



# What do we know about requirements management in software ecosystems?

Paulo Malcher<sup>1,2</sup> · Eduardo Silva<sup>2</sup> · Davi Viana<sup>3</sup> · Rodrigo Santos<sup>1</sup>

Received: 16 November 2022 / Accepted: 12 September 2023 / Published online: 11 October 2023  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

Among the activities in requirements engineering (RE), requirements management ensures that requirements are tracked throughout their life cycle, changes are controlled, and inconsistencies are corrected. Requirements management has become increasingly critical in new ways of developing software and emerging contexts such as software ecosystems (SECO). The changing nature of the SECO introduces complexity in requirements management and results in varied flows of emergent requirements, making managing requirements in SECO challenging. Hence, understanding how requirements management is performed in SECO can help requirements managers improve their practices. This work aims to characterize requirements management in SECO. We have conducted a systematic mapping study (SMS) to achieve this goal. We selected 29 studies using a hybrid search strategy (database search and snowballing). We defined nine characteristics of requirements management in SECO that differentiate it from requirements management in traditional software development. We identified four types of approaches to support requirements management in SECO: tool, method, model, and practice. We found that only three selected studies present an assessment of their approaches. Finally, we characterize requirements management in SECO as an open, informal, collaborative, and decentralized process involving multi-party actors susceptible to power relations.

**Keywords** Requirements engineering · Requirements management · Software ecosystem · Systematic mapping study

## 1 Introduction

Requirements engineering (RE) is commonly accepted as the basis of high-quality software [69, 84]. RE is an iterative socio-technical process to elicit, document, and manage the requirements of a developing system [35]. Heistracher et al. [40] state that in-deep RE and continuous requirements

management are key prerequisites for the success of software engineering (SE) and systems engineering endeavors. Yaseen et al. [104] highlight the importance of requirements management, noting that the successful implementation of all other phases of RE is related to its effectiveness.

Fernández et al. [27] mention that requirements management still faces problems related to incomplete and/or hidden requirements, lack of traceability, requirements change control, and communication failures. For the authors, requirements management is challenging. However, performing requirements management in complex systems within open and dynamic environments that go beyond the limits of a single organization, such as software ecosystems (SECO), is even more challenging [24, 59]. Manikas [65] defines SECO as developing multiple products derived from a common technological platform based on a central architecture integrated with other systems through network actors and artifacts.

SECO are becoming a predominant mode of software development [24]. However, Knauss et al. [53] claim that SECO introduces complexity in requirements management. For the authors, the complexity and changing nature of

---

✉ Paulo Malcher  
malcher@edu.unirio.br

Eduardo Silva  
caedsi95@gmail.com

Davi Viana  
davi.viana@ufma.br

Rodrigo Santos  
rps@uniriotec.br

<sup>1</sup> Federal University of the State of Rio de Janeiro, Rio de Janeiro, RJ, Brazil

<sup>2</sup> Federal Rural University of Amazônia, Capitão Poço, PA, Brazil

<sup>3</sup> Federal University of Maranhão, São Luís, MA, Brazil

SECO result in several new requirements based on ecosystem trends that make requirements management difficult, called emergent requirements. Axelsson and Skoglund [12] state that the dynamic set of stakeholders in SECO introduces challenges to understanding, managing, and maintaining the balance between different stakeholder requirements. According to Vegendla et al. [96], more research should be performed to investigate the current state of requirements management in SECO. Hence, investigating how requirements management is carried out in SECO can help professionals in the improvement of their practices, as well as researchers in the realization of an overview of the field.

In this work, we aim to characterize requirements management in SECO. To achieve this goal, we conducted a systematic mapping study (SMS) [50] to identify and analyze the current state of the requirements management in SECO concerning its characteristics and existing approaches. We considered “characteristics” as process elements [16, 26] that differentiate requirements management in SECO from requirements management in traditional software development (i.e., concepts/definitions, activities, actors’ relationships, and artifacts). We consider “approaches” as types of contributions [25, 71, 86] cited/proposed in the selected studies (i.e., methods, techniques, tools, and practices). In addition, we aim to verify if (and how) the approaches identified are assessed. This work also aims to offer to researchers in this field an overview of what has already been done and where their contributions are still needed.

We retrieved 1028 studies from seven digital libraries in database search (DBS) and 1458 studies in backward snowballing (BS) and forward snowballing (FS). Hence, we selected 29 studies in our SMS. As the main contribution of our SMS, we defined nine characteristics of requirements management in SECO. These characteristics include opening organizational boundaries, power relations between geographically dispersed multi-party actors, open communication channels, and emergent requirements spread across multiple requirements artifacts and stored in different repositories. We also identified four types of approaches (tool, method, model, and practice) to support requirements management in SECO. We identified 11 selected studies that only cite approaches to support requirements management in SECO (e.g., a selected study only cited that issue tracking tools support requirements management activities in SECO) and ten that propose specific approaches (e.g., a selected study proposed a method to support some activity of requirements management in SECO). Only three of the ten selected studies that proposed approaches presented their assessment as well.

Our main results show that requirements management in SECO is an open, informal, collaborative, and decentralized process involving multi-party actors susceptible to power relations. These multi-party actors manage different types of

requirements in SECO (i.e., product, platform, and integration requirements) through strategic and emergent requirements flows. Our results also show that the characteristics and approaches for requirements management in SECO are mainly related to openness and information sharing among a “crowd.” Hence, we discuss the relationship between the characteristics of requirements management in SECO and emerging RE concepts such as Crowd-Based RE (CrowdRE) and Open RE. We also realized that other RE activities influence or are influenced by requirements management in SECO, and we address this relationship in our study. Finally, we discuss challenges regarding the characteristics of requirements management in SECO.

The remainder of this article is structured as follows. Section 2 presents the background and related work. Section 3 introduces the research questions and research method details. Section 4 presents the results, which we then discuss in Sect. 5. Section 6 points out limitations and threats to validity of the study. Section 7 presents a research agenda for requirements management in SECO. Finally, Sect. 8 outlines the conclusion and future work.

## 2 Background and related work

### 2.1 Requirements management

Requirements management is an organized process of documentation, analysis, traceability, prioritization, change control, and requirements communication [41]. García et al. [34] consider requirements management a key area within RE. According to Berenbach et al. [15], requirements management makes RE possible for large projects and helps reduce the chaos inherent in such projects because it includes version control of all artifacts that are the requirements sources of the RE process. Yaseen et al. [104] state that incorrect requirements management is linked to more than half of the reasons for project failures. According to Jayatilleke and Lai [46], failure to manage the requirements changes is the leading cause of the project’s failure. Moreover, requirements management ensures that changes in requirements are reflected in project plans, activities, and work products [21].

Song [88] defines requirements management as a process that accompanies the planning and development of a system, capturing and mapping the origin and context of changes. For Wiegers and Beatty [99], requirements management includes all activities that maintain the integrity, accuracy, and validity of requirements agreements throughout the project. For the authors, the main tasks of requirements management are four main activities: version control, change control, tracking of requirements status, and requirements traceability. Version control is an essential aspect of requirements management and applies at the

level of individual and set requirements. Change control concerns the procedures, processes, and standards used to manage changes to requirements. Tracking the status of each requirement during system development is an essential aspect of requirements management and provides a more accurate indicator of project progress. Requirements traceability aims to document the dependencies and logical links between individual requirements and other system elements [99].

## 2.2 Software ecosystem

Barbosa et al. [13] and Manikas [65] mention that research on SECO started in 2003 in the book by Messerschmitt and Szyperski [66]. According to Bosch [17], SECO have offered organizations advantages related to the acceleration of innovation, the dispersion of innovation costs, and the reduction of software maintenance costs when sharing activities with other members of the ecosystem. Axelsson and Skoglund [12] explain that SECO are based on an open environment and have a range of actors from different organizations, interacting around an open or semi-open platform, thriving among themselves cooperatively and competitively (“coope-tition”), producing several software solutions or services.

Santos and Werner [79] present three vital elements in an SECO: (i) software, which exists as a common technological platform, central software technology, software solutions, software platform, or product line; (ii) transactions, taken as the understanding that encompasses profit, reward or also benefits other than financial; and (iii) relationships, which are the links among the actors based on elements (i) and (ii). There are also two fundamental concepts in the SECO: The first is a common interest in central software technology, and the second is the concept of a network of organizations, which relate to symbiotic aspects of evolution, commercial or technical jointly, where each of the actors can benefit from participating in SECO.

In addition, Santos and Werner [78] address SECO from three dimensions: (i) technical: centered on the platform (technology, infrastructure, or organization). It is driven by three factors: development infrastructure, governance, and requirements management; (ii) business: centered on the knowledge flow (management model for artifacts, resources, and information). It is guided by three factors, i.e., environment vision, innovation, and strategic planning; and (iii) social: centered on stakeholders (influence and knowledge networks). It is conducted by three factors: usefulness, promotion, and knowledge gain. There is a need for special attention to stakeholders in these three dimensions. Defining their roles, putting the user at the center of the process, and performing configuration management may establish some challenges in the SECO context.

## 2.3 Related work

As related work, Vegendla et al. [96] provide an overview of the RE in the SECO context. The work focused on two main points: (i) RE activities in the SECO context and (ii) non-functional requirements in the SECO context. The study only identified topics related to ER activities that have already been studied in the context of SECO. Its results highlight that requirements management is a challenging activity and cites global RE and management practices are topics that have already been studied in the context of SECO. However, it does not describe or discuss how these topics were investigated. The study does not cite any approach or characteristics of requirements management in SECO. This work differs from the study by Vegendla et al. [96] in that it aims to define characteristics of requirements management in SECO and identify approaches to support requirements management in SECO.

More broadly, some SMS on SECO was conducted to provide an overview of the subject. Barbosa et al. [13] present the subject’s benefits, limitations, and challenges. Manikas [65] lead a longitudinal study on SECO to provide an up-to-date overview of the field and document its development. Barbosa et al. [13] mention the importance of communicating the requirements among stakeholders that may be geographically distant. Barbosa et al. [13] also presents aspects related to requirements negotiation as a characteristic of a given SECO. Manikas [65] point out that RE is represented in the SECO literature and cite a work focusing on privacy requirements. Despite this, it is possible to notice that none of the works present more excellent detailing on requirements management in SECO.

Other works sought to analyze SECO through specific topics. Fontão et al. [30] conduct an SMS on mobile software ecosystems (MSECO). The authors state an apparent demand for approaches, methodologies, processes, and tools to support this type of SECO. Besides, relations were identified among the MSECO quality and business requirements and objectives verification. However, as the work is focused on the MSECO domain, the mapped approaches are focused or adaptable to this context and are not directed to requirements management. Franco-Bedoya et al. [31] conduct an SMS to assess state of the art in open-source software ecosystems (OSSECO) research. The study presents the most relevant definitions related to OSSECO and its particularities. In this context, the authors define requirements as an essential activity in OSSECO. The authors give the need to monitor OSSECO and link the collected data to the actors’ needs and quality requirements as a research agenda.

Jayatileke and Lai [46] identify studies that deal with requirements management and are common to SECO, such as, for example, a method that supports the requirements change management (RCM) in the context of global software

development (GSD), a paradigm in which software development activities are carried out beyond the geographic, cultural and temporal boundaries [4]. Furthermore, the existence of geographically distributed actors was identified as one of the challenges in agile RCM. However, despite containing elements in the research that are inherent characteristics of SECO, the authors do not incorporate the SECO perception. Moreover, the work of Jayatilleke and Lai [46] can serve as insight for research regarding requirements management in SECO, as it addresses aspects that permeate the topic.

Alsanoosy et al. [9] conduct a systematic literature review (SLR) that investigated the influence of culture on RE activities. The authors noted that studies on the relations between culture and RE practices are still immature. The authors emphasize that understanding how each culture undertakes the RE process contributes to effectively implementing their practices in a GSD context. In addition, the study highlights that the incompatibility of practices can lead to several challenges, such as the difference in time for developing requirements in different locations, the common understanding among software teams on how the work should be done, and the monitoring of the current stage of each requirement. However, the cited practices are common in the requirements management process. Our work considers the discussion presented by Alsanoosy et al. [9] and aims to investigate specifically the requirements management in a broader context such as SECO.

