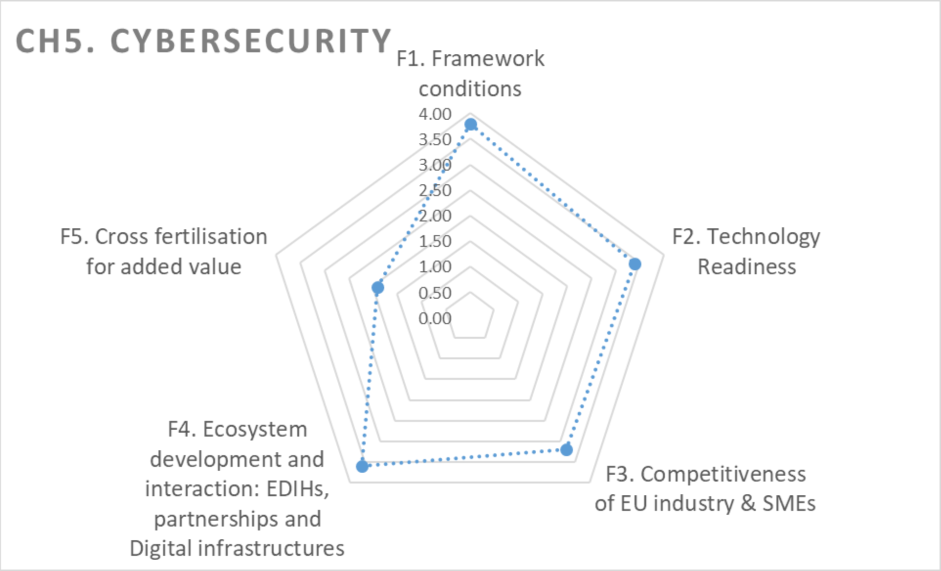

Figure 13. Assessment of CH5 towards the selected factors.

- **For Challenge 6- Quantum software engineering**. Framework conditions, both Digital Decade and Digital Compass are not aligned enough yet with the Framework conditions. Although it is close to be well-known and respected, it is still not having an impact on the challenge, topic, and subtopics. From the point of view of the technology readiness, the challenge shows lack of knowledge on the existence of projects and or infrastructures under and no Best Practices available. This challenge is the one with lower marks in all the factors. This is to be expected as Quantum Computing is one of the most novel recognized cut edge technologies and therefore, the application of software engineering techniques

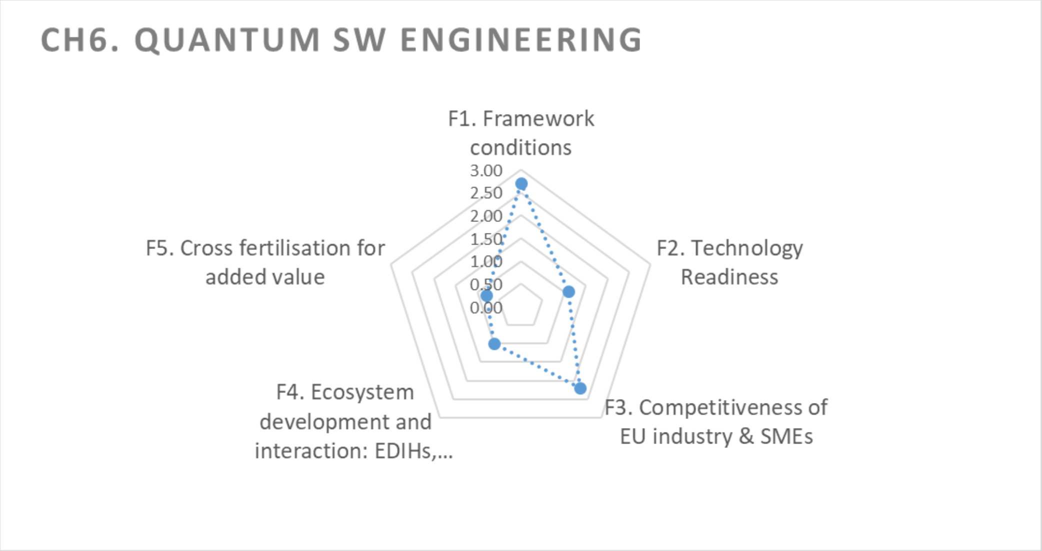

Figure 14. Assessment of CH6 towards the selected factors.

