Baltic J. Modern Computing, Vol. 14 (2026), No. 1, pp. 94–120
https://doi.org/10.22364/bjmc.2026.14.1.05

An Approach for Engineering Software Ecosystems Based on an Industrial Case Study

Daryna PYSHCHYK, Kuldar TAVETER, Mohamad GHARIB

University of Tartu, Estonia

{darina.pishchik,kuldar.taveter,mohamad.gharib}@ut.ee

ORCID 0009-0008-1408-1880, ORCID 0000-0003-3074-7618, ORCID 0000-0003-2286-2819

Abstract. Software Ecosystems (SECOs) represent dynamic and evolving networks of interdependent platforms, components, and services developed through collaborative efforts. Engineering such ecosystems presents unique challenges that traditional software engineering methods are often ill-equipped to handle, particularly given the increasing importance of human actors within modern SECOs. To address this gap, this study introduces a novel, five-stage, human-centered approach for engineering Software Ecosystems (SECOs). Developed by integrating established SECO best practices with empirical insights from the H2020 PHArA-ON (Pilots for Healthy and Active Ageing in Europe) project, our approach is consistent with ISO 9241-210 guidelines. It comprehensively covers the SECO lifecycle—from context-of-use analysis and requirements specification to design, implementation, and continuous evaluation. The approach's relevance and utility were evaluated and affirmed through expert interviews, which also provided valuable feedback for its refinement.

Keywords: Software Ecosystems, SECO, Human-centred design, ISO 9241-210, Industrial case study

1 Introduction

Originally inspired by ecological systems, the concept of a Software Ecosystem (SECO) was introduced in 2005 as a collection of software products exhibiting symbiotic relationships Manikas and Hansen (2013). The SECO concept was subsequently evolved to be defined as interconnected networks of software platforms, components, and services that operate within a shared environment, fostering collaboration and value co-creation Bosch (2009); Hanssen and Dybå (2012). SECOs promote innovation by enabling external developers to contribute components and services, thereby driving the ecosystem's continuous evolution Bosch (2009); Hedges and Furda (2018). They also support seamless integration and interaction among stakeholders Bosch (2009);

Hanssen and Dybå (2012). Prominent examples include iOS, Google Play, and Salesforce.

SECOs are typically composed of actors (keystone players, niche players, and users), artefacts, services, transactions, and relationships Hanssen and Dybå (2012); Axelsson and Skoglund (2016), and they have evolved to encompasses social and business dimensions, forming decentralized networks of organizations, individuals, and software entities Hanssen and Dybå (2012); Hedges and Furda (2018). This socio-technical nature of modern SECO introduces significant complexity as knowledge is widely distributed among its stakeholders/actors, and no single entity has a complete view of the SECO Axelsson and Skoglund (2016). This complexity surpasses the capabilities of traditional software engineering methods.

Although SECOs have gained considerable attention across various industries Axelsson and Skoglund (2016), significant research gaps persist Manikas and Hansen (2013); Axelsson and Skoglund (2016); Wulfert et al. (2022); Fotrousi et al. (2014); Serebrenik and Mens (2015). Existing studies often focus on isolated components or specific phases of the development lifecycle, such as requirements engineering or architectural design (e.g., Knauss et al. (2014); Chirumamilla et al. (2018); Knodel and Manikas (2016)). Moreover, many of these studies either lack practical validation or are tailored to specific domains (e.g., Wulfert et al. (2022); Gharib et al. (2024)). Current research emphasizes the need to address these limitations through robust architectural design, effective governance mechanisms, and stakeholder alignment Hedges and Furda (2018); Fotrousi et al. (2014). As a result, the development of a comprehensive, domain-independent SECO engineering approach that systematically integrates human-centred design principles remains a pressing and open research challenge.

This paper proposes a five-stage human-centered approach for SECO engineering[1]. While there are several applicable standards—such as ISO/IEC 12207:2017 ISO 12207:2017 (2017) (software lifecycle processes) and ISO/IEC 15288:2023 ISO 15288:2023 (2023) (system lifecycle processes)—that could inform the engineering of SECO, we required a dedicated human-centred design (HCD) framework that systematically integrates users and stakeholders into every phase of development. Consequently, the approach has been constructed using the ISO 9241-210 ISO 9241-210 (2010)[2] as a baseline, integrating established SECO best practices with insights gained from the PHArA-ON project[3] into each of its phases. The approach was evaluated through expert interviews that aimed to assess its clarity, practical alignment, usability, and adoption potential.

The remainder of this paper is structured as follows. Section 2 outlines the research baseline, including the foundational concepts of Software Ecosystems and the

[1] The work presented in this paper builds upon the Master's thesis of Daryna Pyshchyk. For a comprehensive discussion and further details, see `https://hdl.handle.net/10062/116946`

[2] A newer version of ISO 9241-210 (2019) is available; however, the 2010 version was used because it was the current standard at the time of the PHArA-ON project's design phase. The 2019 revision refines terminology but does not alter core HCD principles, and our approach remains fully compatible with the updated standard.

[3] `https://www.pharaon.eu/`

ISO 9241-210 standard. Section 3 details the Design Science Research methodology employed. The process of constructing the proposed approach is described in Section 4, while Section 5 presents the approach itself. The results of the expert evaluation are discussed in Section 6. This is followed by a discussion of the approach's implications and benefits in Section 7 and an examination of threats to validity in Section 8. Finally, Section 9 concludes the paper and suggests directions for future work.

2 Baseline

2.1 ISO 9241-210

ISO 9241-210:2010 is an international standard that provides guidance on human-centred design (HCD) for interactive systems ISO 9241-210 (2010). It defines principles and activities that help ensure that the systems are usable and meet the users' needs effectively. The standard promotes understanding users, including them throughout development, and evaluating the design with their feedback. ISO 9241-210:2010 offers a process (depicted in Figure 1) that consists of the following four interrelated activities:

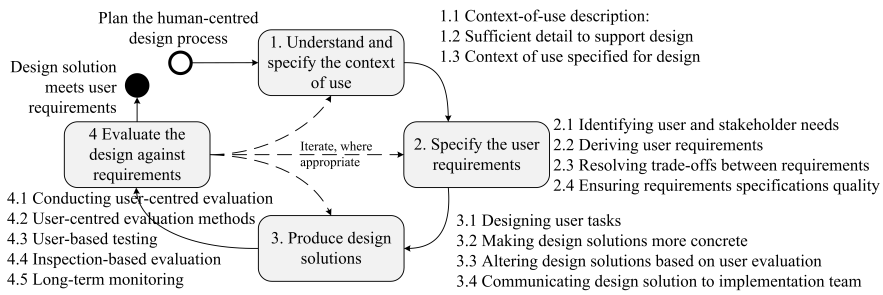

Fig. 1. Interdependence of human-centred design activities ISO 9241-210 (2010)

1. **Understand and specify the context of use:** Identify users, their tasks, environments, and any constraints to ensure that design decisions are based on real-world sittings.
2. **Specify the user requirements:** Translate the understanding of context into clear user needs and usability goals that the system must meet.
3. **Produce design solutions:** Develop and refine design alternatives that address the specified user requirements. These can range from sketches to interactive prototypes.
4. **Evaluate the design against the requirements:** Use user feedback and usability testing to assess how well the solutions meet user needs. This step helps identify improvements and supports iterative design.

2.2 The PHArA-ON project

PHArA-ON (Pilots for Healthy and Active Ageing in Europe) is a project developed under the EU Horizon 2020 programme to address the growing need for smart solutions to enhance the quality of life for older adults by addressing their health, well-being, and social engagement needs. PHArA-ON addresses several key challenges, including making technology more elderly-friendly, defining and monitoring health status, enabling non-intrusive monitoring, and fostering social cohesion. Additional challenges involve creating personalized care plans, reducing loneliness through digital tools, improving accessibility through IT platforms, and improving cultural awareness. To overcome these challenges, PHArA-ON leverages advanced services and tools to build an integrated, customizable, and interoperable ecosystem, namely, the PHArA-ON ecosystem.