In the present work, we have not established a specific type of SECO to be studied. Therefore, requirements management was characterized in studies related to any SECO, including those aimed at MSECO or OSSECO. We notice that several topics have been approached about RE in SECO, among them identifying stakeholders and the roles for the requirements elicitation and analysis. Furthermore, it was possible to make an exciting mapping between RE activities and research topics in the SECO context. For example, topics related to modeling, reference model, and non-functional requirements were mapped for requirements elicitation. For requirements management, only global management practices were mentioned. Thus, the secondary studies in the SECO literature do not focus on requirements management or the definition and evaluation of approaches used to accomplish it. Overall, they seek to understand the RE process and activities in SECO.

### 3 Research method

In this section, we described our research method. We conducted an SMS to achieve the objective we present in Sect. 1. For Kitchenham et al. [51], an SMS aims to (i) survey the available knowledge about a topic; (ii) synthesize

this by categorization; (iii) identify where there are “clusters” of studies that could form the basis of a fuller review; and (iv) identify where there are “gaps” indicating the need for more primary studies. In our SMS, (i) we search the available knowledge about requirements management in SECO; (ii) we synthesize data about characteristics and approaches of the requirements management in SECO; (iii) we identify ‘clusters’ of studies about other RE activities in SECO (Sect. 5); and (iv) we identify “gaps” (Sect. 5).

According to Kitchenham and Charters [50], an SMS or SLR must perform the following phases: planning, conducting, and reporting the review. The following subsections detail the planning phase of this SMS. Section 4 presents details on the conducting phase and results.

#### 3.1 Goals and research questions

This study aimed to **characterize requirements management in SECO**. The objective of this study was formalized based on the GQM (Goal-Question-Metric) method, proposed by Basili and Rombach [14] and defined as follows: **analyze** the requirements management **with the purpose of** characterizing **with respect to** elements (i.e., concepts/definitions, activities, actors’ relationships, and artifacts) and types of contributions (i.e., methods, techniques, tools, and practices) **from the point of view of** researchers **in the context of** SECO studies. We consider “characteristics” as process elements and “approaches” as types of contributions cited/proposed in the selected studies. Hence, we defined three research questions (Table 1).

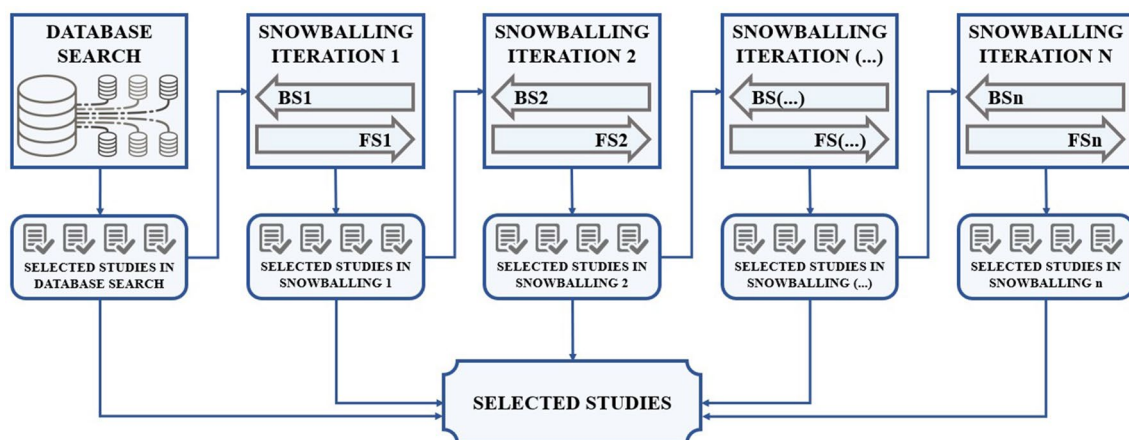
#### 3.2 Search strategy

We applied a hybrid search strategy [67, 101] in our SMS. Wohlin et al. [101] defined a hybrid search strategy as “the pre-planned integration of at least two systematic approaches to searching for articles for a systematic literature study.” Mourão et al. [67] presented four hybrid search strategies used in SE: (i) DBS in Scopus followed by full BS and FS; (ii) DBS in Scopus followed by BS and FS in parallel; (iii) DBS in Scopus followed by BS and then FS; and (iv) DBS in Scopus followed by FS and then BS. Wohlin et al. [101] compared and evaluated the four strategies and concluded that the full-fledged hybrid search strategy (i) is better than the other alternatives. For this reason, we applied strategy (i).

In the full-fledged hybrid search strategy (i), we must initially perform the DBS and then identify new studies by BS and FS based on the initial set of selected studies from the DBS. Our hybrid search strategy involved DBS

**Table 1** Research questions and their rationale

ID	Research question	Rationale
RQ1	What are the characteristics (process elements, i.e., concepts definitions, activities, actors relationships, and artifacts) of requirements management in software ecosystems context?	The answer to this RQ can reveal the main characteristics of requirements management in SECO that differentiate from requirements management in traditional software development. We consider the technical, human, and organizational factors used by Luz et al. [64] to characterize requirements management in SECO
RQ2	What are the approaches (types of contributions, i.e., methods, techniques, tools, and practices) used to support requirements management in software ecosystems context?	The answer to this RQ can reveal the approaches used in requirements management in SECO, as well as their status interest areas in the SECO research community, providing clues for summarizing future research directions. We define approaches (types of contributions of the studies) according to the software engineering research results described by Shaw [86]
RQ3	How have the existing approaches to requirements management in software ecosystems context been assessed?	The answer to this RQ can summarize the evaluation methods applied in the approaches identified in RQ2 and provide a basis for measuring their effectiveness. We use the empirical research methods in software ecosystems described by Abdullai et al. [3] and empirical research methods in software engineering described by Wohlin et al. [100] to categorize our results

**Fig. 1** Hybrid search strategy

in seven digital libraries: ACM Digital Library,<sup>1</sup> ScienceDirect,<sup>2</sup> Engineering Village,<sup>3</sup> IEEE Xplore Digital Library,<sup>4</sup> Scopus,<sup>5</sup> Web of Science,<sup>6</sup> and Wiley Online Library.<sup>7</sup> Wohlin et al. [101] highlight that BS and FS must be performed for all selected studies, i.e., they must

be performed in several iterations until no new studies are selected. Figure 1 presents our hybrid search strategy.

The guidelines provided by Kitchenham and Charters [50] recommend composing a string to find relevant studies by searching several digital libraries or index databases. To conduct the DBS, we initially identified keywords from terms present in the research questions. Afterward, we gathered the keywords into a search string (Table 2). Regarding the keywords, we restricted searches to find studies that investigated/cited requirements management in the specific context of SECO. We highlight that one of the first SMS on “software ecosystems” [13] used synonyms such as “software supply network,” “software vendor,” or “software supply industry.” However, more current studies [57, 65] use only “software ecosystems.” Additionally, the term

<sup>1</sup> <https://dl.acm.org/>.

<sup>2</sup> <https://www.sciencedirect.com/>.

<sup>3</sup> <https://www.engineeringvillage.com/>.

<sup>4</sup> <https://ieeexplore.ieee.org/Xplore/>.

<sup>5</sup> <https://www.scopus.com/>.

<sup>6</sup> <https://www.webofscience.com/>.

<sup>7</sup> <https://onlinelibrary.wiley.com/>.

**Table 2** Search string used in the study

Search string
(“software ecosystem” OR “software ecosystems”) AND (“requirements management” OR “requirements analysis” OR “requirements communication” OR “requirements traceability” OR “requirements documentation” OR “requirements prioritization” OR “requirements change” OR “requirements negotiation”)

**Table 3** Inclusion criteria

ID	Criteria
IC1	The study presents characteristics of requirements management in software ecosystems
IC2	The study proposes or cites requirements management approaches in software ecosystems

**Table 4** Exclusion criteria

ID	Criteria
EC1	The study does not meet at least one IC
EC2	The study is duplicated
EC3	The study is unavailable for free download or through institutional access
EC4	The study is not a peer-reviewed or is a Master’s or Doctoral thesis
EC5	The study is not written in English

“software ecosystems” has been a topic of interest at the International Conference on Software Engineering (ICSE)<sup>8</sup> in the past years and covered by other venues as well.

To increase the reliability of our search strategy, we use control studies in the DBS. Control studies support evaluating the quality and comprehensiveness of the search string [42]. Control studies must be retrieved when searching in a given digital library and are essential for calibrating a search string [68]. We identified the two control studies [53, 62] in exploratory studies we conducted previously. The two control studies investigated requirements management in SECO and were retrieved in different digital libraries as we calibrated our search string.

<sup>8</sup> <https://conf.researchr.org/track/icse-2024/icse-2024-research-track>.

### 3.3 Study selection

Kitchenham and Charters [50] divide the study selection into two parts: (a) definition of study selection criteria and (b) definition of the study selection process. We present below our definitions for each part.

#### 3.3.1 Study selection criteria

According to Kitchenham and Charters [50], for the selection of retrieved studies, inclusion criteria (IC) (Table 3) and exclusion criteria (EC) (Table 4) must be defined and applied. Kitchenham and Charters [50] mention that an SMS requires explicit IC and EC to evaluate each potential primary study and that these criteria are based on the set of research questions. We were inspired by the examples of IC and EC presented by Kitchenham et al. [51] in their recent reference book and the related work presented in Sect. 2.3 to define the set of IC and EC.

#### 3.3.2 Study selection process

In our SMS, we divided the selection process into seven stages and explained each stage and its relationship with the IC and EC, according to Kitchenham and Charters [50], in Table 5. We also considered the selection processes defined in the related work of Sect. 2.3. The selection process involved the four authors. The first two authors initially applied the IC and EC independently. Thereafter, the same two authors compared the results and sought a consensus in case of divergence. Finally, the last two authors, who have more than ten years of experience conducting and publishing secondary studies, assessed the results of each stage.

### 3.4 Data extraction and synthesis

According to Kitchenham and Charters [50], the data extraction process must collect the data items needed to answer the research questions. The data extraction process in a secondary study comprises elaborating forms to accurately record the information obtained in the primary studies [50]. Our data extraction form is openly available via ZENODO.<sup>9</sup>

During data extraction, we identified in each selected study: (a) specific characteristics of requirements management in SECO (RQ1); (b) approaches used for requirements management in SECO (RQ2); (c) whether or not there was an assessment of the approaches used for requirements management in SECO (RQ3); and (d) the method used for assessment (RQ3). We also extracted data that allowed us to do the demographic analysis of the studies selected in our

<sup>9</sup> <https://doi.org/10.5281/zenodo.8131745>.

**Table 5** Study selection process

Stage	Description
Stage 1	We apply the search string in digital libraries to obtain the set of studies retrieved
Stage 2	We apply the 1st filter (removal of duplicates) to the set of studies retrieved in stage 1. We frame all duplicate studies in EC2
Stage 3	We apply the 2nd filter (application of IC and EC after reading the title, abstract, and keywords) in the set of studies analyzed in stage 2
Stage 4	We apply the 3rd filter (application of IC and EC after reading the introduction and conclusion) in the set of studies analyzed in stage 3
Stage 5	We apply the 4th filter (application of IC and EC after the complete reading) in the set of studies analyzed in stage 4
Stage 6	We apply snowballing (BS and FS) in the set of studies analyzed in stage 5. Then, we perform stages 2, 3, 4, and 5 in the set of studies retrieved in snowballing. We iteratively repeat stage 6 until no new studies have been selected
Stage 7	We extract data from the studies analyzed in stages 5 and 6

SMS. We followed the example of some related work mentioned in Sect. 2.3 [9, 96]. These works present an overview of the demographic data of the select studies without referring to specific RQ. We identified in each selected study: (a) Title; (b) Year; (c) Authors; and (d) Country of 1st author.

We performed the data extraction process by following these steps: (a) The first two authors initially extracted data from the studies independently; (b) the first two authors jointly analyzed and compared the individual data extraction results and completed the data extraction form; and (c) the last two authors with more than fifteen years of experience in conducting secondary studies in SE and information systems evaluated the process and the results obtained at each stage. Ampatzoglou et al. [10] suggest the cross-check process of the extracted data by a more senior researcher. During this process, the role of the additional researchers is to cross-check results or resolve conflicts.