5 Lessons learnt

The following lessons learnt are suggested resulting from the work implemented:

5.1 Lesson learnt 1: Policy alignment for software needs to be fostered

L1. Policy alignment for software needs to be fostered

Although already planned, the Commission and the member states need to be aligned and work together with a **shared strategy for digital transformation and, particularly, for software,** as it is a horizontal discipline **connected to many of the policy areas highlighted in the Digital Decade such as** computing continuum, artificial intelligence, **data governance and data spaces** as well as cybersecurity as well as **cutting-edge technologies** for industry such as Quantum Computing.

Cooperation in legal frameworks is needed to ensure adequate legal and certification frameworks in the field of software for better and faster software development and integration.

Also, make better use of connections, promoting cross fertilisation among different technologies and policy areas regarding software should result in additional added value and impacts at regional, national and EU level.

Synergies between Horizon Europe and other EU/national initiatives could still be improved to provide a more holistic support to respond to the SMEs demand regarding new technologies in the field of software.

The specific analysis implemented in the project, including the scoring Methodology shows that still there is a lack of alignment of the challenges defined in the project with the EU Framework conditions, this is Digital Decade and Digital Compass. Open-source software (Challenge 1) is the most aligned challenge with the Framework Conditions together with Self-repairing and self-healing (Challenge 2) whereas Quantum software engineering (Challenge 6) is the less aligned. This is an indication of the need to better align software as a technology with policy and strategy for better results in some key areas that have further space for development.

5.2 Lesson learnt 2: Targeted awareness for software landscape and use for businesses

L2. Targeted awareness for software landscape and use for businesses

Software as already mentioned is a horizontal discipline that runs on top of a large variety of infrastructures and is becoming increasingly pervasive posing the need to be combined with others. This makes this discipline very challenging for researchers as well as for companies. As complexity increases, software related projects are more challenging for companies as for time constraints, lack of skilled workers, lack of resources and budget for the development processes, for the definition of the project´s requirements, for the complexity of systems and legal frameworks, for security issues at organisational and technical level, as well as for the communication among company developers and customers to mention some.

But on the contrary, the right use of software in combination with other technologies can bring important added value for companies. Therefore, **targeted awareness campaigns** are needed to show the benefits that especially European-based software technology projects can bring.

SWForum.eu has also **reached citizens** in general supporting cutting edge technologies for people by means of the **self-sustainable forum** where researchers and practitioners in software technologies and related areas connect, cooperate and work together to better understand software challenges and come with solutions. This platform allows also to raise awareness, a precondition to support the better understanding of these challenges by society in general.

SWForum **enhances the visibility** of these and support the cross fertilisation of the different technologies covered by those projects. This should be supported as it gives visibility of the opportunities and activities of the EU-software related projects under the umbrella of the CSA and potentially replicated for other similar projects.

5.3 Lesson learnt 3: Skills development programmes & training activities for software need to be promoted

L3. Skills development programmes & training activities for software need to be promoted

Finding qualified skills and reskilling of employees is one of the most important challenges for companies in general regarding digital transformation, and this is not an exception for companies dealing with software. There is not only a lack of qualified workers but also a high cost of hiring qualified people in this specific field.

Software engineering is implemented by many employees and people with interdisciplinary skills, but not specially focusing on software by its nature. Also new trends are seen for its application. an understanding of what is needed by companies is a request to define the training needs as well as the corresponding training programmes that fit the demand.

Although the European Commission is committed to tackle the digital skills gap by supporting projects and strategies to improve the level of digital skills in Europe, and many more initiatives are in place at national and regional level, this should be reinforced, and more strategies need to be in place.

Initiatives such as the European **Digital Skills and Jobs Platform**[7] launched under the Connecting Europe Facility Programme should be promoted for SMEs. It offers information and resources on digital skills, as well as training and funding opportunities. Specific programmes for advanced software engineering experts need to be developed to train the professionals of the future, towards European digital sovereignty and autonomy. To this end, programmes on advanced and cutting-edge technologies need to be incorporated and traditional software engineering programmes need to be renewed with advanced paradigms (i.e., software engineering for advanced technologies (AI, Quantum, etc), self-repairing software, Quantum DevOps, development of open-source technologies, IAOps, etc.).

Standardisation is also needed regarding digital qualifications and training in this field as currently there are only a few official programmes covering such aspects in the field of Software Engineering.

The project also contributes to reduce the skills gap by means of a **Fellowship programme.** One of the main aims of the fellowship programme is the incorporation of young researchers into the SWForum community to dynamize and create discussion around the selected topics relevant to software technologies.

5.4 Lesson learnt 4: Promote the visibility of digital infrastructures, platforms and EDIHs

[7] Digital skills and jobs | Shaping Europe's digital future (europa.eu)

L4. Promote the visibility of digital infrastructures, platforms and EDIHs

The connections and cooperation between the software-related ecosystem, making use of available digital infrastructures that can support businesses to increase their competitiveness is very important.