The PHArA-ON ecosystem primarily serve older adults, healthcare professionals, and formal and informal caregivers. Fernandes et al. (2021). To meet the needs of these users, a user-centric approach has been adopted to maximize the usability and acceptance of the PHArA-ON ecosystem among all stakeholders. The PHArA-ON ecosystem undergoes two phases of validation: pre-validation and large-scale pilot (LSP) validation at its six different pilot sites (Spain (Murcia and Andalusia), Portugal, the Netherlands, Slovenia, and Italy), ensuring its effectiveness and readiness for broader implementation Gharib et al. (2024).

3 Methodology

Our human-centred approach for SECOs engineering has been developed following a Design Science Research (DSR) methodology Hevner et al. (2004). Applying this methodology ensures that the resulting SECO engineering approach is both rigorous and suitable for real-world application. Following Bell et al. Bell et al. (2007), our methodology (depicted in Figure 2) is composed of the following steps:

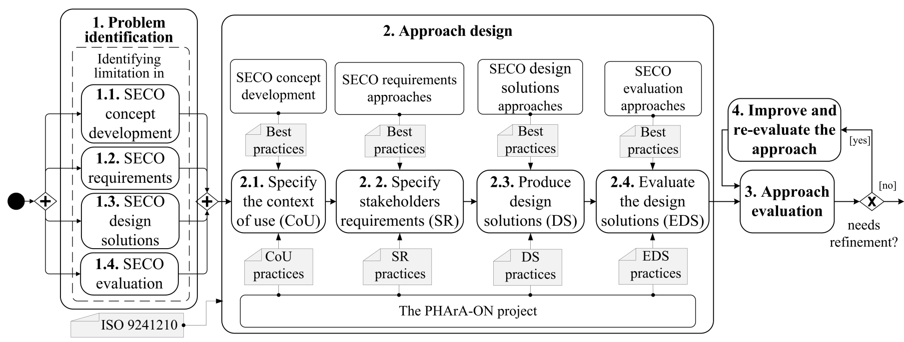

Fig. 2. DSR Method Application

1. **Problem identification**: As previously discussed, existing work lacks a domain-independent SECO engineering approach, which systematically integrates human-centred design principles that spans requirements, architecture, implementation, and evaluation. Consequently, there is a need for the development of such approach.
2. **Approach design**: Building on ISO 9241-210 principles ISO 9241-210 (2010), we constructed our five-stage human-centered approach by integrating established SECO best practices with practical insights from the PHArA-ON project. A detailed description of how the approach was constructed is provided in Section 4, and the approach is presented in Section 5.
3. **Approach evaluation**: Semi-structured interviews with SECO experts were conducted to assess the practicality, completeness, and applicability of the approach. A detailed description of this phase is provided in Section 6.
4. **Improvement and re-evaluation**: Insights from expert evaluation informed improvements to the approach, addressing identified limitations and enhancing its utility for SECO engineering in diverse domains.

4 The process for constructing the approach

This section discusses the process we followed for designing the SECO engineering approach (depicted in Step 2 in Figure 2). Considering the four key activities offered in ISO 9241-210 ISO 9241-210 (2010), the approach has been constructed by integrating SECO best practices with the experiences used for the PHArA-ON ecosystem. In PHArA-ON, human-centred design is evidenced through active user participation, co-design activities, and iterative evaluation across diverse pilot sites. Accordingly, PHArA-ON is a strong example of how the principles of ISO 9241-210 ISO 9241-210 (2010) can guide the development of SECO. In the rest of this section, we present and discuss how each of the key steps in the approach was developed, using the best practices of SECO engineering, followed by how it was followed and used in PHArA-ON the project.

1. SECO Context of Use Specification. SECOs are commonly developed to achieve specific goals or solve prominent problems, which shape their structure and stakeholder roles. These should be defined during the early phases of the SECO development, i.e., the context of use. According to Knauss et al. Knauss et al. (2014), following the five-themes approach that identifies key stakeholder roles helps in strategic planning and participation. Cioroaica et al. Cioroaica et al. (2021) suggested dividing SECO goals into strategic, tactical, and operational levels to improve management and adaptability. Christensen et al. Christensen et al. (2014) also noted that SECO goals often address integration challenges.

In **PHArA-ON,** ten top-level goals (to mitigate prominent challenges) were identified at this phase Fernandes et al. (2021), aiming to address aspects such as health monitoring, social cohesion, personalized care, and support for caregivers. Additionally, the consortium members were assigned tasks aligned with these goals, covering requirements, architecture, and evaluation. The stakeholder groups in PHArA-ON were organized hierarchically: keystone players (platform and solution providers), niche players (service providers that address specific needs), and users (older adults, caregivers, professionals) Fernandes et al. (2021). Additional roles, such as researchers and experts, supported compliance and scalability Fernandes et al. (2021).

2. SECO Requirements Specification. Requirements specification plays a crucial role in SECO engineering as it involves the elicitation, analysis, classification, prioritization, and validation of SECO requirements, ensuring their accuracy, completeness, and consistency as well as their alignment with the objectives of the system and the needs of stakeholders Hanssen and Dybå (2012); Damian et al. (2021). Various methods have been developed for each of these activities. Despite this, most of these methods seem to be limited to dealing with SECOs requirements without adjustments. A key aspect in SECO engineering involves *cross-cutting concerns*, encompassing requirements that affect multiple components of the system and stakeholders Knauss et al. (2014). Typical cross-cutting concerns include security, privacy, or interoperability, which often require coordination between the platform and its peripheral actors. When different stakeholders request features that overlap or conflict, experts must decide whether these concerns should be managed at a central or a more peripheral level in the SECO Knauss et al. (2014). Moreover, interdependencies among requirements Gharib, and Mirzazada (2025) could be another issue for such complex systems.

In **PHArA-ON,** requirements were elicited by a variety of methods Mooses et al. (2022) by systematic mapping of challenges to goals and solutions and representing goals as motivational goal models Sterling and Taveter (2009); Miller et al. (2014, 2015). The classification of goals included functional, quality, and emotional goals Gharib et al. (2024). Cross-cutting requirements were captured by considering robust security standards, a GDPR-compliant privacy framework Gharib et al. (2021), and flexible interoperability solutions supporting multiple platforms and open APIs. Requirements were prioritized using a streetlight approach and validated through an iterative review of stakeholders, ensuring practical alignment and system reliability Fernandes et al. (2021).

3. Producing SECO Design Solutions. SECO design solutions require modular, interoperable architectures with strong governance to ensure scalability and sustainability Manikas and Hansen (2013). Consequently, this phase takes the requirements as input and derives the reference architecture of the overall SECO in terms of its components Gharib et al. (2024). In SECO, a *reference architecture* acts as a key blueprint, outlining common structures, design principles, and best practices to tackle domain-specific challenges Knodel and Manikas (2016). In ISO 9241-210 ISO 9241-210 (2010), this phase not only involves deriving the design solutions, but also implementing them Zajdel et al. (2022). Implementation practices such as DevSecOps combine continuous delivery with automated security controls, ensuring that vulnerabilities are addressed early in the development pipeline Zajdel et al. (2022). Further practices include coding guidelines, central repositories, and sandbox environments, which strengthen SECO quality Wulfert et al. (2022).

In **PHArA-ON,** the modular reference architecture was selected since it balances interoperability and user needs Grguric et al. (2021). Its architecture incorporated layered models (devices, platforms, services, applications, and collaboration), with cross-cutting concerns addressed at every layer Di Girolamo et al. (2021). This modular design allowed pilots to include or omit certain layers based on existing technologies and requirements at each site. Agile and work package-based management, shared repositories, and local adaptations supported flexible implementation D'Agostini et al. (2022). A DevSecOps-based CI/CD process enabled integration of various technologies, maintaining quality and security standards across all partners D'Agostini et al. (2022).