To answer the research questions, we used data synthesis. Kitchenham and Charters [50] state that there are two main data synthesis methods: (i) descriptive (qualitative) and (ii) quantitative. We used the qualitative method to analyze the extracted data and answer our research questions, which led to descriptive data synthesis [22]. We applying open and axial coding procedures [18] to support the qualitative synthesis method. Open coding involves applying codes that are derived from the text, and axial coding categorizes the initial codes into key codes [18]. Qualitative methods such as content analysis, thematic analysis, and grounded theory use coding procedures.

## 4 Results

Figure 2 presents an overview of the selection process. We performed a DBS in the seven digital libraries. We retrieved 1,028 studies published up to March 2023 in the DBS. Then, we applied the filters defined in the stages of the selection process (Sect. 3.3.2). We selected 25 studies in the DBS. Thus, we applied BS and FS to the 25 studies selected from the search database and retrieved 1300 studies. We selected

three studies in the first snowballing iteration. Then, we applied BS and FS for the second time in 120 retrieved studies in the three selected studies in the first snowballing iteration. Thus, we selected one study in the second snowballing iteration. Finally, we applied BS and FS for the third time in 38 retrieved studies in the selected study in the second snowballing iteration. However, when we applied the filters defined in the selection process, we did not select any new studies. Hence, we got a total of 29 selected studies in our SMS. Appendix A presents the selected studies in ascending order by year of publication and are referred to as S1–S29.

### 4.1 Demographic data and overview of results

Figure 3 shows the distribution of selected studies over the years. We noted that at least one study was published from 2012 to 2021. The selected studies had 87 different authors. However, for analyzing the countries related to each study, we considered only the countries of the first author. Figure 4 presents the distribution of publications by country. Sweden (9 studies) and Brazil (7 studies) are the countries that authored most of the selected studies.

Figure 5 presents this study's main findings and the summary of answers to research questions (Sects. 4.2, 4.3, and 4.4). In addition, Fig. 5 also shows other information discussed in Sect. 5, such as the main reasons requirements management in SECO is considered a challenge and what other activities of the RE are related to requirements management in SECO.

### 4.2 Characteristics of requirements management in SECO (RQ1)

In our work, characteristics are process elements [16, 26] related to requirements management in SECO that differentiate it from requirements management performed in traditional software development. To define them, we considered the technical, human, and organizational factors in SECO used by Luz et al. [64]. Luz et al. [64] state these factors can influence SECO technical and organizational elements.

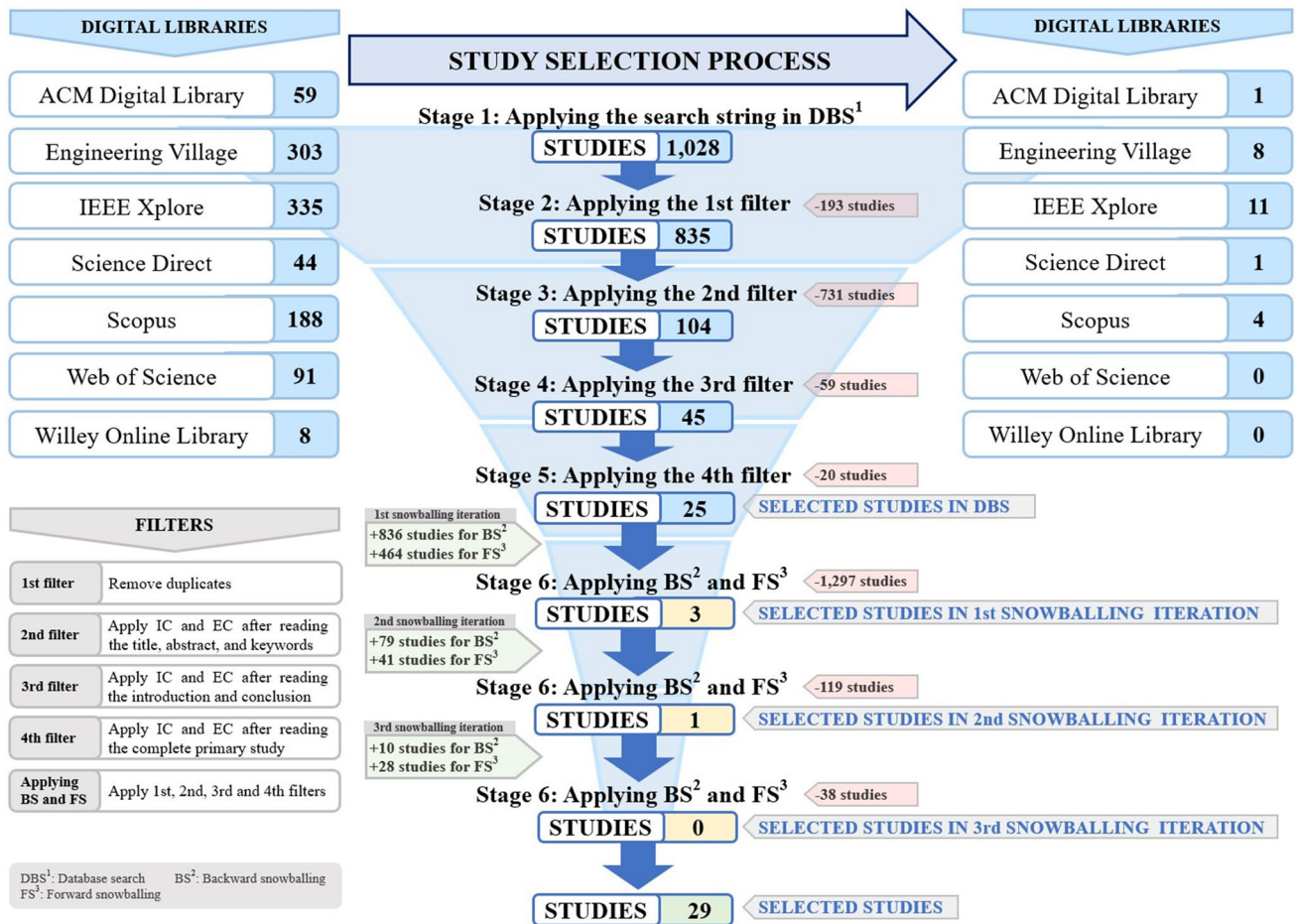
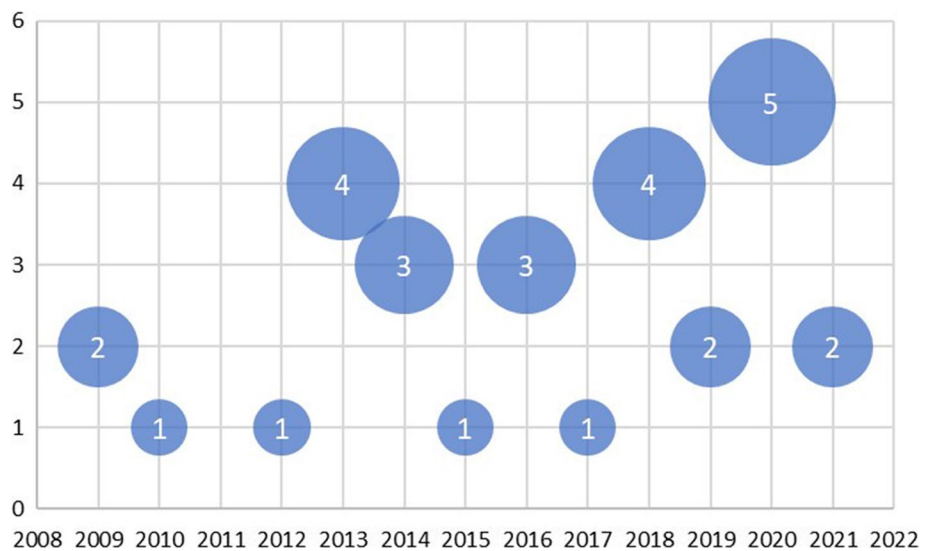


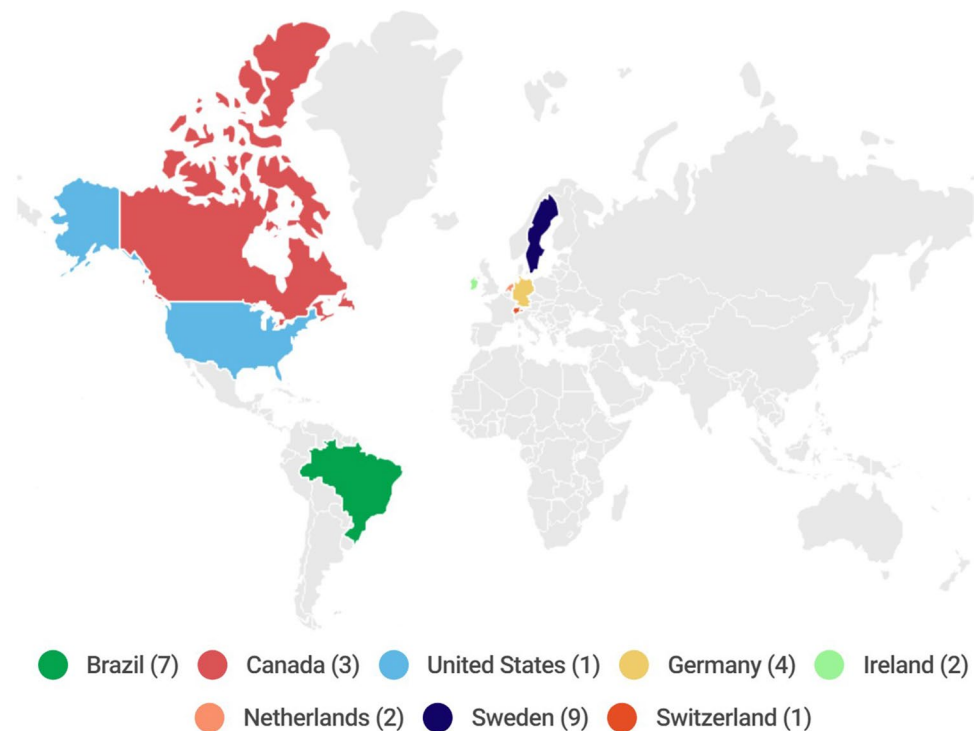
Fig. 2 Study selection process

Fig. 3 Distribution of select studies by year





**Fig. 4** Distribution of selected studies by country



For example, the factor *multiplicity of organizations and relationships of a SECO* influence the requirements management in SECO.

We defined the characteristics of requirements management in SECO through the data synthesis process detailed in Sect. 3.4. We used the synthesis qualitative method [22] to analyze the extracted data and answer our research questions. To do so, we initially used an open coding approach. We coded the extracted data of the selected studies in an inductive (bottom-up) way [18]. We performed a detailed reading of selected studies and divided parts of the text of these studies into coherent units (sentences or paragraphs). Subsequently, we agreed on a set of codes that captured the most frequent and relevant characteristics of requirements management in SECO cited in the selected studies. The coding was done by two authors and reviewed over several iterative cycles by the other two authors (with more than fifteen years of experience with SE research).

After executing open coding, we used axial coding to group the codes into categories, as described by Charmaz [18]. We categorized the characteristics into (i) concepts/definitions; (ii) activities; (iii) actors' relationships; and (iv) artifacts. Table 6 shows examples of the coding process of the selected studies (the resulting codes and their categories). The complete coding process is openly available via ZENODO.<sup>10</sup> Finally, Table 7 presents the nine characteristics of requirements management in SECO (C1–C9).

<sup>10</sup> <https://doi.org/10.5281/zenodo.8131745>.

#### 4.2.1 Concepts/definitions

C1 refers to the openness, informality, collaboration, and decentralization of requirements management in SECO. S13, S14, S17, S19, and S24 state that requirements management in SECO is informal, has overlapping practices and uses informalisms [83]. S3, S11, S13, S15, S17, and S24 mention that the requirements management process in SECO is decentralized and collaborative. S15 highlights that SECO require a collaborative and joint requirements engineering process. S24 defines requirements management in SECO as a lightweight and evolutionary process compared to more traditional processes characterized by heavy processes and supporting tools.

S11, S13, S14, S15, S16, S17, S20, S22, and S29 indicate that requirements management in SECO is open. S11, S16, S17, and S22 cite that the requirements management in SECO is cross-domain and uses the open innovation paradigm. S16 suggests that open innovation positively impacts requirements change management, quality requirements, and requirements communication. However, open innovation negatively impacts identifying stakeholders' needs, requirements prioritization, and requirements traceability. S20 mentions that SECO are, by nature, cross-domain and open. Then, requirements managers in SECO no longer deal with one domain at a time and must accept the openness inherent to SECO in requirements management activities. S11 points out that involving new business models to support open RE is necessary.