By the software ecosystem we understand stakeholders cooperating e.g., scientific researchers, providers, developers, operators, policy makers relevant to software technologies, digital infrastructures, and cybersecurity, etc. representing the industry, the government, the universities as well as citizens.

Especially relevant is the connection to the recently launched European Digital Innovation Hubs (EDIHs), which are a priority to implement the Digital Europe Programme. EDIHs support companies to respond to the digital challenges and become more competitive offering services as a one-stop-shop. EDIHs provide access to technical expertise and experimentation for companies as well as the possibility to 'test before invest' using digital technologies. EDIHs target digital technologies such as Artificial Intelligence, Cybersecurity and High-performance computing, but also provide support to create and sustain an innovation ecosystem as well as networking for companies. Companies should approach EDIHs for support.

On top of this EDIHs, the European Networks and platforms are also key to support connections and knowledge in this ecosystem such as **ECSO**- European Cyber Security Organisation and **ADRA**- AI Data Robotics Partnerships as well as relevant Digital infrastructures.

The new Testing and Experimentation Facilities (TEFs): Testing and Experimentation Facilities under the Digital Europe Programme | Shaping Europe's digital future (europa.eu)[8] are expected to be specialised large-scale reference sites open to all technology providers across Europe to test and experiment state-of-the art **AI-based soft-and hardware solutions and products**, including robots, in real-world environments, and at scale. Other relevant challenge is to promote these infrastructures towards the whole ecosystem, overcoming the barrier od industrial usage and adoption of such solutions, specially by the SMEs ecosystem.

Connections and cooperation with these initiatives will support in the **cross-fertilisation** between the areas of software, digital infrastructures, and cybersecurity as well as support and respond to companies' demand.

The analysis of the Ecosystem development and interaction (EDIHs, partnerships and Digital infrastructures) resulting from the scoring methodology shows that Requirements, Architecture and development (challenge 4) is the ecosystem more developed, where more interactions exist and therefore where more potentiality to create a self-sustainable forum. The evidence shows that there is at least one project that connects with relevant digital infrastructures, platforms and EDIHs. Overall, for the other challenges the values of the scoring are very low, so that there is not probably enough knowledge on the existence of relevant digital infrastructures, platforms and EDIHs to assess the Challenge, topic/subtopic, the ecosystem is not ready yet and in conclusion there is not enough potentiality to create a self-sustainable forum.

5.5 Lesson learnt 5: Identification and promotion of good practices

[8] Testing and Experimentation Facilities under the Digital Europe Programme | Shaping Europe's digital future (europa.eu)

L5. Identification and Promotion of Good Practices

Software Industry, businesses and in particular SMEs, need to see the benefits and the added value of digital transformation by means of real examples. The identification and dissemination of good practice examples are a means for it.

Good practices are recommended to be collected broadly showcasing the impacts of the software related technologies in business operations. Presenting software technologies as way to improve the company's productivity and competitiveness.

The SWForum projects can contribute to identify and share good practices of SDLC and SOLC along the SWForum projects**.** This has been done specially through events and webinars organized for and with the projects**.**

5.6 Lesson learnt 6: Advanced Software Engineering related research should be promoted

L6. Advanced Software Engineering related research should be promoted

Nowadays software is pervasive because specific software is needed in almost every industry, in every business, and for every function. Therefore, Software is seen as a commodity to many other domains. Software can help the technological evolution in many other disciplines as a supporting mean to improve and create advance research. This does not preclude the importance of researching in the domain of "software engineering" itself. Software engineering is usually understood as programming, but it involves a much broader concept. The main aim of Software Engineering is to develop reliable models and techniques for producing high quality software in an efficient way, from theory to practice. Therefore, software engineering needs to evolve as new paradigms (computing, social, environmental, technological) arises. As any other scientific domain research on software engineering needs to be promoted to be able to tackle future challenges and advance in the domain.

Research should be promoted for software as a topic working together with industry to better understand their needs as well as government and the software engineering community.

5.7 Lesson learnt 7: Technology readiness regarding software technologies

L7. Technology readiness regarding software technologies

The software related technologies analysed by means of the Challenges overall are not technology ready yet and more research is needed. This has been concluded on the basis of the evidence shown by the scoring methodology as there are not enough European based software technology projects, digital infrastructures and cybersecurity both in the research and in the market domain at EU level as well as Best Practices. This is specially the case for Quantum software engineering (Challenge 6), which values are the lowest followed by and Open-source software (Challenge 1) Self-repairing and self-healing (Challenge 2). The other challenges show values that indicate the lack of projects and or infrastructures as well as the lack of Best Practices that could be used and replicated as learning example.