4. SECO Evaluation. SECO evaluation systematically assesses the structure, performance and sustainability by analyzing architecture, governance, quality assurance, and interoperability Axelsson and Skoglund (2016). SECO health is measured by robustness, productivity and the ability to create niches, typically using metric-based assessments such as software contributions and developer retention Amorim et al. (2017). Key Performance Indicators (KPIs) are commonly used to benchmark the health and sustainability of SECO, covering size, finances, satisfaction, performance, risk mitigation, and compatibility Fotrousi et al. (2014).

In **PHArA-ON,** the evaluation was designed to generate evidence on the impact of created services, focusing on health maintenance, health sustainability Levy et al. (2023), user acceptance, and cost effectiveness. A harmonized methodology was implemented to ensure that each pilot site follows a shared procedure for data collection and impact assessment. This process is guided by an "Evidence Generation Protocol", which defines the indicators, targets, and tools to measure the results. The evaluation follows a co-design approach to identify a mandatory set of common KPIs applicable across all pilot sites, ensuring consistency and comparability. The KPIs were developed iteratively, considering local SECO needs and PHArA-ON's overall objectives. The evaluation included two cohorts: an experimental and a control groups, and compared the measures at different time points, using statistical significance tests to assess the impact of interventions.

5 An approach for SECO engineering

The SECO engineering approach comprises five stages, each corresponding to a human-centred engineering activity grounded in the principles of ISO 9241-210 ISO (2010). While ISO 9241-210 integrates the creation of design solutions and their implementation within a single phase—"Produce design solutions"—we have deliberately decoupled this into two distinct stages: SECO design solution (architectural and conceptual design) and SECO implementation (technical realization and deployment). This separation is motivated by the distributed, multi-stakeholder nature of SECOs, where clarity of responsibility, enhanced traceability between architecture and code, and the need to manage socio-technical complexity are critical. By distinguishing design from implementation, we reduce role ambiguity, support parallel development streams, and improve the overall quality and sustainability of the ecosystem. The approach is cyclical, with evaluation results refining earlier stages. In what follows, we describe each of its five stages:

1. **Context of Use Specification.** Is the first stage and focuses on the initial definition of the context within which the SECO will operate, outlining the steps required to thoroughly understand SECO goals, its stakeholders, their roles, tasks, and needs. The outcome of this stage is a consolidated *context of use specification* for the SECO, and it is composed of two activities:

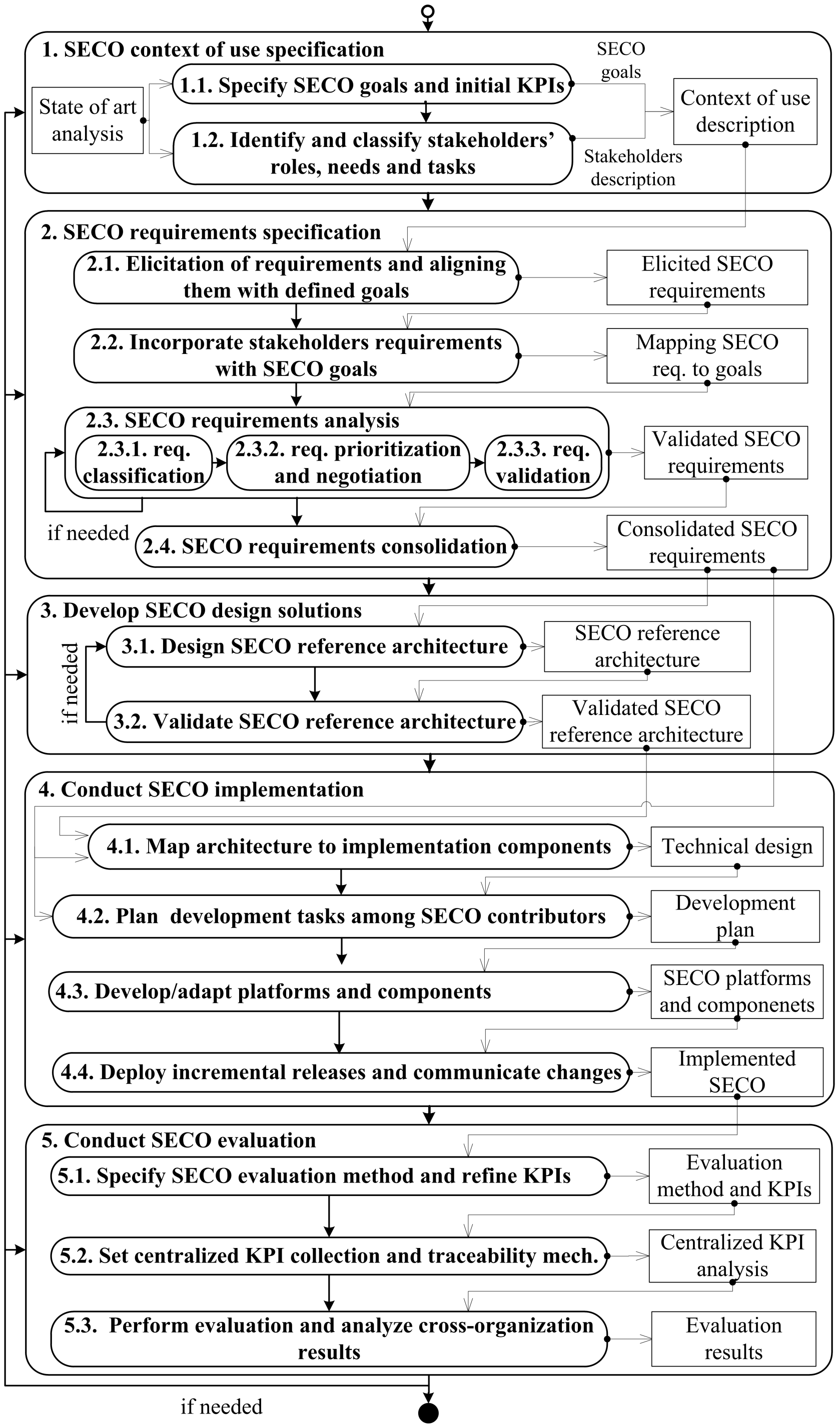

Fig. 3. SECO engineering approach

- ***1.1 Specify SECO goals and initial KPIs***: This activity involves identifying the strategic SECO goals and refining them into tactical and operational sub-goals. To ensure the feasibility of these goals, it is required to analyze the current advances in SECO development, review existing or comparable SECOs, and perform a preliminary evaluation of the target system along with prospective user and stakeholder groups. Where applicable, domain-specific regulations and strategic documents should also be analyzed to ensure alignment within the domain. This establishes a deeper context and directs the main focus of the system. Refining goals into sub-goals also facilitates the distribution of the responsibilities for their achievement among involved stakeholders, allowing each party to work towards objectives that aligns with their expertise and interests Cioroaica et al. (2021). To enhance clarity in measurement and evaluation, each SECO sub-goal must incorporate specified initial KPIs. These KPIs will act as benchmarks to assess the progress and effectiveness of SECO initiatives. The outcome of this activity is SECO's goals.
- ***1.2 Identify and classify stakeholders' roles, needs, and tasks***: This activity identifies and classify all relevant stakeholder groups based on their *roles* and *responsibilities* to support systematic engagement, which ensures that all contributions and responsibilities are well understood as different stakeholders could follow different goals Hedges and Furda (2018); Gharib et al. (2016). This also allows for targeted and inclusive decision-making, and it is a crucial step as it influences how requirements will be generated Knauss et al. (2018). While keystone players, niche players, and users provide a starting point, further subcategories may align better with specific objectives, i.e., additional roles (standardization bodies, operators, components/platforms providers/developers, different types of end-users, etc.) can be considered. The outcome of this activity is the stakeholders' description.