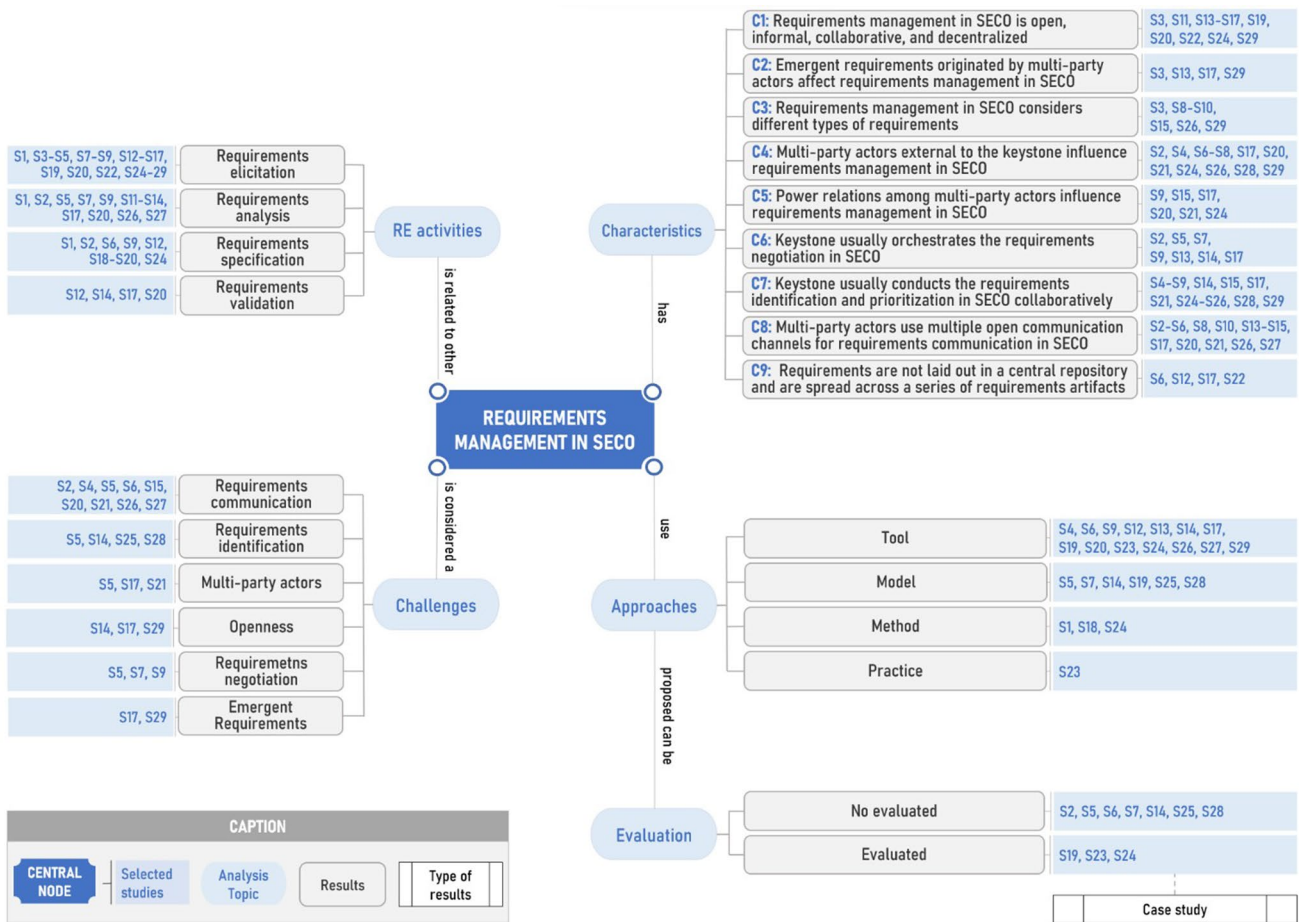


Fig. 5 Requirements management overview in SECO

C2 focuses explicitly on the concept of emergent requirements in requirements management in SECO. S17 mentions that the complexity and ever-changing nature of SECO results in several new requirements based on ecosystem trends and defines them as emergent requirements. S3, S13, and S17 cite that emergent requirements originate from end users of consumers in the ecosystem, who find new ways to use existing services. S29 mentions that new functionalities offered by external developers (complementors) can generate emergent requirements. S7 highlights that multi-party actors who are not responsible for the requirements but contribute to discussing them beyond organizational boundaries originate emergent requirements. For S29, emergent requirements allow keystone to innovate further to grow platform offerings, creating a circular flow of requirements knowledge. According to S17, it is difficult to map emergent requirements to specific actors, especially since they are generally transversal to several SECO actors.

C3 focuses on the types of requirements that affect requirements management in SECO. S3 bifurcates the requirements in SECO into kernel requirements and

periphery requirements. S8 ranks the requirements in SECO into partner requirements and customer requirements. S8 highlights that partner requirements in SECO may be a higher priority than customer requirements. For S9, there are product, integration, and platform requirements in SECO. Integration requirements must be “generic and reusable” among partners as they are part of a shared infrastructure. Platform requirements are the basis of the development environment actors use to develop complementary products in SECO. S15 highlights that most demands in a SECO are integration requirements involving data types exchanged and data dictionary, for instance. S29 mentions that evangelists are a key source of feedback and provide several integration requirements.

S10 identifies several requirements communicated by the keystone to other SECO members. S10 classifies them into three generic categories of requirements: strategic, brand, and platform requirements. Strategic requirements ensure a particular function that is essential to the systemic product the customer will acquire.

Branding requirements play a role in preserving SECO's brand reputation. Platform requirements facilitate product design and development in SECO. S26 carries out a case study in a specific SECO. It suggests that the RE process in SECO should have three main focuses: RE for the product, RE for the platform, and RE for the ecosystem (holistic view to identify integrations between products). For S26, the developers of partner applications are direct users of the platform, and their needs are essential for the ecosystem's health.

#### 4.2.2 Actors' relationships

C4 refers to groups of multi-party actors that influence the requirements management in SECO. S7 and S17 cite that requirements emerge and propagate through multiple stakeholders' collaboration across a SECO. S2, S4, S6, S26, and S28 point out that in SECO, it is necessary to consider the interaction of several stakeholders to define requirements for their products and platforms. For S6, it is necessary to identify requirements from several customers with different contexts of use and coordinate the work with several teams responsible for the products of the SECO. S8, S20, S21, S24, and S29 highlight that requirements managers must consider external and interdependent stakeholders in SECO. For S21, one of the major differences between distributed teams in GSD and SECO is that the teams or actors are not within the same company or organization but instead spread across several organizations, whereby each actor contributes different elements to the system or product.

C5 focuses on power relations among different multi-party actors that influence requirements management in SECO. However, C5 focuses explicitly on the power relationship between actors in SECO. For S15, companies exercise their legitimate power by proposing new requirements for partners in SECO. S9 mentions that power conflicts are frequent between partners in SECO. S17 cites that multiple stakeholders are susceptible to issues of stakeholder power affecting requirements management activities. Power relations strongly affect stakeholder identification (S17), negotiation (S9, S20, and S21), prioritization (S20, S24), and requirements communication (S17) in SECO. For S9, the notion of power is a common problem affecting RE decision-making. S24 highlights that power relations can result in misalignment with internal RE processes.

#### 4.2.3 Activities

C6 focuses on requirements negotiation activity that impacts requirements management in SECO. S14 highlights that requirements negotiation in SECO is fundamental, as a consensus often is needed to make certain decisions. S2 and S9 mention that requirements negotiation in SECO establishes

an agreement of interests and intentions between multiple stakeholders distributed globally. S5 states that geographic distribution causes the inefficiency of negotiation activities that depend on face-to-face communication or require interaction between teams. S7 justifies that since the stakeholders are multiple and spread across the SECO, the requirements negotiation is complex, as it must consider shared objectives and relationships between the ecosystem actors. S5 points out that these actors have different and sometimes conflicting expectations.

S7 suggests using specific technologies or techniques to address requirements negotiation in SECO. However, S7 mentions that these technologies or techniques do not have an in-depth view of the social dynamics of SECO. For S7, SECO introduce specific requirements negotiation questions, such as aligning interests and expectations among companies and its markets. S13 and S17 point out that emergent stakeholders from across the ecosystem and the relationship with partners are important in requirements negotiation in SECO. For S5, requirements negotiation involves a decision-making process conducted by several different actors along the SECO lifecycle. In traditional software development, requirements negotiation is often less complex, as it usually occurs between a limited number of customers and a limited set of organizations developing the software.

C7 focuses on requirements identification and prioritization that impacts requirements management in SECO. S4, S6, S7, S15, S17, S25, S26, and S29 cite that keystone must consider interacting with several stakeholders to identify requirements. S15 highlights that the requirements are not equally important and must be ranked or prioritized with a value. For S5 and S9, SECO provide products for a broad market, which means that requirements are often jointly defined by the suppliers of the ecosystem rather than elicited from users. S17 points out that traditional RE methods were reported as insufficient to support requirements elicitation in SECO. For S5 and S25, the high number of stakeholders in SECO presents challenges regarding requirements elicitation.

S14 and S24 mention that SECO maintainers (keystones) commonly conduct requirements prioritization, though care is often taken to the opinions of other developers and users. For S5 and S8, requirements prioritization in SECO should only be carried out by relevant stakeholders. According to S21, requirements prioritization can differ highly between SECO actors because different stakeholders value different attributes when dealing with a requirement. Every partner contributes requirements to the ecosystem, which might result in a requirements overload, causing complications in the prioritization process. S24 cites that practices such as requirements identification and prioritization in SECO overlap and are done collaboratively through iterative and transparent discussions and negotiations. S5 highlights that

**Table 6** Illustration of the coding process for the definition of characteristics of requirements management in SECO

<i>C1 (Table 7)</i>	
Category	Concepts/definitions
Code	Requirements management in SECO is open, informal, collaborative, and decentralized
Coherent units	<b>S11:</b> “Changes happen from a centralized to a <b>decentralized management</b> . It also involves new business models to support an <b>open requirements engineering</b> , with a <b>collaborative process</b> where clients suggest and vote for new features” <b>S13:</b> “RE has moved to become more <b>decentralized</b> and <b>collaborative</b> with an evolving set of stakeholders” <b>S14:</b> “Looking at OSS ecosystems, <b>requirements practices are often informal and overlapping</b> ”
<i>C4 (Table 7)</i>	
Category	Actors’ relationships
Code	Multi-party actors (i.e., users, external developers, and others) who are external to the keystone influence requirements management in SECO
Coherent units	<b>S2:</b> “Large-scale organizations need to consider the interplay of a <b>considerable number of stakeholders</b> for defining requirements of their commercial and technical products and product platforms” <b>S28:</b> “Having too <b>many stakeholders to manage</b> completely is a problem that requirements managers of the technological base product of SECOs face” <b>S29:</b> “In a SECO, the keystone has to <b>consider the wishes and needs from not just its own end users, but also its ecosystem</b> ”
<i>C7 (Table 7)</i>	
Category	Activities
Code	Keystone usually conducts the requirements identification and prioritization in SECO collaboratively and involves multi-party actors (i.e., users, external developers, and others)
Coherent units	<b>S5:</b> “The majority of <b>requirements are defined by participants of the ecosystem</b> , rather than elicited from users” <b>S6:</b> “Large-scale organizations need to consider the interplay of a <b>considerable number of stakeholders for defining requirements</b> of their commercial and technical products and platforms” <b>S14:</b> “Requirements are commonly asserted through transparent discussions and suggestions. <b>Prioritization is commonly conducted by ecosystem maintainers</b> ” <b>S24:</b> “Practices such as <b>elicitation and prioritization overlap and are done collaboratively through iterative discussions</b> ”

the requirements definition cannot be disconnected from business processes governing the SECO.

C8 focuses on requirements communication activity that is fundamental to requirements management in SECO. S10 and S17 highlight that the requirements must be communicated inside and outside the SECO. For S5, requirements emerge through communications among the actors of the ecosystem. S20 and S26 cite that CrowdRE supports requirements communication through open and heterogeneous communication channels. According to S2, S6, and S8, a requirements communication network (RCN) enables the analysis and design of requirements communication between multiple stakeholders. RCN describe the stakeholders, the communicated requirements traces, and the structure of a SECO in terms of actors, groups of actors, and negotiation channels.