5.8 Lesson learnt 8: Competitiveness of EU industry & SMEs should be better targeted.

L8. Competitiveness of EU industry & SMEs should be better targeted

The Competitiveness for EU industry has been assessed based on the influence of the technologies to strengthen the competitiveness of the European Software Industry - including the underlying digital infrastructures together with the needed of security mechanisms. Overall, the challenges show that the companies involved on the associated projects under the umbrella of SWForum.eu are big companies, although projects might target SMEs and large companies for their impacts and results. In the case of Quantum software engineering (challenge 6) big companies are targeted on the associated projects under the umbrella of SWForum.eu. Therefore, these technologies are not yet involving SMEs or specially addressing SMEs when looking for impacts or projects. This result is also related to the fact that technologies are not mature/ready enough as indicated under L6.

6 Conclusions

This report has presented the work performed in SWForum.eu focused on the identification and prioritisation of the following **six challenges**, that have permitted to guide the project and assess early-stage technologies as shown in Figure 15. All these research and innovation challenges aim at reaching a common vision to build a “perfect” software system that is produced and operated at no cost.

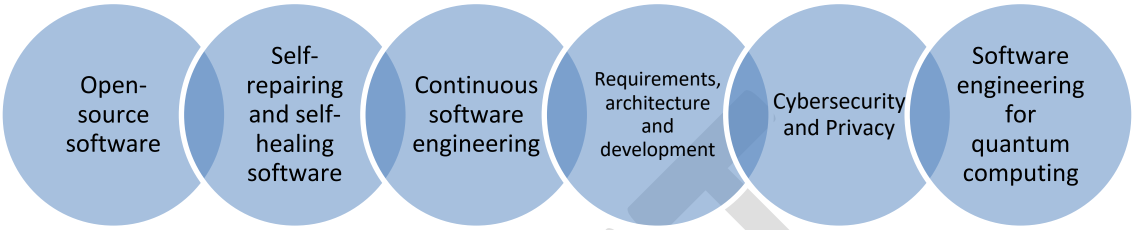
Open-source software
Self-repairing and self-healing software
Continuous software engineering
Requirements, architecture and development
Cybersecurity and Privacy
Software engineering for quantum computing

Figure 15. The SW Forum.eu challenges

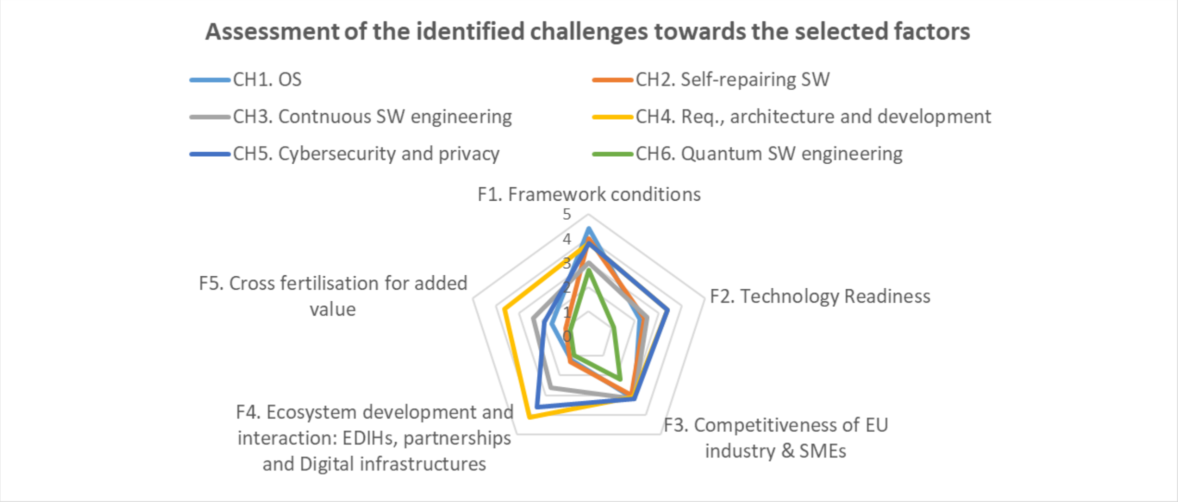

Figure 16. Assessment of the identified challenges towards the selected factors.

We can conclude that all the challenges, topics and subtopics related to each challenge, among others have the following positive impacts but also, some limitations summarised in the following Table. These learnings are the result of the analysis of the projects under the umbrella of SWForum.eu project that the research team has specially worked when implementing the scoring Methodology.