2. Requirements Specification. This stage elicits, analyzes and validates the requirements for the SECO-to-be. It ensures that both technical and organizational aspects of the SECO are systematically captured and aligned with stakeholder expectations. This stage produces the *consolidated SECO requirements*, and includes the following steps:

- ***2.1 Systematic elicitation of SECO requirements aligned with defined goals***: This activity involves eliciting and refining the stakeholder needs, expectations, and constraints concerning the SECO-to-be. Given the evolving and complex nature of SECO, it is crucial that requirements align with SECO objectives. Consequently, each entity responsible for specific sub-goal(s) – that require functional component(s) to realize it - should choose suitable elicitation methods (e.g., interviews or workshops). Still, the resulting requirements should be presented in a consistent format (goal models, scenarios, user stories) to support traceability and promotes effective communication among stakeholders.
- ***2.2 Incorporate stakeholders' requirements with SECO goals:*** This activity ensures alignment of diverse perspectives with initial objectives. Since SECO requirements originate from multiple sources (platform providers, developers, users, etc.), systematic mapping to defined SECO goals helps identify gaps or

inconsistencies for correction. This ensures unified, balanced, and strategically consistent SECO requirements. We recommend performing this integration using a goal-oriented requirements engineering (GORE), which allows verifying how each stakeholder requirement supports or impacts strategic, tactical, or operational SECO goals. This step is also useful for communicating the rationale behind decisions to stakeholders. Regular review of this mapping is encouraged, especially in dynamic SECOs, where goals and stakeholder roles may evolve over time. Ultimately, this activity helps maintain internal coherence, supports transparency, and ensures that the SECO evolves in line with both stakeholder expectations and strategic direction.

2.3 SECO requirements analysis: This activity main objective is to categorize, prioritize, and validate SECO requirements.

- ***2.3.1 Requirements classification***: This sub-step classifies the requirements into four main categories. *1. Functional requirements (FRs)* define SECO services, APIs, and processes. *2. Non-functional requirements (NFRs)* specify quality attributes such as performance, scalability, usability, maintainability, and resilience. *3. Cross-cutting requirements,* include aspects that influence the architecture and coordination mechanisms through SECO components such as security, privacy, interoperability. In contrast, FRs and NFRs may only apply to specific components or services rather than the entire SECO. *4. Domain-specific requirements*, includes aspects specifically related to the SECO of concern or emerge due to its unique structure and stakeholder dynamics. These requirements include aspects such as emotional requirements that serve as a catalyst for building trust between organizational boundaries, cultivating a sense of community, and encouraging collaboration.
- ***2.3.2 Requirements prioritization and negotiation***: This sub-step involves the ranking of SECO requirements based on their significance, feasibility, and overall impact on SECO goals. Given the distributed nature of SECO, prioritization must consider factors beyond individual stakeholder influence, incorporating the impact of SECO, interdependency between components, and long-term sustainability. For example, keystone players might prioritize interoperability and security, while niche creators may focus on API stability and monetization opportunities, leading to potential conflicts. If disagreements persist, *requirements negotiation* should be used to resolve differences. Using structured frameworks (e.g., EasyWinWin and StakeSource2.0), along with iterative techniques, helps align stakeholder expectations with technical feasibility.
- ***2.3.3 Requirements validation***: This sub-step verify that the SECO requirements are valid, consistent, complete, and feasible while also aligning with stakeholder needs. Unlike conventional software development, SECO requirements validation must consider compatibility, external contributions, and evolving platform limitations. Validation should involve iterative methods, such as stakeholder walk-through, prototyping, and user acceptance testing to assess both functionality and usability in real-world scenar-

ios. This process ensures that any gaps or issues are quickly identified and addressed, allowing for refinement of requirements and supporting the sustainable development of the SECO. To improve the validation, it is vital to utilize automated tools when possible to identify inconsistencies, embrace common standards for seamless interaction.

2.4 SECO requirements consolidation: This activity refines and consolidates the requirements into a structured and validated list that will guide the design and implementation of SECO. Iterative feedback is used to ensure that the requirements are actionable, technically sound and fully aligned with the evolving goals of the SECO. The consolidated requirements are documented in a form suitable for direct use by architecture and development teams, i.e., it serves as the foundation for the development phase, ensuring precision, feasibility, and alignment with the general goals of the SECO.

3. **Develop SECO design solution:** Based on the validated requirements, this stage produces the *validated SECO reference architecture* that defines how the SECO will be organized and operated, and involves two activities:

   3.1 Design SECO reference architecture: This activity develops a reference architecture that provides a modular and layered structure for the SECO. The process begins with an analysis of existing reference architectures relevant to the target SECO domain, which can include models such as ISO/IEC 30141, RAMI 4.0, or domain-specific architectures. The architecture should support the required functionalities, addresses cross-cutting concerns, and fosters SECO sustainability, and be designed around the following key layers:

   - ***Collaboration layer***: This layer provides tools and resources for SECO participants, including a marketplace for extensions, documentation for niche creators, developer portals, and forums for knowledge sharing. It fosters engagement and supports the co-creation of new services within the SECO.
   - ***Cross-cutting concerns layer***: This layer addresses critical aspects such as *interoperability*, ensuring seamless data exchange between components, *security*, protecting data and system integrity, and *data privacy*, ensuring compliance with regulations and safeguarding user information. These concerns impact all parts of the SECO and ensure its long-term stability and trustworthiness.
   - ***Domain-specific layers***: These layers accommodate domain-specific needs, such as additional services, specialized devices, and custom functionalities. Depending on the SECO's focus, this could include IoT integrations, AI models, or real-time processing engines.

   3.2 Validate SECO reference architecture: This activity involves checking the reference architecture against the requirements. During this step, we recommend iterative validation of architecture design where technical requirements and architectural components are refined step by step and validated in realistic scenarios. Throughout, focus on interoperable standards and modularity to accommodate the dynamic nature of SECOs, and utilize evaluation metrics to guide architecture updates.

4. **SECO Implementation.** This stage translates the reference architecture into a working SECO by developing, integrating, and deploying its components and services. It ensures that the SECO is implemented in an iterative and collaborative manner, while maintaining alignment with the SECO goals and requirements. This stage involves the following activities:

   4.1 Map reference architecture to implementation components and technical design: This activity focusses on realizing the SECO reference architecture into concrete implementation elements, ensuring that each architectural element is connected to an actual system component, interface, or integration pattern. A well-defined mapping from architecture to implementation is crucial for directing distributed development and minimizing team ambiguities. This requires a detailed description of the technologies, communication methods, and deployment settings for each component explicitly guided by the requirements obtained previously. To reduce the risk of under-specification, we recommend explicitly linking architecture views to API, module, and data format implementations, ensuring traceability. This improves team coordination, maintains architectural consistency, and ensures that the behavior of the SECO is in accordance with its goals.

   4.2 Plan and align development tasks among SECO contributors: This activity prepares the collaborative implementation process. Development tasks must be clearly scoped and assigned to SECO contributors in accordance with their roles and expertise. Iterative development cycles should be planned, supported by transparent coordination mechanisms such as shared repositories and regular integration reviews. It is essential to establish well-defined tasks for each group while simultaneously providing accessible channels for open collaboration. To preserve uniformity and build trust within SECO, it is equally important to articulate clear quality guidelines along with precise acceptance criteria for those who contribute code. Moreover, it is essential to implement robust review mechanisms specifically designed to assess adherence to security standards and conformity with interoperability guidelines. This ensures that high-quality solutions are delivered consistently.

   4.3 Develop/adapt core platform and components: This activity emphasizes the practical advancement of the SECO core platform along with its components. Development efforts should prioritize reuse of existing tools and frameworks, where possible, complemented by targeted development of SECO-specific components. This strategy helps reduce development overhead and accelerates implementation. APIs, data models, and integration mechanisms must be implemented in strict adherence to the interoperability standards defined in the architecture. Continuous technical documentation is critical to support collaboration and future evolution. All interfaces, services, and components should be documented with clear guidelines for third-party developers. Documentation of cross-cutting mechanisms (security, privacy, interoperability) must be prioritized to foster trust and transparency across SECO participants.