S4, S21, and S27 highlight using multiple open communication channels to allow transparent communication between the multiple SECO actors. S4 cites that RE via open channels is an innovative process. The studies present discussion forums (S3, S13, S15), problem/issue trackers (S13, S14, S27), mailing lists (S13, S14, S15, S27), ticket/helpdesk systems (S14, S15, S27), and version control systems (S27) as examples of open communication channels. This scenario diverges from requirements management in traditional software development, where the actors are usually well defined,

the view on requirements management tends to be more centralized, and communication channels are generally closed.

#### 4.2.4 Artifacts

C9 focuses on the requirements repositories in SECO. For S8, requirements sharing in SECO must be expanded with relevant and authorized external actors. S14, S19, and S24 mention that there is often no central repository with requirements defined, along with heavy processes and tools for examining the requirements for completeness and consistency in SECO. According to S19, decentralized repositories (i.e., issue trackers, mailing lists, and source-code repositories) store the requirements in SECO. S19 and S24 highlight that in OSSECO, several artifacts can represent a requirement, often complementing each other to give a complete picture, i.e., as an issue, in a mail thread, and/or as a prototype or a finished implementation.

### 4.3 Approaches used to requirements management in SECO (RQ2)

To categorize the approaches identified in the selected studies, we used the taxonomy defined by Shaw [86] and the definitions of types of contributions cited by Paternoster et al. [71] and Dermeval et al. [25]. Shaw’s taxonomy uses

**Table 7** Characteristics of requirements management in SECO

ID	Characteristics of requirements management in SECO	Selected Studies
<i>Concepts/definitions</i>		
C1	Requirements management in SECO is open, informal, collaborative, and decentralized	S3, S11, S13, S14, S15, S16, S17, S19, S20, S22, S24, S29
C2	Emergent requirements originated by multi-party actors affect requirements management in SECO	S3, S13, S17, S29
C3	Requirements management in SECO considers different types of requirements (i.e., platform, integration, and product requirements)	S3, S8, S9, S10, S15, S26, S29
<i>Actors' relationships</i>		
C4	Multi-party actors (i.e., users, external developers, and others) who are external to the keystone influence requirements management in SECO	S2, S4, S6, S7, S8, S17, S20, S21, S24, S26, S28, S29
C5	Power relations among multi-party actors influence requirements management in SECO	S9, S15, S17, S20, S21, S24
<i>Activities</i>		
C6	Keystone usually orchestrates the requirements negotiation in SECO and counts on the participation of multi-party actors (i.e., users, external developers, and others)	S2, S5, S7, S9, S13, S14, S17
C7	Keystone usually conducts the requirements identification and prioritization in SECO collaboratively and involves multi-party actors (i.e., users, external developers, and others)	S4, S5, S6, S7, S8, S9, S14, S15, S17, S21, S24, S25, S26, S28, S29
C8	Multi-party actors (i.e., keystone, users, external developers, and others) use multiple open communication channels for requirements communication inside and outside SECO	S2, S3, S4, S5, S6, S8, S10, S13, S14, S15, S17, S20, S21, S26, S27
<i>Artifacts</i>		
C9	Requirements are not laid out in a central repository, and requirements discussions and implementations are spread across a series of requirements artifacts, each of them with its own repository	S8, S14, S19, S24

some research results in SE, such as procedure or technique; model (qualitative or descriptive, empirical, analytic); tool or notation; or a specific solution (i.e., practice).

Paternoster et al. [71] also use the taxonomy described by Shaw (2003). For Paternoster et al. [71], types of contributions can be divided into weak (advises and implications, lessons learned, tools and guidelines) and strong (theory, framework/method, and model) contributions. Dermeval et al. [25] mention that its classification of approaches (type of contribution) was based on the work of Petersen et al. [72]. Dermeval et al. [25] defines method, model, tool, and process as types of approaches. Finally, we also considered the classifications of approaches defined by the authors of the selected studies. For example, S23 mentions the term “practice” to refer to the approach that the study cites. Table 8 presents the definitions of each type of approach according to Shaw [86].

The results obtained show that requirements management in SECO has been performed with the support of **Tool** (S4, S6, S9, S12, S13, S14, S17, S19, S20, S23, S24, S26, S27,

S29); **Model** (S5, S7, S14, S19, S25, S28); **Method** (S1, S18, S24); and **Practice** (S23). Figure 6 shows that most of the approaches found are related to the use of tools. In the context of RQ2, the selected studies propose or cite the identified approaches. Furthermore, a unique selected study may propose or cite multiple approaches. We present more details about each type of approach in the following subsections.

#### 4.3.1 Tool

S4 presents a qualitative and quantitative investigation of RE in SECO. S4 conducts a case study on SECO IBM Collaborative Lifecycle Management (IBM CLM) and cites two tools related to requirements management in SECO: **Rational Team Concert (RTC)** and **Rational Requirements Composer (RRC)**. S4 also cites **issue trackers**, **discussion forums**, and other **open communication channels** to the public as tools that support requirements management in SECO. S17 also conducts a study related to SECO IBM CLM and demonstrates requirements flow in SECO.

S17 evidences the use of **RTC** and cites that the tool facilitated the discussion of requirements and the creation of bug reports and change requests.

S6 proposes a framework for dealing with various aspects of SECO, called Reuse-based Software Ecosystems Engineering and Management (ReuseSEEM). The study highlights the need to create tools that map socio-technical networks to manage the needs, platform requirements, and SECO products and services. For S6, these tools allow communication and requirements management based on community participation, suggesting and solving customer needs on an **extended social networking site**.

S9 presents a case study of two software companies participating collaboratively and competitively in an SECO. One of the interviewees mentioned that one of the companies used the **One Studio tool** to record requirements change. Furthermore, client company stakeholders used a **web-based request system** to describe their new demands. Requests were automatically forwarded to the helpdesk service, which selected or discarded the requests. Then, company teams analyzed these demands and tracked their status in the tool, from registering the demand to fulfilling the requirements in a future release.

S12 presents a case study to analyze RE challenges in a specific SECO (AUTOSAR Ecosystem). One of the interviewees from the study pointed out that in their new project, they have started using an **issue tracker** for requirements management. He believed using the **issue tracker** has facilitated requirements management and communication. S13, S14, S19, S20, S24 S26, S27, and S29 mention **issues trackers**, **discussion lists**, and a **version control system** as tools that support requirements management in SECO. S23 and S27 also cite **requirements portals**, S29 cites **Stack Overflow** and **social media (such as Twitter and LinkedIn)**, and S24 mentions **source-code repositories**, and **code reviews** to support the requirements management in SECO.

#### 4.3.2 Model

S5 and S7 propose a **requirements negotiation model** in the SECO context to enrich the requirements management process. The proposal is associated with the management of the software platform and seeks to establish requirements for negotiation activities in this social context. S14 presents the **Requirements Analysis and Management for Benefiting Openness (RAMBO) model**. This model is used for requirements analysis and management to benefit open innovation. The model presents how the main stages of RE and requirements management processes can be adjusted to benefit from the new context's openness.

S19 presents the model called **Contribution Acceptance Process (CAP)**. It aims to help motivate the contribution of companies in OSSECO. Its operationalization in

the organization is performed through a meta-model that integrates the requirements management infrastructure so that the contribution strategies of the primary model can be communicated and tracked. The meta-model enables inspiration and guidance on how developer organizations should implement necessary adaptations to their requirements management infrastructure or create a new one.

S25 and S28 propose a **reference process model to identify key stakeholders as part of requirements elicitation in SECO**. According to S25, this model can be applied by requirements managers of the common technological platform of SECO to design a business process model to identify stakeholders. In this model, requirements managers identify stakeholders they already know and document their roles in SECO. After that, they open lines of communication with the identified stakeholders (or primary stakeholders) and ask them to identify others (secondary stakeholders). S28 suggests that requirements managers of the keystone use the reference process model in their activities. The model will provide a structured approach to identifying stakeholders in complicated environments. For S28, the existing stakeholder identification methods in the literature do not take a comprehensive approach to consider all stakeholders while pinpointing only a subset from whom to elicit requirements and also with approaches that preserve SECO health, especially in SECO that include proprietary software.

#### 4.3.3 Method

S1 mentions three ecosystem elements that constitute agile methods: barely sufficient methodology, collaborative values, and chaordic perspective. Thus, the more volatile the requirements are, and the more experimental the technology, the more agile approaches increase the chances of success in SECO. The study cites that **agile methods** hold significant promise to deliver greater value to all main SECO actors. For S1, agile methods in the SECO context can help in (i) technical issues: requirements and tests management, and (ii) organizational issues: process adaptation, knowledge sharing, transfer, and culture change.

S1 highlights the need for new approaches to requirements elicitation in SECO to ensure that the voice of the customer is accurately captured. S1 cites goal-oriented brainstorming and workshops as potential strategies to support agile requirements elicitation. Moreover, S1 suggests that visual specifications (high-fidelity prototypes) can be used as partial replacements for exclusively textual specifications, thus increasing the emphasis on requirements assessment. S1 also cites the user story-driven development for requirements management in SECO. According to S1, **agile methods** mitigate the impact of emerging changes between systems and embedded software that can unexpectedly increase the cost and duration of the development process.

**Table 8** Description of types of approaches

Type of approach	Description by Shaw [86]	Description in our work
Tool	Implemented tool that embodies a technique; formal language to support a technique or model (should have a calculus, semantics, or other basis for computing or doing inference [86])	“Tool” provides (semi-) automated support for some technique, method, or model to support requirements management in SECO
Model	(a) <b>Descriptive model:</b> Structure or taxonomy for a problem area; architectural style, framework, or design pattern; non-formal domain analysis, well-grounded checklists, well-argued informal generalizations, guidance for integrating other results, well-organized interesting observations; (b) <b>Empirical model:</b> Empirical predictive model based on observed data; and (c) <b>Analytic model:</b> Structural model that permits formal analysis or automatic manipulation [86]	“Model” can be defined as a representation or abstraction of the activities characterized in the requirements management process in SECO
Method	New or better way to do some task, such as design, implementation, maintenance, measurement, evaluation, selection from alternatives; includes techniques for implementation, representation, management, and analysis [86]	“Method” is the procedure, technique, or way of doing some activity related to the requirements management process in SECO
Practice	Shaw [86] does not define “practice”	“Practice” is a less prescriptive way of doing some activity related to the requirements management process in SECO

S18 presents the results of an investigation into the effects of SECO in two software consumer organizations that conducted information technology management activities. In one of the cases, the study reveals that the demand management team faced challenges related to requirements management concerning the frequent “re-prioritization” of the project portfolio due to external issues. An **agile method** was adopted to mitigate this challenge. The study does not provide further details on use. However, it is possible to figure out that agile methods have been used to support requirements management in SECO.

S24 proposes the **stakeholder influence analysis (SIA) method**. SIA aims to help companies analyze an OSSECO to identify its stakeholders’ influence by their impact on the requirements implemented. The method SIA was based on social network analysis constructs that have proven useful in characterizing the influence of stakeholders but also effective when analyzing a firm’s participation in OSSECO and requirement-centric stakeholder collaborations.