Table 15. Impacts and limitations of the challenges

	Impacts	**Limitations**
Societal	• Broaden access to employment. • Security and safety improvement. • Legislation awareness. • More trustworthy software. • Improved data protection and more trust and privacy compliance (GDPR). • Better quality products and services.	• Lack of adequate legal and certification frameworks. • Lack of trust.
Business	• Better company profitability. • Savings related to the development of software, cost reduction including in energy consumption. • Shorten time to market. • Enhanced innovation capability. • Better quality products and services that meet customers' needs and behaviours. • Faster return on investment (ROI). • Increased productivity and effectiveness. • Product development and marketing.	• Lack of skilled people. • Lack of adequate monitoring systems, targeted solutions. • Business agility. • Complexity of systems. • Lack of appropriate tools to manage complex systems (in terms of size, distribution, concurrency, etc). • Time spent designing and testing system takes still too much effort. • Costs for testing • Security issues at organisational and technical level for complex systems (data spaces, IoT platforms, etc).
Technology	• Shorten development cycles. • More maintainable software. • Trustworthy software. • Standardisation. • Increased security and safety.	• License compatibility and integration. • Trustworthiness and OSH. • Lack of coding standards. • Added value of cross-cutting technologies. • Migration of traditional architectures to new loosely coupled architectures. • System sustainability and evolution. • Conformance of quality standards. • Adequate platforms. • Security issues for complex systems and data privacy and security ensured in ecosystem infrastructures.

This deliverable, the second of a set of three, has as its main goals the identification of research challenges, gaps, and trends, which will finally result in the research roadmaps to be delivered to the European Commission by the end of the project.

The document has introduced the scoring methodology that has been followed for such road mapping starting from the initially identified research topics and providing a detailed description of the second version of those topics as well as an initial scoring of the selected challenges towards a set of identified relevant factors to be considered at European level.

Finally, the initial findings have been reported in terms of:

- Detailed explanation of the identified 6 research topics, including current shortcomings and proposed lines of work.
- Assessment of different identified sub-topics towards 5 relevant factors aligned with the overall SWForum objectives and expected impact.
- Analysis of the assessment results and proposition of lessons learnt.

This initial assessment and prioritization exercise has been implemented with a concrete team of experts belonging to the SWForum project. For the next deliverable (D3.5), the process will be open in form of Open Consultation to external SWForum.eu stakeholders and constituency, deriving in a new version of prioritized topics relevant to Software Technologies and which will be the basis for the final recommendations and road mapping.

7 References

[1] SWForum.eu Consortium, «D3.3- Software Forum Research Roadmap v1,» 2021.

[2] HUB4CLOUD, «The European Cloud Computing Hub to grow a sustainable and comprehensive ecosystem,» [En línea]. Available: https://cordis.europa.eu/project/id/101016673/.

[3] HUB4Cloud consortium, «D1.4 Contributing to the European Cloud Computing Strategic Research and Innovation Agenda,» 2021.

[4] Scribbr, «Likert Scale,» [En línea]. Available: https://www.scribbr.com/methodology/likert-scale/.

[5] SWForum Project, «D4.4-MTRL methodology and assessments v1,» 2022.

[6] FASTEN Consortium, "FASTEN project," [Online]. Available: https://www.fasten-project.eu/.

[7] DECODER Consortium, "DECODER project," [Online]. Available: https://www.decoder-project.eu/.

[8] Carnegie Mellon Institute, «Architecting the future of Software Engineering,» 2021.

[9] J. Bosch, I. Crnkovic y H. H. Olsson, «Engineering AI Systems: A Research Agenda,» *Cornell University ArXiv,* 2020.

[10] P. Santhanam, E. Farchi y V. Pankratius, «Engineering Reliable Deep Learning Systems,» de *AAAI FSS-19: Artificial Intelligence in Government and Public Sector*, Arlington, Virginia, USA, 2019.

[11] L. Li, H. Feng, W. Zhuang y B. R. Na Meng, «CCLearner: A Deep Learning-Based Clone Detection Approach,» de *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2017.

[12] U. Alon, M. Zilberstein, O. Levy y a. E. Yahav, «code2vec: Learning Distributed Representations of Code,» *ACM Programming Languages,* vol. 3, 2019.

[13] QuIC - European Quantum Industry Working Group, «Roadmap (draft),» 2022.

[14] SWForum consortium, "Title 1," *Computer Magazine,* vol. 3, no. 1, p. 2, 2013.

[15] SWForum consortium, "Title 2," in *CloudCom*, Madrid, 2012.