   4.4 Deploy incremental releases and communicate changes: This activity focuses on the methodical deployment of incremental updates to the SECO platform and its individual components. A continuous integration and delivery (CI/CD) process aligned with DevSecOps principles is recommended to ensure consistent quality and security across releases. The release process should accommodate both centralized and distributed pipelines to reflect the diversity of SECO contributors. Each release must be accompanied by clear and accessible release notes, changelogs, and technical documentation. Transparent communication of changes is essential to maintain alignment between SECO actors. Together, these practices facilitate efficient partnership collaboration, reduce barriers to product release, and foster SECO development.

5. **SECO Evaluation.** The SECO evaluation stage evaluates the structure, performance, and impact of the SECO by systematically analyzing its alignment with the defined SECO goals and KPIs. It ensures that the evaluation is conducted in a structured and transparent manner, providing actionable insights for SECO improvement.
   5.1 Specify SECO evaluation method and refine KPIs: To effectively evaluate SECO, it is essential to implement a clear and systematic evaluation methodology. This approach should assess key attributes, including robustness, productivity, and niche creation, while accounting for domain-specific characteristics and quality attributes. Initially, KPIs were defined per each SECO goal, establishing a foundational baseline for evaluation. In this step, these KPIs must be revisited and refined based on the insights gained during the implementation phase. Furthermore, refined KPIs should remain closely aligned with the predefined common SECO goals. It is crucial to map each KPI explicitly to an appropriate data collection method and clearly defined evaluation periods, ensuring accurate and meaningful ongoing analysis.
   5.2 Set up centralized KPI collection and traceability mechanisms: This activity focuses on establishing a centralized infrastructure for collecting, aggregating, and analyzing KPIs between SECO participants, while ensuring traceability and comparability of evaluation data. A consistent and coordinated evaluation setup enables meaningful cross organizational insights and supports evidence-based improvement. A centralized dashboard is recommended to aggregate KPI data from across the SECO. To ensure data consistency, semantic data models should be defined and adopted in all SECO. These models help interpret the evaluation results uniformly, even when data is gathered from diverse sources. In addition, advanced analytics methods, such as data mining, can be used to identify influencing factors and forecast trends based on KPI outcomes. Particular attention should be paid to privacy and security and related mechanisms must be implemented in compliance with legal and ethical standards (e.g. GDPR). Evaluation data must be traceable to specific SECO goals and components to enable root cause analysis and targeted improvement.
   5.3 Perform evaluation and analyze cross-organisation results: For a thorough assessment and examination of the results involving several SECO participants, it is crucial to consistently track specific KPIs. Use multiple imputation methods to address missing data and maintain statistical soundness when comparing across organizations. Employ an iterative process where results are

reviewed both locally and globally, allowing for the gradual improvement of SECO strategies. Regular evaluations are necessary. It is important to refine the SECO's implementation, design, or requirements by incorporating updated policies, integration guidelines, developer incentives, and architectural constraints based on the results of these evaluations. Additionally, it is recommended to publicly share the evaluation results to foster trust and encourage greater partner participation.

To support practitioners in applying the five-stage approach, a detailed, actionable checklist is provided in Appendix A - (Table **??**). This checklist summarizes key activities, expected deliverables, and recommended tools or methods for each stage, offering a practical guide for implementing the approach in real-world SECO projects.

6 Approach evaluation

In DSR, an artifact is evaluated based on how effectively it addresses the problem of concern Hevner et al. (2004). This section presents our evaluation of the extent to which our approach is both usable—meaning comprehensible and easy to follow—and useful, as perceived in terms of its value for SECO engineering.

Methodology. Experts in SECO requirements engineering, architecture, implementation, and evaluation were invited to an interview as practical input from professionals is prioritized over laboratory results in the context of DSR, aligning with human-centred design principles Venable et al. (2012). Specifically, we selected semi-structured interviews to facilitate a deeper, more conversational examination of expert experiences and to elicit detailed, qualitative input that reflects real-world practice. This choice is consistent with qualitative evaluation norms in DSR, where interactive discussion is essential for uncovering both the strengths and limitations of an artifact. The flexible format allowed us to probe unexpected points, seek clarification, and gather rich, practitioner-informed perspectives that a predetermined questionnaire could not easily capture.

Four experts[4] accepted to participate. The four experts were affiliated with different organizations (two from industry, one from a research organization, and one from academia), ensuring a balance of research and practical perspectives. This diversity was intentionally sought to mitigate organizational bias. Three of these experts also held key roles—two as work package leaders—in the PHArA ON project, providing them with applied, end-to-end experience across the SECO lifecycle. Each interview[5] lasted around one hour and followed a semi-structured format. Participants were first introduced to the approach, including its five stages and supporting diagrams, then, were asked the interview questions concerning the following criteria: **1. Clarity and completeness**: Are the approach steps understandable and sufficient? **2. Alignment with SECO practice**: Does the approach reflect real-world engineering experience? **3. Perceived usefulness**: Would the approach help in practical SECO projects? **4. Adoption Barriers**: What challenges may arise during implementation?

[4] While inviting potential participants, it became evident that professionals with direct SECO expertise are scarcely available.

[5] A detailed description of the interview procedure can be found in the Appendix in `https://thesis.cs.ut.ee/9f31a147-4e39-409a-8ec5-389a7cc39cf3`

Results and Discussion. The four experts (P1 - P4) have more than five years of experience with SECO in various areas (see Table 1[6]). Despite the small sample, the experts provided valuable and in-depth feedback. Key insights are summarized below.

Table 1. SECO experts: demographics and primary domains

ID	Exp. (yrs)	Affiliation	SECO Engineering Stage				
			1. Context	2. Req.	3. Design	4. Implementation	5. Evaluation
P1	10–15	Industry	○	○	●	●	⊘
P2	6	research org.	●	●	●	⊘	○
P3	5–6	Industry	○	●	○	○	⊘
P4	5–6	Academia	●	●	⊘	⊘	⊘

● = High expertise, ⊘ = medium expertise, ○ = limited expertise.

1. **Clarity and completeness**: The approach was generally considered clear and thorough by all participants. Each of them offered minor suggestions to improve clarity. P1 emphasized the need to refine stakeholder classification and requirements prioritization, proposing additional tools, such as stakeholder maps and MoSCoW matrices, to support transparent negotiation and prioritization among partners. P2 appreciated the decomposition from strategy to operational goals and recommended formalizing goals and requirements to allow traceability and validation between levels. P3 welcomed the dedicated requirements negotiation step. P4 advised viewing the reference architecture from three complementary perspectives—information, interaction, and behavior—to cope with SECO complexity.
2. **Alignment with practice**: All participants recognized the structure of the approach in their previous projects. P2 related it to orchestrated SECOs and noted that the flow fits projects where top-down goal decomposition is required. P4 confirmed that the flow mirrors common practice, but stressed that GDPR and cybersecurity requirements must remain visible throughout design and validation. P4 also underlined that the initial KPIs should be set during the context-of-use stage and refined during evaluation.
3. **Perceived usefulness**: P1 and P3 highlighted that the structured approach helps align requirements and communicate between teams. P2 observed that the approach enables alignment of the SECO development with its objectives. P4 saw the method as transferable to SECO projects in other domains and stressed that such a method is urgently needed because demand for systematic SECO engineering is increasing.
4. **Barriers to adoption**: The experts noted that they did not observe significant barriers to adopting the approach. P1 emphasized the importance of supporting teams in distributed SECOs through better documentation practices. P2 raised the need to put more emphasis on the mapping step between high-level architecture and concrete implementation and recommended that future iterations of the approach con-

[6] The coverage pattern reflects the conceptual vs. implementation-oriented in SECO engineering, where experts more commonly contribute to ecosystem architecture than to component-level realization or validation.

sider how unmet requirements at the operational level could trigger a re-evaluation of tactical or strategic goals. P3 elaborated that the approach itself does not pose a major adoption barrier: its stages are clear and can be followed ‘without major difficulties’, but any serious obstacles are likely to appear later during the concrete implementation work. P4 cautioned that effective adoption hinges on clearly defining what SECO is as misunderstandings may arise otherwise.