#### 4.3.4 Practice

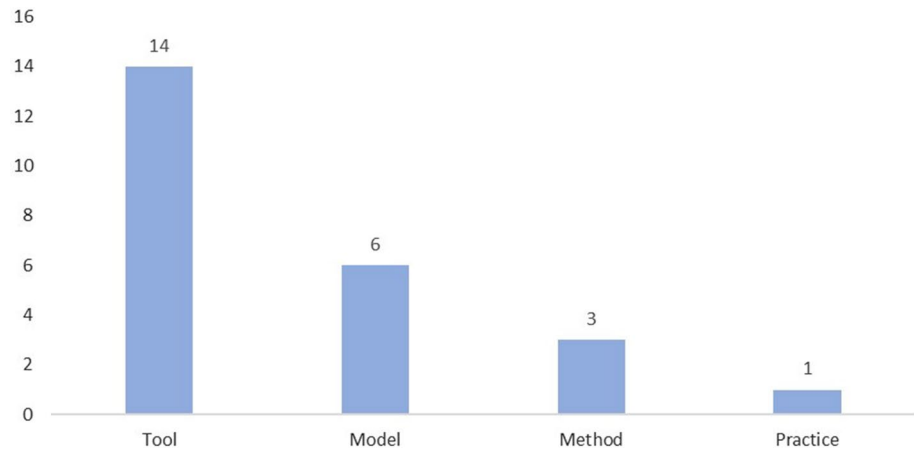
S23 presents the Software Ecosystem Governance Maturity Model (SEG-M2). This model seeks to evaluate and classify, from the keystone, various elements of SECO in terms of governance. The model is divided into seven focus areas concentrating on a set of practices. S23 points out a practice related to requirements management, which is **requirements sharing**. This practice is evidenced at level 3 of the model when adopting a requirements portal (cited in Sect. 4.3.1) is recommended. Furthermore, the study mentions that implementing an open requirements management system would face technical and cultural problems in the organization. A third-party system interfaced with the company is already used internally through an application programming interface (API) and manual data copying.

In the context of RQ2, we identified the approaches to support requirements management in SECO that were proposed or cited in the selected studies. Hence, we noticed that some requirements management tools used in traditional software development also are used in the SECO context. However, the selected studies highlight several difficulties in using the SECO context. Additionally, when it comes to models, methods, or practices, none focus specifically on requirements management in SECO. The studies only cite that these approaches affect or benefit the requirements management in SECO and do not present much detail.

#### 4.4 Assessment of approaches used to requirements management in SECO (RQ3)

RQ3 aimed to identify if (and how) the approaches to requirements management in SECO identified in RQ2 were

**Fig. 6** Types of approaches to support requirements management in SECO



assessed. We classified the selected studies into (i) studies that do not cite or propose approaches (S3, S8, S10, S11, S15, S16, S21, S22), (ii) studies that only cite approaches (S1, S4, S9, S12, S13, S17, S18, S20, S26, S27, S29), and (iii) studies that propose approaches to requirements management in SECO (S2, S5, S6, S7, S14, S19, S23, S24, S25, S28). We identified multiple approaches in the same study (e.g., in S24, we identified a tool and method). We also identified that some selected studies (S14, S19, S23, S24) propose a certain approach while only citing others. Thus, we classified these studies as (iii). To answer RQ3, we considered only selected studies that proposed approaches (iii).

Among the selected studies that proposed approaches (iii), we classified them into (a) studies that present assessment (S19, S23, S24) and (b) studies that do not present assessment (S2, S5, S6, S7, S14, S25, S28). Among those that do not present an assessment, some mention the need for future assessment (S2, S6, S7, S25, S28).

We use the empirical research methods in SECO described by Abdullahi et al. [3] and empirical research methods in SE described by Wohlin et al. [100] to categorize our results. Abdullahi et al. [3] enumerate nine research methods used in SECO: (a) Case study; (b) Mixed methods; (c) Design science; (d) Survey; (e) Longitudinal study; (f) Experimental study; (g) Exploratory study; (h) Interview; and (i) Repository mining. Wohlin et al. [100] enumerate four research methods used in SE: (a) controlled experiment; (b) case study; (c) survey; and (d) postmortem analyses.

S19 defines the model described from the collaboration of Sony Mobile professionals. However, the authors conducted three exploratory **case studies** to evaluate it, verifying its usability and applicability. The companies that participated in the validation were: (i) a small company that creates a platform product in the agricultural domain, the Chief Technical Officer (CTO) was interviewed; (ii) a small business that creates games for mobile platforms, the founder was interviewed; and (iii) a large company in the field of telecommunications, a workshop with 8 participants was

held. From the three evaluations, the study concludes that the model can be applied to a set of features, a product, or a complete project.

S23 assesses in six **case studies** with professionals from the following companies and/or platforms the model and practices presented: (i) platform for augmented reality that provides API through an extension from Unity; (ii) enterprise resource planning company; (iii) credit management company; (iv) network equipment manufacturer; (v) platform specifically designed to facilitate the ecosystem of the company mentioned in item iv; and (vi) a business-to-business platform that allows telecommunications providers to build their ecosystems. First, they applied the model to companies and assessed whether it was valid, functional, and effective. The overall conclusion of the evaluation showed that the model is a valuable collection of practices that enable decision-making regarding governance in SECO.

S24 conducted a **case study** on the Apache Hadoop OSSECO. S24 applied a proof of concept to demonstrate that the model is functional and practical. The Apache Hadoop OSSECO was chosen due to the high concentration of firms in the ecosystem and because it is the Apache project with the highest number of committers. The case study further helped to evolve and refine the model and its seven steps.

## 5 Discussion

The results presented in Sect. 4 provided some insights that will be discussed in this section. Initially, we discuss the results obtained with the RQ responses with the RE and SECO literature in other contexts. Next, we discuss how other RE activities influence or are influenced by requirements management in SECO. Finally, we discuss why the selected studies find requirements management challenging in SECO.



## 5.1 Characteristics and approaches of requirements management in SECO

In Sect. 4.2, we identified nine characteristics of requirements management in SECO. Some of these characteristics relate to challenges and concepts emerging from RE. C1 to C3 and C9 refer to the openness and informality of requirements management in SECO, among other aspects. According to S16, SECO are inherently open, to some extent, which influences requirements management activities. Openness in RE processes is also discussed in the context of open-source software (OSS) projects, which can form OSSECO. Iyer and Lytinen [43] point out that RE processes in OSS projects are less formal and reliant on online documentation and communication tools.

We notice that the open innovation paradigm is explored in requirements management in SECO. Open innovation has been linked to both SECO and RE in the literature. Hanssen and Dybå [38] cite that the emergence of SECO relates to the inherent potential for open innovation. Linåker et al. [61] highlight that RE needs to consider the implicit changes in open innovation and adapt to this new context. Yin and Pfahl [105] cite that the application of open innovation strategies in RE is little investigated and that there is no support from proprietary tools for open innovation in RE.

We look deeper at openness and open innovation in requirements management in SECO and identify their relationship to emerging concepts in RE. S20 and S22, for example, discuss the ubiquitous RE in the context of digital transformation involving the formation of SECO. S20 and S22 envision RE in multiple dimensions (i.e., RE everywhere, RE with everyone, RE for everything, automated RE, open RE, and cross-domain RE), resulting in the digital transformation of RE itself into ubiquitous RE. In these dimensions, S20, S22, and S26 highlight the importance of the CrowdRE paradigm for requirements management in SECO. CrowdRE refers to an emerging paradigm that promotes the active involvement of a “crowd” of stakeholders, including current and potential users of a software product [36].

S26 mentions that RE in SECO involves a crowd that includes different types of users, external developers, and the keystone. According to S26, CrowdRE in SECO brings new opportunities and challenges for keystones and its partners. S22 highlights that traditional requirements elicitation techniques, such as interviews or focus groups, present scalability problems in SECO. For S22, the CrowdRE paradigm addresses typical problems experienced in RE in SECO (i.e., involvement of multiple actors and the lack of scalability of RE) using global automation to perform with a crowd. According to S20, in heterogeneous and dispersed settings that include a crowd, such as SECO, allowing anyone to provide potential requirements can be more effective than involving only a selection of representative end users or

applying personas. S20 states that CrowdRE help address RE challenges in a social context such as SECO.

Current works have investigated CrowdRE in different contexts. Abdullah et al. [1] present research efforts and challenges in CrowdRE and conclude that researchers in CrowdRE continually explore the research area and propose new techniques and tools to improve it. Tizard et al. [90] state that researchers found relevant information about requirements in feedback from multiple online communication channels, including app stores, social media, and product forums. For Wouters et al. [102], CrowdRE expands the reach of established RE approaches, which involve a selected sample of stakeholders, extending the notion of market-driven RE toward the democratic participation of users in RE. Gómez et al. [37] demonstrate that a SECO designed to leverage different types of crowds greatly benefits all actors in the ecosystem. The growing trend in CrowdRE research indicates that this field has more room to be improved and explored, including the context of SECO.

C4 to C8 highlight the existence of geographically distributed multi-party actors groups that influence requirements management in SECO. Geographical distribution has been studied in the context of the GSD. Ali and Lai [7] cite that organizations collaborate with software teams worldwide in GSD. GSD collaboration can be inter-organizational, intra-organizational, and of mixed nature (inter+intra). The authors indicated a significant difference between the number of intra and inter-organizational collaborations due to management and communication processes, which often differ in inter-organizational setups. S21 identifies one of the main differences between teams distributed in traditional organizations and SECO. In SECO, the teams or actors are not within the same company or organization but are spread across several organizations. Each actor contributes different elements to the system or product.

In RQ2, we identified approaches to support requirements management in SECO, and the most cited was “Tool.” Several selected studies mentioned using “issue trackers” as a tool to support requirements management in SECO. CrowdRE explores the “issue trackers” as a source of feedback. Ali Khan et al. [8] propose a CrowdRE framework. The framework can be applied in issue trackers to identify or respond to critical issues. Cleland-Huang [20] mentions that the use of digital technologies as tools have brought about disruptive changes in all domains, widely known as “digital transformation.” For Cleland-Huang [20], disruptive change is a powerful force that explodes on the scene, introducing a new dominant solution or an unprecedented practice.

Regarding methods in RQ2, we identified that agile methods had been used to support requirements management in SECO. Cleland-Huang [20] highlights that agility is just one example of a disruptive change that significantly impacts RE. Disruptive changes bring unprecedented opportunities

for innovation. In addition, the author argues that the widespread adoption of agile methods across multiple project domains threatens many of the existing true practices associated with requirements management. However, for the author, there is an industrial tendency to move away from “traditional” requirements processes in the digital transformation scenario.

Jayatilleke and Lai [46] differentiate requirements change management between traditional and agile software development. In addition, they cite challenges for these two contexts. In our work, we noticed that there are initiatives to use agile methods to help manage requirements in SECO, despite the additional disruptions in new ways of software development and emerging contexts such as SECO. Hence, when discussing requirements management challenges in SECO, we relate them to some challenges identified by Jayatilleke and Lai [46]. For example, we highlight the challenges related to requirements communication are different in SECO. We also note that the analysis of how organizations make decisions presented by Jayatilleke and Lai [46] focuses on the context of a single organization. In the results of our work, we extend the analysis to a requirements management context that goes beyond the limits of organizational boundaries, as happens in SECO.

Santos and Werner [82] mention that SECO emerged to improve software reuse in the GSD industry, considering the relationships between companies and stakeholders worldwide. The authors also observe that the SECO vision has motivated the leading players in the sector to rethink their operational actions to open their platforms to external agents and keep their businesses alive. S29 points out that success no longer depends solely on keystone’s efforts to manage end user expectations of its offering. Instead, it works with a more complex set of business relationships within the SECO. The situation becomes even more complex when these relationships include collaboration between competitors, that is, a state of competition. S21 points out that in SECO, several types of relationships exist between actors, which can be competitors or share mutual benefits about requirements. The complexity of relationships and dependencies increases with the number of stakeholders in the ecosystem.

## 5.2 Requirements engineering activities related to requirements management in SECO

Vegendla et al. [96] investigated RE activities in SECO and divided RE into five activities: requirements elicitation, requirements analysis, requirements specification, requirements validation, and requirements management. In our SMS, we characterize requirements management in SECO. We use the definition of ISO/IEC/IEEE [41], which considers requirements management an organized process of

documentation, analysis, traceability, prioritization, change control, and communication of requirements. However, when answering our RQ, we noticed that other RE activities influence or are influenced by requirements management in SECO. Table 9 presents the citation frequency in the selected studies of other RE activities related to requirements management in SECO.

### 5.2.1 Requirements elicitation

S25 focuses on one aspect of requirements management in SECO, identifying stakeholders as a preliminary step of requirements elicitation. S25 considers that the first step of requirements management is requirements elicitation, but it is imperative to know from whom the requirements should be elicited before that. According to Lim et al. [56], digital transformation has spawned new opportunities for requirements elicitation in different contexts. The authors highlight that currently, requirements elicitation can consider an unprecedented and increasing amount of high-velocity and heterogeneous data. Vegendla et al. [96] highlight that the platform’s openness in SECO allows everyone to provide the requirements without role identification.