[16] A. Horneman, A. Mellinger y I. Ozkaya, «AI Engineering: 11 Foundational Practices,» Carnegie Mellon University, Software Engineering Institute, 2019.

APPENDIX A: SWForum.eu projects aligned to research and innovation challenges

Initially considered projects for the analysis of their support to the challenges.

Colour code:

🟩 Fully aligned
🟨 Partially aligned

	Challenges					
	1. OPEN-SOURCE SW	2. Self-repairing and self-healing : Defect prediction and fault localization using artificial intelligence	3. Continuous software engineering	4. Requirements, Architecture and development	5. Cybersecurity and privacy	6. QUANTUM software engineering
H2020-ICT-2018-2	🟩 FASTEN https://www.fasten-project.eu/	🟨 FASTEN https://www.fasten-project.eu/				
	🟩 DECODER https://www.decoder-project.eu/		🟨 DECODER https://www.decoder-project.eu/	🟩 DECODER https://www.decoder-project.eu/		
					🟨 UNICORE https://unicore-project.eu/	
			🟩 RADON https://radon-h2020.eu/	🟨 RADON https://radon-h2020.eu/		
H2020-ICT-2020-1			🟩 FOCETA http://www.foceta-project.eu/	🟨 FOCETA http://www.foceta-project.eu/		
			🟨 ELEGANT https://cordis.europa.eu/project/id/957286/es	🟩 ELEGANT https://cordis.europa.eu/project/id/957286/es		
			🟨 XANDAR https://xandar-project.eu/		🟩 XANDAR https://xandar-project.eu/	
			🟨 COSMOS https://www.cosmos-devops.org/	🟨 COSMOS https://www.cosmos-devops.org/		
			🟨 VERIDEVOPS https://sites.mdu.se/veridevops	🟨 VERIDEVOPS https://sites.mdu.se/veridevops	🟩 VERIDEVOPS https://sites.mdu.se/veridevops	
		🟩 PIACERE https://www.piacere-project.eu/	🟩 PIACERE https://www.piacere-project.eu/		🟨 PIACERE https://www.piacere-project.eu/	

H2020-ICT-2019-2						
					RAINBOW https://www.accordion-project.eu/ Fog Computing	
		MORPHEMIC https://www.morphemic.cloud/		MORPHEMIC https://www.morphemic.cloud/		
				PLEDGER http://www.pledger-project.eu/		
					FOGPROTECT https://fogprotect.eu/	

H2020-ICT-2020-1					ASSIST-IoT https://assist-iot.eu/	
					IntellIoT https://intelliot.eu/	
				https://vedliot.eu/ VEDLIOT		
H2020-ICT-2019-2					https://h2020update.eu/ UP2DATE	
				https://teaching-h2020.eu/ TEACHING		
			AMPERE https://h2020-ampere.eu/	AMPERE https://h2020-ampere.eu/		
			CPSoSaware https://cpsosaware.eu/	CPSoSaware https://cpsosaware.eu/		
			ADEPTENESS https://adeptness.eu/			
			1-SWARM SELENE:https://cordis.europa.eu/project/id/871467/es			
		ADMORF http://admorph.eu/				

APPENDIX B: Complete scoring of research and innovation challenges

	SWForum													
	1-Framework conditions:			2-Technology Readiness				3-Competitiveness of EU		4- Ecosystem		5-Cross fertilisation for added value	4-Ecosystem	
TOPICS/Sub-topics	**C1.1-Digital Decade 2030**	**C1.2- Digital Compass**	Existence **of policies/strategies at member state level**	Existence of Research and Innovation Roadmaps	**C2.1- Projects**	**C2.2**- Best practice available	The Market & Technology Readiness Level (MTRL)	**C3.1**- Influence of the technologies to strengthen the competitiveness of the European Software Industry	Expected vertical and horizontal impact of the technology on SMEs	**C4.1**-Impact on the cybersecurity of digital infrastructures	**C4.2**- Degree of established ecosystem	**C5.1**-Potentiality of the technology to cross-fertilice	**C4.3**- Potentiality to create a self-sustainable forum of researchers and practitioners	
CHALLENGE 1 - Open source software and hardware for European Digital Autonomy														
OSH	5	3			3	1		3		1	1	5	1	
OSS for Quantum computing	5	4			3	1		3		1	1	5	1	
OSS sustainability and interoperability with privative software	5	3			4	1		4		1	1	5	1	
Trusted and secure OSS	5	4			4	1		3		1	1	5	1	
Open source for AI	5	4			3	1		3		1	1	1	1	
Open source for Computing continuum	5	5			4	1		3		1	1	5	5	
TOTAL RAW	5	3,83333333	0	0	3,5	1	0	3,16666667	0	1	1	4,333333333	1,666666667	
TOTAL MEAN	Framework conditions				Technology readiness			Competitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
	4,416666667				2,25			3,166666667		1,222222222		1,666666667		2,225
CHALLENGE 2 - Self-repairing and self-healing : Defect prediction and fault localization using artificial intelligence														
AI enabled code repair code error detections predictions and solution proposal	4	4			3	1		3		2	1	1	1	
AI enabled execution configuration proposal and supervision	4	4			4	1		3		2	1	1	1	
AI enabled cybersecurity threats detection	4	4			4	1		3		2	1	1	1	
AI enabled automatic evolution and adaptation	4	4			4	1		3		2	1	1	1	
TOTALS RAW	4	4			3,75	1	0	3	0	2	1	1	1	
CHALLENGE 2	Framework conditions				Technology readiness			Competitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
TOTALS MEAN	4				2,375			3		1,333333333		1		2,075