The evaluation confirmed the approach practicality and highlighted specific areas where additional support could improve its adoption in real-life projects. The experts viewed the approach as a solid starting point for structuring the SECO engineering tasks and accurately reflecting the activities carried out in practice. All feedback from P1–P4 was analyzed and incorporated into the approach final version to facilitate smoother adoption across diverse SECOs.

7 Discussion

The human-centred SECO engineering approach presented in this paper offers a structured methodology to address the significant challenges inherent in developing complex, multi-stakeholder software ecosystems. By integrating the established principles of ISO 9241-210 with empirically-grounded best practices from SECO research and the large-scale PHArA-ON project, the approach provides a comprehensive, domain-independent framework that advances beyond prior, often fragmented, efforts.

The primary benefit of this approach is its holistic coverage of the entire SECO lifecycle. Unlike methods that focus on isolated phases like requirements engineering or architectural design, our five-stage approach ensures continuity and traceability from the initial context-of-use analysis through to implementation and continuous evaluation. This end-to-end guidance is crucial for managing the socio-technical complexity of SECOs, where decisions made in early stages have profound implications for long-term sustainability and stakeholder alignment. The explicit separation of the ISO 9241-210 “Produce design solutions” phase into two distinct stages—“SECO design solution” and “SECO implementation”—is a deliberate adaptation informed by the distributed and multi-actor nature of software ecosystems. While the standard integrates design and implementation within a single iterative phase, SECOs involve contributors who are often geographically and organizationally dispersed, with varied roles ranging from platform architects to external developers. Merging design and implementation in such contexts can obscure accountability, hinder modular evolution, and complicate governance. Our two-stage separation ensures that architectural decisions are explicitly documented, validated, and aligned with cross-cutting concerns (e.g., security, interoperability) before being translated into deployable components. This enhances transparency, facilitates parallel development streams, and improves the overall quality and sustainability of both architectural and implementation artifacts. This structural adjustment emerged from practical challenges observed in the PHArA-ON project, where modular reference architectures required clear handovers between design and development teams across European pilot sites.

Furthermore, the approach provides practical, actionable guidance for key SECO engineering activities. For practitioners, the framework acts as a systematic checklist.

During the “Context of Use Specification”, it guides teams in defining strategic goals and classifying diverse stakeholder roles, which is foundational for inclusive design. In the “Requirements Specification” stage, the method for incorporating cross-cutting concerns like security and interoperability, coupled with structured prioritization and negotiation techniques, offers a concrete mechanism to reconcile the often-conflicting needs of keystone players, niche creators, and end-users. The emphasis on developing a modular reference architecture and mapping it clearly to implementation components directly addresses the need for interoperability and scalable governance. Finally, the “SECO Evaluation” stage, with its focus on refining KPIs and centralized data collection, moves beyond artificial assessment to enable evidence-based, continuous improvement of the ecosystem’s health.

The expert evaluation confirmed that the approach is not merely theoretical but is perceived as highly usable and aligned with real-world practice. Experts recognized its value in facilitating communication between distributed teams, aligning development activities with strategic objectives, and providing a much-needed common structure for SECO projects. The ability to tailor the approach to different domains, as evidenced by its genesis in the health-tech sector with PHArA-ON and its perceived transferability by experts, underscores its generalizability.

In practical terms, this approach can be used as a guiding blueprint for organizations embarking on SECO development. It empowers engineering teams to systematically address human factors alongside technical ones, ensuring that the resulting ecosystem is not only robust and scalable but also usable and valuable for all participants. By making the design process transparent and iterative, the approach helps build trust among contributors and increases the likelihood of creating a sustainable ecosystem. While the initial investment in following the structured process may be higher than ad-hoc methods, the long-term benefits—including reduced rework, improved stakeholder satisfaction, and greater ecosystem resilience—are substantial.

While the framework is comprehensive by design—reflecting the multifaceted nature of SECO engineering—we recognize that its adoption may require initial investment in training and organizational alignment. To lower the entry barrier, future iterations will include lightweight variants, tool-supported automation of routine tasks, and context-specific templates that allow teams to adopt the approach incrementally. These supporting materials will help balance theoretical completeness with practical usability, ensuring the framework can be effectively deployed even in resource-constrained environments.

8 Threats to Validity

While the evaluation of the proposed approach provides positive initial evidence of its utility, several threats to the validity of the findings must be acknowledged.

Construct Validity concerns the relationship between theory and observation. The evaluation relied heavily on expert perceptions of the approach’s clarity, alignment with practice, and usefulness. While these are appropriate metrics for an initial evaluation in a Design Science Research context, they are subjective measures. The positive feedback may be influenced by the participants’ general agreement with the underlying principles of human-centred design rather than a rigorous assessment of the approach's causal impact on SECO success. Future empirical studies applying the approach in live projects are needed to gather objective data on its effectiveness.

Internal Validity relates to the causal relationships within the study. A primary threat is the limited number of expert participants (n=4). Although the experts possessed relevant and extensive experience, the small sample size limits the generalizability of the findings and increases the risk that the results reflect the specific opinions of a small group. Furthermore, the selection of experts, while necessary given the niche expertise required, may introduce selection bias. The experts who agreed to participate might have a pre-existing interest in structured methodologies, potentially leading to a more favorable assessment than a randomly selected group of practitioners might provide.

External Validity concerns the generalizability of the results. The approach was constructed and evaluated primarily based on insights from a single, large-scale project in the health-tech domain (PHArA-ON). Although this project involved diverse technologies and stakeholders, the specific constraints, regulations, and stakeholder dynamics of the active ageing domain may not be fully representative of SECOs in other areas, such as e-commerce or open-source platforms. Consequently, the approach's applicability and effectiveness in different domains remain to be further tested. The expert interviews, while valuable, serve as a proof-of-concept rather than a broad validation across multiple industrial contexts.

Reliability refers to the extent to which the data collection and analysis are consistent and repeatable. The semi-structured interview format, while flexible, introduces a potential threat as the line of questioning might have varied slightly between participants. To mitigate this, an interview protocol was followed. However, the qualitative analysis of the feedback involves interpretation by the researchers, which could introduce subjectivity. The provision of the detailed interview procedure[7] helps to address this by improving the transparency and repeatability of the evaluation process.

In summary, while the initial evaluation is promising, the findings should be interpreted with these limitations in mind. The threats identified highlight the need for further validation through longitudinal case studies and broader application of the approach in diverse SECO projects.

9 Conclusions and future work

This paper has presented a comprehensive, human-centred approach for engineering Software Ecosystems (SECOs). The proposed five-stage framework addresses a significant research gap by integrating the principles of ISO 9241-210 with established SECO best practices and practical insights from the industrial-scale PHArA-ON project. By providing end-to-end guidance, the approach offers a structured methodology to manage the socio-technical complexity inherent in SECO development. Its key contribution lies in systematically embedding human-centred design throughout the entire SECO lifecycle, ensuring that ecosystems are not only technically robust but also usable and valuable for all stakeholders.

[7] A detailed description of the interview procedure can be found in the Appendix in `https://thesis.cs.ut.ee/9f31a147-4e39-409a-8ec5-389a7cc39cf3`

The expert evaluation confirmed the approach's practicality, highlighting its clarity, alignment with real-world engineering challenges, and perceived usefulness in facilitating communication and strategic alignment within SECO projects. The framework provides an actionable blueprint for practitioners, enabling them to navigate complex stakeholder relationships, reconcile conflicting requirements, and foster sustainable ecosystem growth.

Despite these promising results, several limitations must be acknowledged. The primary limitation is the scale of the evaluation, which involved a small number of experts, potentially limiting the generalizability of the findings. Furthermore, while the approach was constructed and initially validated using a large, real-world project, its grounding in the specific context of the PHArA-ON health-tech ecosystem may not fully capture the challenges present in other domains, such as e-commerce or open-source platforms. The inherent complexity of the approach may also present an adoption barrier, requiring significant initial effort and experience from engineering teams.