According to Vegendla et al. [96], requirements elicitation involves identifying SECO members’ requirements according to their business goals and can be considered a challenging issue. Fricker et al. [33] mention that large organizations need to consider the interaction of several actors to define the requirements of their products and commercial platforms. However, conventional requirements elicitation techniques are not sufficiently scalable or capable of considering stakeholder groups that are becoming increasingly large and global [56, 91]. For S1 and S7, requirements elicitation requires new approaches that must provide mechanisms to capture the customer’s voice. We consider that the scalability that SECO requires in requirements elicitation approaches also affects requirements management, mainly concerning change control and traceability.

### 5.2.2 Requirements analysis

Our results show that requirements management and analysis strongly relate to SECO. Vegendla et al. [96] cite that requirements analysis and management should be conducted in parallel to handle the conflicts and ambiguity in the requirements. However, S9 mentions that companies participating in SECO perform requirements analysis differently. Alsanoosy et al. [9] mention the influence of culture on requirements analysis. For the authors, each culture relies on its context to interpret and understand information. Finally, Oriol et al. [70] state that SECO health objectives are achieved as part of a requirements analysis and refinement process with the stakeholders.

**Table 9** RE activities in SECO related to requirements management in SECO

RE activity	Selected studies
Requirements elicitation	S1, S3–S5, S7–S9, S12–S17, S19, S20, S22, S24–29
Requirements analysis	S1, S2, S5, S7, S9, S11–S14, S17, S20, S26, S27
Requirements specification	S1, S2, S6, S9, S12, S18–S20, S24
Requirements validation	S12, S14, S17, S20

We consider that requirements management and analysis must jointly and continually control changes and requirements status considering the open and dynamic environment of SECO. Nonetheless, S14 points out that the challenge of requirements analysis in SECO is the analysis of requirements with openness in mind. For the authors, from this broader perspective, requirements analysis should calculate the probability of completion if someone else in the ecosystem contributes to co-development or analyzes optimistic or pessimistic scenarios with or without other contributions within the same SECO.

### 5.2.3 Requirements specification

S24 points out that multi-party actors dispersed geographically collaboratively conduct the requirements specification at SECO. Vegendla et al. [96] cite several works on SECO modeling related to the requirement specification and highlight the importance of GSD in SECO. In this scenario, S19 mentions the difficulty of specifying requirements linked to different repositories of software artifacts within a global requirements management infrastructure in SECO. For Ali and Lai [6], social, linguistic, geographic, and cultural differences in GSD make it challenging to understand poorly written requirements specifications by introducing pauses and delays in communication. We consider that this is not different in the SECO context. However, we emphasize that SECO's open and inter-organizational environment makes this scenario even more difficult.

S18 mentions the need to develop formal requirements specifications that consider the specificities of the ecosystem platform to assist in decision-making in SECO. Shahzad et al. [85] state that it is necessary to consider the properties of ecosystems, such as sustainability, contribution, and incentives in requirements elicitation and specification. The authors justify that people usually have requirements for socio-technical systems, and such a system consists of several different actors (e.g., other people and machines). In addition, according to the authors, most socio-technical systems are operated for an extended period, and they should be improved and sometimes adapted to the current situation.

### 5.2.4 Requirements validation

S14 highlights that informality facilitates social interaction in requirements management in SECO and supports requirements validation. S20 mentions that using crowdsourcing techniques that include crowd members was essential for collaborative and systematic requirements validation in SECO. Vegendla et al. [96] cite that there are few studies on requirements management and validation in SECO. Nevertheless, for the authors, without validation and management of the requirements, the consistency and completeness of the requirements and models and the changes in the requirements are not ensured. For Kamalrudin and Sidek [48], the requirements validation process in the context general of software development needs to consider consistency, completeness, and correctness for producing software specifications.

Cysneiros and Zisman [23] claim that requirements traceability, an activity of requirements management, supports requirements validation in traditional software development. In turn, Vegendla et al. [96] state that requirements traceability is a significant task to provide requirements changes in SECO. However, according to the authors, more research should be performed for requirements validation and traceability in SECO. Che and Perry [19] state that SECO need efficient automated traceability between system drivers (such as requirements, business, and market needs) and architecture design decisions. Nonetheless, the authors also point out that extensive research is still required on architectural knowledge traceability to SECO.

## 5.3 Challenges of requirements management in SECO

During data extraction and synthesis to answer RQ1, we found that 16 of the 29 selected studies in our SMS (S2, S4–S7, S9, S14, S15, S17, S20, S21, S25–S29) consider requirements management (or some related activity) a challenge in SECO. Thus, we realized that the main challenges pointed out by the selected studies are related to the characteristics of requirements management in SECO (Table 7). Table 10 presents the relationship between the identified challenges and the characteristics of requirements management in SECO. Table 11 illustrates some examples of coherent units we define to answer RQ1. The rest of the

**Table 10** Relationship between challenges and characteristics of requirements management in SECO

Challenge	Characteristic in Table 7	Selected studies
Requirements communication	C8	S2, S4, S5, S6, S15, S20, S21, S26, S27
Requirements identification	C7	S5, S14, S25, S28
Multi-party actors	C4	S5, S17, S21
Openness	C1	S14, S17, S29
Requirements negotiation	C6	S5, S7, S9
Emergent requirements	C2	S17, S29

coherent units are available in the RQ1 complete coding process, which is available at ZENODO.<sup>11</sup> Then, we detail and discuss the challenges.

S5 mentions that the SECO approach brings new challenges to RE. S6 states that **requirements communication** and management in SECO are challenging. For S2, S5, S6, and S20, requirements communication is challenging, especially when multiple actors must collaborate in dispersed locations in the GSD scenario. In the GSD context, Qureshi et al. [73] mention several challenges related to requirements communication as “inadequate use of informal communication” and “inadequate transfer of information.” Several works consider the requirements communication between stakeholders in distributed geographic locations as a challenge [2, 11, 46, 103]. In addition, S4 points out that requirements need to be communicated and understood by all relevant stakeholders in SECO and identifies this as a challenge. S15 reports that there are challenges in establishing centralized requirements communication in SECO. For S21, S26, and S27, requirements communication in SECO is challenging due to multiple communication channels.

S14 cites challenging aspects that open innovation brings to **requirements identification** in SECO. For S5, S25, and S28, the number of SECO stakeholders may become too large, which presents a challenge for requirements identification. Lim et al. [56] highlight that conventional elicitation techniques are often time-consuming and not sufficiently scalable for considering stakeholder groups that are becoming increasingly large and global. For Lim et al. [56], CrowdRE is an excellent example of dealing with this challenge. The authors cite that the primary focus of CrowdRE is the requirements identification from explicit crowd users’ feedback (e.g., app reviews and data from social media) by applying various techniques based on machine learning and natural language processing.

S5 cites that the existence of **multi-party actors** introduces challenges to requirement management in SECO. S17

points out that connecting the right actors to clarify requirements in SECO is difficult. S21 mentions that the coordination between multi-party actors has already been identified as a significant challenge in agile and large-scale distributed teams. However, for S21, this challenge cannot be taken similarly in SECO because each actor has a way of working that can hardly be standardized. This results in several different practices being applied across the ecosystem. In addition, S21 highlights that it is necessary to consider that the actors in SECO do not belong to a single organization but to many organizations that share different, possibly competing relationships (coopetition). Roth et al. [74] explore open coopetition, i.e., open innovation cooperation between competitors that includes third parties such as networks, platforms, communities, ecosystems, or triple helices. The authors suggest using open innovation management principles to deal with open coopetition.

S14, S17, and S29 mention that **openness** brings RE-related challenges in SECO. S14 cites the challenge of analyzing requirements with openness in mind. For S14, in an open perspective, the requirements analysis should calculate the probability of completion of the implementation of the requirements if someone from the ecosystem contributes in co-development or analyzes optimistic or pessimistic scenarios with or without other contributions within the same ecosystem. S17 points out that openness brings challenges related to requirements engineering in SECO, such as managing dynamic and emerging contributions from ecosystem stakeholders and collecting their input while protecting their intellectual property. Zaggl et al. [106] ensure that finding the right degree of openness is challenging, especially in complex open innovation ecosystems. For the authors, being too open involves the risk of imitation and slippage of values. Furthermore, users can gain a lot of influence on corporate decisions, leading to the challenge of requirements negotiation.

S5 points out that the SECO approach makes requirements negotiation difficult. S7 considers that the requirements negotiation in SECO is a relevant challenge for the social dimension of an ecosystem and states that it directly affects the success of a SECO. For S9, interacting with multiple partners in a SECO introduces challenges to requirements negotiation. Abdullahi et al. [2] cite the need for SECO actors to reach agreements effectively through requirements negotiation. However, for the authors, there is little understanding of how stakeholders make decisions in requirements engineering. Abdullahi et al. [2] also state that decision-making is the most challenging and difficult task in requirements negotiation in SECO. According to the authors, when making a decision, several challenges need attention before starting the negotiation. The authors cite some of them as the interaction between multiple actors in a SECO and the existence of incompatible objectives of

<sup>11</sup> <https://doi.org/10.5281/zenodo.8131745>.

**Table 11** Illustration of the process for defining challenges of requirements management in SECO

<p><i>Challenge related to C7 (Table 7)</i></p> <p>Challenge</p> <p>Coherent units</p>	<p>Requirements identification in SECO</p> <p><b>S5:</b> “SECO approach impacts on the traditional Software Engineering (SE) models and brings new challenges to Requirements Engineering (RE). By putting together multiple and dispersed actors, <b>SECO approach hampers requirements elicitation</b>”</p> <p><b>S25:</b> “The number of stakeholders may become too large to manage completely. This presents challenges in the case of <b>requirement elicitation</b>”</p>
<p><i>Challenge related to C8 (Table 7)</i></p> <p>Challenge</p> <p>Coherent units</p>	<p>Requirements communication in SECO</p> <p><b>S2:</b> “<b>Requirements communication</b> is challenging, in particular when people and organizations need to collaborate over dispersed locations”</p> <p><b>S6:</b> “<b>Requirements communication</b> and management in SECOs is a challenge, especially in the distributed software development scenario”</p> <p><b>S26:</b> “We describe open challenges for SECOs in the context of CrowdRE. <b>New communication mechanisms</b> are needed that scale with the crowd”</p>

these actors that require careful negotiation to avoid further conflicts.

S17 and S29 mention that managing **emergent requirements** in SECO is challenging. For S17, it is difficult to map emergent requirements to specific actors in SECO. Ailane et al. [5] cite challenges regarding emergent requirements and propose a new attempt to define the emergence phenomenon from a SE perspective. According to the authors, the emergence phenomenon is an important phenomenon that is difficult to formalize and standardize in complex systems (such as SECO).

## 6 Threats to validity

This section presents potential threats to the validity of this study and our actions to mitigate them. The discussion builds on the guidelines for addressing threats to the validity of secondary studies provided by Ampatzoglou et al. [10]. Threats in this context can be classified as study, data, and research validity.

### 6.1 Study validity

Threats in this category are directly related to the study selection process. Therefore, they may lead to the omission of relevant studies or the inclusion of irrelevant studies. This potential threat includes (i) limitations in search strategy; (ii)

ineffective search strings; and (iii) bias in the selection of studies that were selected in this SMS.

To mitigate (i), we applied a full-fledged hybrid search strategy (DBS + snowballing via BS and FS). Mourão et al. [67] mention that hybrid search strategies are efficient compared to the simple DBS strategy. According to Wohlin et al. [101], a full-fledged hybrid search strategy is better than other alternative hybrid strategies. In addition, we run searches in seven digital libraries, including Scopus, which index works from several other digital libraries. Then, we selected 29 studies in our SMS. Therefore, in this category, we emphasize the limitation regarding the generalization of the results. To mitigate (ii), we used two control studies to evaluate the quality and comprehensiveness of our search string. To mitigate (iii), our study had the participation of four researchers, two of them with more than ten years of experience in conducting secondary studies, which helped in conducting the activities. Two researchers applied the IC and EC independently, and two more experienced researchers evaluated the results. Regarding disagreements, a consensus was sought between the parties.

### 6.2 Data validity

The data extraction and synthesis process can threaten the data’s validity, as it could lead to dubious results and conclusions. This potential threat includes (i) unverified data extraction and (ii) researcher bias during the data extraction

and synthesis. To mitigate (i), we documented all data transformations so that it is possible to trace the synthesis back to the corresponding selected study. Concerning (ii), two researchers performed data extraction independently. Afterward, the same two researchers performed a joint analysis, and finally, they condensed the data. Later in this process, two other more experienced researchers verified the result.