	1-Framework conditions:			2-Technology Readiness				3-Competitiveness of EU		4- Ecosystem		5-Cross fertilisation for added value	4-Ecosystem	
TOPICS/Sub-topics	**C1.1-Digital Decade 2030**	**C1.2- Digital Compass**	Existence **of policies/strategies at member state level**	Existence of Research and Innovation Roadmaps	**C2.1- Projects**	**C2.2**- Best practice available	The Market & Technology Readiness Level (MTRL)	**C3.1**- Influence of the technologies to strengthen the competitiveness of the European Software Industry	Expected vertical and horizontal impact of the technology on SMEs	**C4.1**-Impact on the cybersecurity of digital infrastructures	**C4.2**- Degree of established ecosystem	**C5.1**-Potentiality of the technology to cross-fertilice	**C4.3**-Potentiality to create a self-sustainable forum of researchers and practitioners	
CHALLENGE 3- Continuous Software Engineering														
Smart (re-)assurance	4	4			5	5		4		4	5	1	5	
Co-engineering	3	5			4	1		4		1	1	5	5	
Leveraging feedback and runtime information	3	4			4	1		4		3	5	1	1	
Optimized DevOps	3	4			4	1		4		3	5	5	1	
TOTALS RAW	2,6	3,4	0	0	3,4	1,6	0	3,2	0	2,2	3,2	2,4	2,4	
	Framework conditions				Technology readiness			Competitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
TOTALS MEAN	3				2,5			3,2		2,6		2,4		2,7
CHALLENGE 6 Software Engineering for Quantum computing														
Quantum software viability s	4	3				1		2			1	1	1	
Quantum software architect	3	4			4	1		3		4	1	5	1	
Quantum testing	1	4				1		2			1	5	1	
Quantum software interope	4	4			3	1		4		3	1	5	1	
TOTALS RAW	2,4	3	0	0	1,4	0,8	0	2,2	0	1,4	0,8	3,2	0,8	
	Framework conditions				Technology readiness			Competitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
TOTALS MEAN	2,7				1,1			2,2		1		0,8		1,39

	SWForum													
	1-Framework conditions:			2-Technology Readiness				3-Competitiveness of EU		4- Ecosystem		5-Cross fertilisation for added value	4-Ecosystem	
TOPICS/Sub-topics	**C1.1-Digital Decade 2030**	**C1.2- Digital Compass**	Existence **of policies/strategies at member state level**	Existence of Research and Innovation Roadmaps	**C2.1- Projects**	**C2.2**- Best practice available	The Market & Technology Readiness Level (MTRL)	**C3.1-** Influence of the technologies to strengthen the competitiveness of the European Software Industry	Expected vertical and horizontal impact of the technology on SMEs	**C4.1**-Impact on the cybersecurity of digital infrastructures	**C4.2**- Degree of established ecosystem	**C5.1**-Potentiality of the technology to cross-fertilice	**C4.3**-Potentiality to create a self-sustainable forum of researchers and practitioners	
CHALLENGE 4 - Requirements, Architecture and development. SE in AI-enabled systems														
NLP/ML/DL applied to system requirements and specification	3	4			3	1		3		4	5	1	1	
Scalable, secure and privacy safe, performant and standarized IoT reference architectures & compliAnt platforms.	4	4			5	5		4		4	5	5	5	
New and adaptation of existing OT deployment techniques and tools (DevOps, DataOps, DevSecOps,etc) of complex systems and System of Systems.	3	3			5	1		3		4	5	5	1	
New Modelling techniques & tools and adaptation of existing modelling techniques & tools to be applied to SDLC and AI-enhanced systems.	3	4			3	5		3		3	5	1	1	
AI-based automatic alignment between design and code.	3	3			3	1		3		4	5	1	1	
Design and analysis methods for AI enabled systems; Uncertainty management methods	3	3			3	1		3		3	5	1	5	
Predicting program properties based on neuronal networks.	3	3			3	5		3		4	5	1	1	
DL/ML/NLP applied to code clone detection, code smell and other programming assistence .	3	3			3	5		3		4	5	1	5	
AI based coding combining multiple software engineering artifacts.	3	3			3	1		3		4	5	1	5	
AI-based developper tools&AI-assisted development workflow	3	4			3	5		3		4	5	1	5	
AI-based code generation approaches to take advantage of commonly repeated applications & **Automated code repair** AI, and in particular ML, is good at recognizing patterns in huge amounts of data.	3	4			3	1		3		4	5	1	5	
NLP/ML/DL applied to Open Source Development environments	3	4			4	1		4		4	5	5	5	
Engineering AI-Enabled Software Systems	3	4			3	5		3		4	5	1	5	
AI-Augmented Software Development	3	4			3	5		3		4	5	1	5	
TOTALS RAW	3,07142857	3,57142857			3,35714286	3	0	3,14285714	0	3,85714286	5	1,857142857	3,571428571	
	Framework conditions				Technology readiness			ompetitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
TOTALS MEAN	3,321428571				3,178571429			3,142857143		4,142857143		3,571428571		3,553571429