Future work will focus on addressing these limitations and enhancing the framework's utility. First, we plan to conduct more extensive empirical validations, including longitudinal case studies applying the approach in diverse SECO projects across different domains. This will provide robust, objective data on its effectiveness and adaptability. Second, to enhance practical adoption, we will develop tool-assisted implementations (e.g., for stakeholder mapping, requirement traceability, and KPI monitoring) alongside simplified, scenario-specific guides that enable teams to tailor the approach to their project's scale and domain. Finally, we aim to create a richer set of supporting materials, including detailed guidelines and pattern libraries for common SECO engineering scenarios, to further assist practitioners in implementation. Through these efforts, we seek to refine the approach and solidify its role as a foundational method for human-centred SECO engineering.

Acknowledgments

This work was supported by the Estonian Research Council grant "Developing human-centric digital solutions" (TEM-TA120).

References

Amorim, S.S., Neto, F.S.S., McGregor, J.D., de Almeida, E.S., Chavez, C.v.F.G. (2017). How Has the Health of Software Ecosystems Been Evaluated? A Systematic Review, *Proceedings of the XXXI Brazilian Symposium on Software Engineering*, pp. 14–23. ACM. `https://doi.org/10.1145/3131151.3131174`

Axelsson, J., Skoglund, M. (2016). Quality assurance in software ecosystems: A systematic literature mapping and research agenda, *Journal of Systems and Software*, 114, pp. 69–81. `https://doi.org/10.1016/j.jss.2015.12.020`

Bell, D., de Cesare, S., Iacovelli, N., et al. (2007). A framework for deriving semantic web services, *Information Systems Frontiers*, 9(1), pp. 69–84. `https://doi.org/10.1007/s10796-006-9018-z`

Bosch, J. (2009). From software product lines to software ecosystem, *Proceedings of the 13th International Software Product Line Conference*, ACM, pp. 111–119.
https://doi.org/10.1145/1753235.1753251

Chirumamilla, A., Nguyen Duc, A., Gao, S., Sindre, G. (2018). A Systematic Mapping Study on Requirements Engineering in Software Ecosystems, *Journal of Information Technology Research*, 11, pp. 49–69.
https://doi.org/10.4018/JITR.2018010104

Christensen, H.B., Hansen, K.M., Kyng, M., Manikas, K. (2014). Analysis and design of software ecosystem architectures – Towards the 4S telemedicine ecosystem, *Information and Software Technology*, 56(11), pp. 1476–1492.
https://doi.org/10.1016/j.infsof.2014.05.002

Cioroaica, E., Purohit, A., Buhnova, B., Schneider, D. (2021). Goals within Trust-based Digital Ecosystems, *2021 IEEE/ACM Joint 9th International Workshop on Software Engineering for Systems-of-Systems and 15th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (SESoS/WDES)*, pp. 1–7. IEEE.
https://doi.org/10.1109/SESoS-WDES52566.2021.00006

D'Agostini, S., Giammatteo, G., Vitale, P., Fabriani, P., et al. (2022). D5.1. PHArA-ON ecosystem – Initial release, PHArA-ON Project, Horizon 2020 (Grant Agreement No. 857188), D5.1, v1.0.

Damian, D., Linåker, J., Johnson, D., Clear, T., Blincoe, K. (2021). Challenges and Strategies for Managing Requirements Selection in Software Ecosystems, *IEEE Software*, 38(6), pp. 76–87.
https://doi.org/10.1109/MS.2021.3105044

Di Girolamo, M., Coffano, G., Nava, C., Sestu, C., Bonferini, C., Diarena, A., Denissiouk, V., Temporale, C., Mošmondor, M., Grgurić, A., Barriga García, J.M., Melero, F.J., Maestre, R., Perandrés Gómez, A., Dantas, C., Ortet, S. (2021). D2.2. PHArA-ON initial ecosystem architecture, PHArA-ON Horizon 2020, EU Horizon 2020 Research Programme.

Fernandes, A., Silva, H., Dantas, C., Ortet, S., Rodrigues, F., Nery, M., Teixeira, S., Correia, P., Fiorini, L., D'Onofrio, G., Nijboer, F., Grünloh, C., Burnard, M., Perandres, A., Sánchez, M., Melero, F.J., Moreno, J., Mosmondor, M., Mooses, K., Iqbal, T., Taveter, K. (2021). D2.1. User and Pilot Requirements, PHArA-ON Consortium, Horizon 2020, Grant Agreement No 857188, Confidential Report.
https://pharaon.eu

Fotrousi, F., Fricker, S., Fiedler, M., Le Gall, F. (2014). KPIs for Software Ecosystems: A Systematic Mapping Study, *Software Business*, 182.
https://doi.org/10.1007/978-3-319-08738-2_14

Gharib, M., Falco, M., Nijboer, F., Tinga, A., D'Agostini, S., Rovini, E., Fiorini, L., Cavallo, F., Taveter, K. (2024). Dealing with Emotional Requirements for Software Ecosystems: Findings and Lessons Learned in the PHArA-ON Project, *Software Ecosystems*, pp. 99–114.
https://doi.org/10.1007/978-3-031-59465-6_7

Gharib, M., Giorgini, P., Mylopoulos, J. (2021). COPri v.2 — A core ontology for privacy requirements, *Data and Knowledge Engineering* **133**, 101888.
https://linkinghub.elsevier.com/retrieve/pii/S0169023X2100015X

Gharib, M. and Mirzazada, E. (2025). ReInTa: A Novel Requirements Interdependencies Taxonomy, *Baltic Journal of Modern Computing*, 13(4), pp. 834–861
https://doi.org/10.22364/bjmc.2025.13.4.05

Gharib, M., Salnitri, M., Paja, E., Giorgini, P., Mouratidis, H., Pavlidis, M., Ruiz, J. F., Fernandez, S., Siria, A. D. (2016). Privacy Requirements: Findings and Lessons Learned in Developing a Privacy Platform, *the 24th International Requirements Engineering Conference, RE 2016*, IEEE, pp. 256–265.
https://doi.org/10.1109/RE.2016.13

Grguric, A., Khan, O., Ortega-Gil, A., Markakis, E. K., Pozdniakov, K., Kloukinas, C., Medrano-Gil, A. M., Gaeta, E., Fico, G., Koloutsou, K. (2021). Reference architectures, platforms, and pilots for european smart and healthy living—analysis and comparison, *Electronics (Switzerland)* **10**(14).
https://www.mdpi.com/2079-9292/10/14/1616

Hanssen, G., Dybå, T. (2012). Theoretical foundations of software ecosystems, *Software Business*, 879, pp. 6–17.

Hedges, J., Furda, A. (2018). The Emerging Role of the Ecosystems Architect, *Digital Ecosystems: Principles and Semantics*, ch. 6, Taylor & Francis Group, LLC/CRC Press.
https://doi.org/10.1201/9780429022067-6

Hevner, A.R., March, S.T., Park, J., Ram, S. (2004). Design Science in Information Systems Research, *MIS Quarterly*, 28, pp. 75–105.

ISO/IEC/IEEE (2017). ISO/IEC/IEEE 12207:2017 Systems and software engineering - Software life cycle processes *I Iso/Iec/Ieee 12207:2017*.
https://www.iso.org/standard/63712.html

ISO/IEC/IEEE (2023). ISO/IEC/IEEE 15288:2023 Systems and software engineering — System life cycle processes *ISO/IEC 15288:2008(E), Std 15288-2008 (Revision of IEEE Std 15288-2004*.
https://www.iso.org/standard/81702.html

International Organization for Standardization (2010). ISO 9241-210: Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.
https://www.iso.org/standard/52075.html

Knauss, E., Damian, D., Knauss, A., Boriçi, W.A. (2014). Openness and Requirements: Opportunities and Tradeoffs in Software Ecosystems, *22nd International Requirements Engineering Conference (RE)*, IEEE.
https://doi.org/10.1109/RE.2014.6912263

Knauss, E., Yussuf, A., Blincoe, K., Damian, D., Knauss, A. (2018). Continuous clarification and emergent requirements flows in open-commercial software ecosystems, *Requirements Engineering*, 23.
https://doi.org/10.1007/s00766-016-0259-1

Knodel, J., Manikas, K. (2016). Towards reference architectures as an enabler for software ecosystems, *Proceedings*, pp. 1–4.
https://doi.org/10.1145/2993412.3003387

Levy, M., Groen, E. C., Taveter, K., Amyot, D., Yu, E., Liu, L., Richardson, I., Spichkova, M., Jussli, A., Mosser, S. (2023). Sustaining human health: A requirements engineering perspective, *Journal of Systems and Software* **204**.