	SWForum													
	1-Framework conditions:			2-Technology Readiness				3-Competitiveness of EU		4- Ecosystem		5-Cross fertilisation for added value	4-Ecosystem	
TOPICS/Sub-topics	C1.1-Digital Decade 2030	C1.2- Digital Compass	Existence of policies/strategies at member state level	Existence of Research and Innovation Roadmaps	C2.1-Projects	C2.2- Best practice available	The Market & Technology Readiness Level (MTRL)	C3.1- Influence of the technologies to strengthen the competitivenness of the European Software Industry	Expected vertical and horizontal impact of the technology on SMEs	C4.1-Impact on the cybersecurity of digital infrastructures	C4.2- Degree of established ecosystem	C5.1-Potentiality of the technology to cross-fertilice	C4.3-Potentiality to create a self-sustainable forum of researchers and practitioners	
Challenge - 5 Cybersecurity and Privacy														
European reference security frameworks to automate (or assist) security tests and European security framework compliant platforms	4	4			5	5		3		4	5	5	5	
Automatically express and manage security requirements in an effective and unambiguous way, Task allocation and authorization constraints represent an important aspect in compliance requirements.	4	4			4	1		3		4	5	5	1	
Holistic design methods and architectures that guarantee non-functional properties "by construction" throughout all phases of the SDLC	4	4			4	1		3		4	5	5	1	
New and advanced architectures, technologies and methodologies to protect sensitive data in computing continium	4	4			4	1		3		4	5	5	1	
Vulnerability free code correctors.	3	4			3	5		4		4	5	1	1	
Safety verification tools to assess safety in a automatic or semi-automatic way. Test&monitor efficiency of key systems & procedures	4	3			3	5		3		4	5	1	1	
AI to Detect, prevent&respond to system failure. Continuos analysis of 3rd party libraries to detect security & privacy issues.	4	4			3	1		4		4	5	1	1	
Tools to assess security risk assessment in containerized clouds.	4	3			4	5		3		4	5	1	1	
Security in APPs and services (Sw Engineering in SDLC and OT - APPs in operation).	4	4			5	5		3		4	5	1	1	
Compliant management frameworks at organization level.	4	4			3	5		4		4	5	1	5	
Others: .- Security in MDE (Model Drive Engineering): Intent properties related with security issues and model transformation.	3	3			3	5		3		4	5	1	1	
(Cloud) Secure Posture Management	4	3			3	1		3		4	5	1	1	
Development kit for the automatization of IaC-Infrastructure as Code secure solucions for cloud paradigms (i.e. containers)	4	4			5	1		3		3	5	5	1	
Secure Adaptive edge/cloud compute & network continuum over a heterogeneous sparse edge infrastructure	4	4			4	1		3		4	5	5	5	
TOTALS RAW	3,85714286	3,71428571			3,78571429	3	0	3,21428571	0	3,92857143	5	2,714285714	1,857142857	
	Framework conditions				Technology readiness			Competitiveness		Ecosystem		Cross fertilization		TOTAL Weighted
TOTALS MEAN	3,785714286				3,392857143			3,214285714		3,595238095		1,857142857		3,15