Manikas, K., Hansen, K.M. (2013). Software ecosystems – A systematic literature review, *Journal of Systems and Software*, 86(5), pp. 1294–1306.
https://doi.org/10.1016/j.jss.2012.12.026

Miller, T., Lu, B., Sterling, L., Beydoun, G., Taveter, K. (2014). Requirements elicitation and specification using the agent paradigm: The case study of an aircraft turnaround simulator, *IEEE Transactions on Software Engineering* **40**(10), 1007–1024.

Miller, T., Pedell, S., Lopez-Lorca, A. A., Mendoza, A., Sterling, L., Keirnan, A. (2015). Emotion-led modelling for people-oriented requirements engineering: The case study of emergency systems, *Journal of Systems and Software* **105**, 54–71.
https://www.sciencedirect.com/science/article/pii/S0164121215000667

Mooses, K., Camacho, M., Cavallo, F., Burnard, M. D., Dantas, C., D'Onofrio, G., Fernandes, A., Fiorini, L., Gama, A., Perandrés Gómez, A., Gonzalez, L., Guardado, D., Iqbal, T., Sanchez Melero, M., Melero Muñoz, F. J., Moreno Muro, F. J., Nijboer, F., Ortet, S., Rovini, E., Toccafondi, L., Tunc, S., Taveter, K. (2022). Involving Older Adults During COVID-19

Restrictions in Developing an Ecosystem Supporting Active Aging: Overview of Alternative Elicitation Methods and Common Requirements From Five European Countries, *Frontiers in Psychology* **13**, 676.

Serebrenik, A., Mens, T. (2015). Challenges in Software Ecosystems Research, *Proceedings of the 2015 European Conference on Software Architecture Workshops*, art. 40, 6 pp. ACM. `https://doi.org/10.1145/2797433.2797475`

Sterling, L., Taveter, K. (2009). *The art of agent-oriented modeling*, MIT Press.

Venable, J., Pries-Heje, J., Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research, *Design Science Research in Information Systems. Advances in Theory and Practice*, Springer, pp. 423–438.

Wulfert, T., Woroch, R., Strobel, G., Seufert, S., Möller, F. (2022). Developing design principles to standardize e-commerce ecosystems, *Electronic Markets*, 32, pp. 1–30. `https://doi.org/10.1007/s12525-022-00558-8`

Zajdel, S., Costa, D.E., Mili, H. (2022). Open source software: an approach to controlling usage and risk in application ecosystems, *Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume A*, pp. 154–163. ACM. `https://doi.org/10.1145/3546932.3547000`

Appendix A: Practical Checklist for SECO Engineering

This appendix provides a structured, actionable checklist for implementing the five-stage human-centred SECO engineering approach presented in this paper. Each stage is detailed with key activities, expected deliverables, and recommended tools or methods to support practitioners in real-world SECO projects.

Table 2: Stage 1: SECO Context of Use Specification

<table>
<tr><td colspan="3">Stage 1: SECO Context of Use Specification
Define the ecosystem's purpose, goals, stakeholders, and operational context.</td></tr>
<tr><th>Key activities</th><th>Key deliverables</th><th>Suggested tools/methods</th></tr>
<tr><td rowspan="2">1.1. Specify SECO goals and initial KPIs</td><td>Documented strategic, tactical, and operational goals</td><td>Goal-Oriented Requirements Engineering (GORE) (e.g., i*, Knowledge Acquisition in autOmated Specification (KAOS))</td></tr>
<tr><td>Preliminary KPIs linked to each goal</td><td>Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis and Benchmarking existing SECOs</td></tr>
<tr><td rowspan="2">1.2. Identify and classify stakeholders' roles, needs and tasks</td><td>Stakeholder registry with roles & responsibilities</td><td>Stakeholders analysis, Stakeholder mapping</td></tr>
<tr><td>Influence/interest matrix categorizes and prioritizes stakeholders based on their level of influence/power, and their level of interest/-concern in its outcomes</td><td>Persona templates offer profiles of key users to embody their specific needs, behaviors, and goals</td></tr>
</table>

Stage 2: SECO Requirements Specification Elicit, analyze, prioritize, and validate SECO requirements.		
Key activities	**Key deliverables**	**Suggested tools/methods**
2.1. Elicit SECO requirements and align with defined goals	Requirements inventory or User stories/scenarios	Interviews, workshops, questionnaire, Goal-oriented RE (GORE), or user stories
2.2. Incorporate stakeholders' requirements with SECO goals	A mapping between stakeholders' requirements and SECO goals	Requirements Traceability Matrix (RTM)
2.3. SECO requirements analysis	Categorized requirements	A requirements taxonomy that systematically organizes and categorizes the requirements into consistent, predefined groups or types.
	Prioritized requirements	MoSCoW prioritization that categories requirements based on their importance and necessity for project success
	Negotiation minutes	EasyWinWin / StakeSource 2.0
	Validated requirements	Model-Based validation, User Acceptance Testing (UAT), scenario-based Walkthroughs, compliance checklists, expert judgment
2.4. SECO requirements consolidating	Consolidated requirements	Model-Based validation, UAT, scenario-based Walkthroughs, compliance checklists, expert judgment

Stage 3: Develop SECO Design Solution Develop and validate a modular, interoperable reference architecture.		
Key activities	**Key deliverables**	**Suggested tools/methods**
3.1. Design SECO reference architecture	SECO reference architecture diagram	Architecture model (e.g., ArchiMate, SysML (Systems Modeling Language), UML (Unified Modeling Language), C4 (Context, Container, Component, Code) model)
3.2. Validate SECO reference architecture	SECO architecture validation report	Architecture Trade-off Analysis Method (ATAM)
	Updated architecture based on feedback	Scenario-based walk-throughs

Stage 4: SECO Implementation Translate architecture into deployed components with clear governance.		
Key activities	**Key deliverables**	**Suggested tools/methods**
4.1. Map architecture to implementation components	Component mapping matrix	Interface-Driven Mapping using OpenAPI/Swagger for REST APIs or AsyncAPI for event-driven interfaces. Architecture-to-components traceability report
4.2. Plan & align development tasks among SECO contributors	Task assignment matrix	Gantt charts / roadmaps
	Integration roadmap	Shared repositories (GitHub/GitLab)
4.3. Develop/adapt core platform & components	Deployable components (independently releasable, executable units of software)	DevSecOps practices/ CI/CD pipelines
4.4. Deploy incremental releases and communicate changes	Release notes & changelogs	GitHub Releases for integration with repositories, and platforms like Backstage for centralized publishing
	Deployment reports	Incident management systems link deployments to operational events, while dashboard tools (e.g., Grafana) visualize deployment success and trends for cross-team transparency.

Stage 5: SECO Evaluation Continuously assess SECO health, performance, and impact.		
Key activities	**Key deliverables**	**Suggested tools/methods**
5.1. Specify SECO evaluation method and refine KPIs	Updated KPI set	KPI dashboard templates (Grafana)
	An evaluation plan that defines the what, how, when, and who of assessing a SECO's performance and impact	GQM (Goal-Question-Metric) can be used to translate high-level goals into measurable indicators
5.2. Centralized KPIs and traceability mechanisms	Centralized KPI dashboard	A mapping mechanism linking each KPI to SECO goals, data sources, collection methods, and responsible stakeholders
	Data collection procedures	Automated data collection, transformation, and loading from distributed sources.
5.3. Perform evaluation and analyze cross-organisation results	Evaluation report	Statistical analysis tools
	Improvement action plan	Cross-organizational review workshops

Received November 5, 2025 , revised January 19, 2026, accepted February 1, 2025