### 6.3 Search validity

The lack of auditing and the difficulty in replicating the research threaten the validity of the research. These threats are related to (i) lack of documentation on study planning and (ii) lack of justification for changes made during the conduct of the study. To mitigate (i) and (ii), we defined a detailed protocol based on well-established guidelines of Kitchenham and Charters [50]. Finally, we analyzed, justified, and documented all necessary changes that occurred during the conduct of this study.

## 7 Research agenda

Motivated by the results of this SMS, we suggest a research agenda inspired by the work of Santos et al. [75] and Santos and Carvalho [76], where some issues raised during the SMS can investigate in the future:

1. How can CrowdRE, open innovation, and ubiquitous RE support requirements management in software ecosystems?
2. How can the different requirements (product, platform, and integration) be managed for different types of software ecosystems (open, proprietary, and hybrid software ecosystems)?
3. Are the requirements management approaches identified in this SMS sufficient to support the problems and specificities of different types of software ecosystems (open, proprietary, and hybrid software ecosystems)? How to assess or adapt them?
4. How to carry out requirements traceability in software ecosystems considering the characteristics of requirements management in software ecosystems?
5. How to monitor emergent requirements flows in software ecosystems and manage such requirements in conjunction with other requirements?
6. How to perform the requirements change control in software ecosystems considering the changes requested in multiple open communication channels by multi-party actors?

## 8 Conclusion and future work

In this study, we characterize requirements management in the SECO context. We define specific characteristics (process elements, i.e., concepts/definitions, activities, actors' relationships, and artifacts) that differentiate requirements management in SECO from requirements management in traditional software development. In addition, we identify approaches (types of contributions, i.e., methods, techniques, tools, and practices) to support requirements management in SECO. To do so, we conducted an SMS to identify studies related to the topic in the literature.

From a total of 1028 studies retrieved in DBS and 1458 studies retrieved in snowballing, we selected 29 studies for data extraction. Of the 29 selected studies, 25 contributed to defining characteristics of requirements management in SECO, and 21 cited/proposed one or more support approaches to requirements management in SECO. Only ten of 21 studies proposed a specific approach, and only three presented the evaluation.

We defined nine characteristics of requirements management in SECO that differentiate it from requirements management in traditional software development. Requirements management at SECO is open, informal, collaborative, and decentralized. It is influenced by geographically dispersed multi-party actors affected by power relations. We emphasize that some of these characteristics may have already been identified in other software development contexts, such as OSS, GSD, and agile development. However, we emphasize that the SECO context differs mainly because it allows a holistic view of an ecosystem formed by different projects, products, networks of actors, and artifacts. Thus, in SECO, it is possible for multi-party actors geographically distributed to conduct OSS projects and to implement different development practices, including agile development.

Among the found approaches, we identified that tools (14 studies) and models (6 studies) were the most frequent in the selected studies. However, we observed that most of the approaches identified were not focused only on requirements management. This can happen due to the transversal character of requirements management in RE activities. Case studies assessed a few of these approaches (3 studies). In addition, we identified different types of SECO in the studies selected in our SMS (i.e., automotive SECO, MSECO, OSSECO), which highlights the scope of this terminology for several scenarios.

Finally, based on the knowledge built from this SMS, we defined a research agenda that presents research topics that can help understand requirements management activities in SECO and the approaches used for their realization. As a result, in future research, we will investigate in a more directed way the different requirements flows and

their characteristics, considering the different dimensions of an SECO. Furthermore, after investigating the requirements flows, we can propose a specific approach to support requirements management in the SECO context.

## Appendix A: List of selected studies

ID	Title	References
S1	'State of the Art' in Using Agile Methods for Embedded Systems Development	Srinivasan et al. [89]
S2	Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems	Fricker [32]
S3	The metropolis model and its implications for the engineering of software ecosystems	Kazman and Chen [49]
S4	Towards Understanding Requirements Engineering in IT Ecosystems	Knauss et al. [52]
S5	Analysing Requirements Negotiation in Software Ecosystems with Multi-Agent Systems Techniques	Valença et al. [92]
S6	On the Impact of Software Ecosystems in Requirements Communication and Management	Santos and Werner [80]
S7	Requirements Negotiation Model: A Social Oriented Approach for Software Ecosystems Evolution	Valença [94]
S8	Software Ecosystems: From Software Product Management to Software Platform Management	Jansen et al. [45]
S9	Competition and Collaboration in Requirements Engineering: A Case Study of an Emerging Software Ecosystem	Valença et al. [95]
S10	Innovation Processes in Business Ecosystems Creating a Common Understanding by Requirements	Harland et al. [39]
S11	Using Social Networks to Support Software Ecosystems Comprehension and Evolution	Santos et al. [77]
S12	Cross-Organizational Challenges of Requirements Engineering in the AUTOSAR Ecosystem: An Exploratory Case Study	Soltani and Knauss [87]
S13	How Firms Adapt and Interact in Open Source Ecosystems: Analyzing Stakeholder Influence and Collaboration Patterns	Linåker et al. [58]
S14	Requirements Analysis and Management for Benefiting Openness	Linåker and Wnuk [62]
S15	Understanding How Power Influences Business and Requirements Decisions in Software Ecosystems	Valença and Alves [93]
S16	A Survey of Practitioners Use of Open Innovation	Fernandez and Svensson [28]
S17	Continuous clarification and emergent requirements flows in open-commercial software ecosystems	Knauss et al. [53]
S18	Ecosystems effects on software-consuming organizations an experience report on two observational studies	Santos et al. [81]
S19	Motivating the contributions An Open Innovation perspective on what to share as Open Source Software	Linåker et al. [63]
S20	Towards Ubiquitous RE: A Perspective on Requirements Engineering in the Era of Digital Transformation	Villela et al. [97]
S21	Scaling Agile Beyond Organizational Boundaries - Coordination Challenges in Software Ecosystems	Figalist et al. [29]
S22	Ubiquitous Requirements Engineering: A Paradigm Shift That Affects Everyone	Villela et al. [98]
S23	A focus area maturity model for software ecosystem governance	Jansen [44]
S24	A method for analyzing stakeholders' influence on an open source software ecosystems requirements engineering process	Linåker et al. [59]
S25	Identifying Key Stakeholders as Part of Requirements Elicitation in Software Ecosystems	Lewellen [54]
S26	Open CrowdRE Challenges in Software Ecosystems	Johnson et al. [47]
S27	Public Sector Platforms going Open: Creating and Growing an Ecosystem with Open Collaborative Development	Linåker and Runeson [60]
S28	A comprehensive approach to identifying key stakeholders in complicated software ecosystems	Lewellen [55]

ID	Title	References
S29	Challenges and Strategies for Managing Requirements Selection in Software Ecosystems	Damian et al. [24]

**Acknowledgements** This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors thank UNIRIO (PPQ 03/2022 and 03/2023) and FAPERJ (Proc. 211.583/2019) for partial support to the work. The third author would like to thank the FAPEMA (Process BEPP-01608/21; UNIVERSAL00745/19).

**Data availability** The datasets generated during and/or analyzed during the current study are available in the ZENODO repository, <https://doi.org/10.5281/zenodo.8131745>.

## Declarations

**Conflict of interest** The authors declared that they have no conflict of interest.

## References

1. Abdullah R, Ahmad S, Asmai S et al. (2021) Research efforts and challenges in crowd-based requirements engineering: a review. *Int J Adv Comput Sci Appl* 12(9):395–402. <https://doi.org/10.14569/IJACSA.2021.0120945>
2. Abdullahi S, Ahmed Zayyad M, Yusuf N et al. (2021) Software requirements negotiation: a review on challenges. *Int J Innov Comput* 11(1):1–6. <https://doi.org/10.11113/ijic.v11n1.264>
3. Abdullai L, Shamshiri H, Mahmud H et al. (2022) A systematic mapping study of empirical research methods in software ecosystems. In: Carroll N, Nguyen-Duc A, Wang X et al. (eds) *Software business*. Springer International Publishing, Cham, pp 182–195. <https://doi.org/10.1007/978-3-031-20706-813>
4. Ågerfalk PJ, Fitzgerald B, Holmström Olsson H et al. (2008) Benefits of global software development: the known and unknown. In: Wang Q, Pfahl D, Raffo DM (eds) *Making globally distributed software development a success story*. Springer, Berlin, Heidelberg, pp 1–9
5. Ailane TM, Abboush M, Knieke C, et al. (2021) Toward formalizing the emergent behavior in software engineering. In: 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 32–39. <https://doi.org/10.1109/SESoS-WDES52566.2021.00010>
6. Ali N, Lai R (2017) A method of software requirements specification and validation for global software development. *Requir Eng* 22:191–214. <https://doi.org/10.1007/s00766-015-0240-4>
7. Ali N, Lai R (2021) Global software development: a review of its practices. *Malays J Comput Sci* 34(1):82–129. <https://doi.org/10.22452/mjcs.vol34no1.5>
8. Ali Khan J, Liu L, Wen L et al. (2020) Conceptualising, extracting and analysing requirements arguments in users' forums: The CrowdRE-Arg framework. *J Softw: Evol Process* 32(12):e2309. <https://doi.org/10.1002/smr.2309>
9. Alsanoosy T, Spichkova M, Harland J (2020) Cultural influence on requirements engineering activities: a systematic literature

- review and analysis. *Requir Eng* 25(3):339–362. <https://doi.org/10.1007/s00766-019-00326-9>
10. Ampatzoglou A, Bibi S, Avgeriou P et al. (2019) Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Inf Softw Technol* 106:201–230. <https://doi.org/10.1016/j.infsof.2018.10.006>
11. Anwer S, Wen L, Wang Z (2019) A systematic approach for identifying requirement change management challenges: preliminary results. In: 23rd International Conference on Evaluation and Assessment in Software Engineering. Association for Computing Machinery, New York, NY, USA, pp 230–235. <https://doi.org/10.1145/3319008.3319031>
12. Axelsson J, Skoglund M (2016) Quality assurance in software ecosystems: a systematic literature mapping and research agenda. *J Syst Softw* 114:69–81. <https://doi.org/10.1016/j.jss.2015.12.020>
13. Barbosa O, Santos RP, Alves C et al. (2013) A systematic mapping study on software ecosystems from a three-dimensional perspective. In: Jansen S, Brinkkemper S, Cusumano M (eds) *Software ecosystems: analyzing and managing business networks in the software industry*. Edward Elgar Publishing, Cheltenham, pp 59–81. <https://doi.org/10.4337/9781781955628.00011>
14. Basili VR, Rombach HD (1988) The tame project: towards improvement-oriented software environments. *IEEE Trans Softw Eng* 14(6):758–773. <https://doi.org/10.1109/32.6156>
15. Berenbach B, Paulish DJ, Kazmeier J et al. (2009) *Software & systems requirements engineering: in practice*. McGraw-Hill Education, New York
16. Bhuta J, Boehm B, Meyers S (2006) Process elements: components of software process architectures. In: Li M, Boehm B, Osterweil LJ (eds) *Unifying the software process spectrum*. Springer, Berlin, Heidelberg, pp 332–346. [https://doi.org/10.1007/11608035\\_28](https://doi.org/10.1007/11608035_28)
17. Bosch J (2009) From software product lines to software ecosystems. In: 13th International Software Product Line Conference, SPLC '09, vol 9. Carnegie Mellon University, USA, pp 111–119
18. Charmaz K (2006) *Constructing grounded theory: a practical guide through qualitative analysis*. Sage Publications, Thousand Oaks
19. Che M, Perry DE (2014) Architectural design decisions in open software development: a transition to software ecosystems. In: 2014 23rd Australian Software Engineering Conference, pp 58–61. <https://doi.org/10.1109/ASWEC.2014.37>
20. Cleland-Huang J (2018) Disruptive change in requirements engineering research. In: IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, pp 1–2. <https://doi.org/10.1109/RE.2018.00-58>
21. CMMI (2010) *CMMI for development v1.3*. Tech. rep., Software Engineering Institute, Carnegie Mellon University. <https://doi.org/10.1184/R1/6572342.v1>
22. Cruzes DS, Dyba T (2011) Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement, IEEE, Banff, AB, Canada, pp 275–284. <https://doi.org/10.1109/ESEM.2011.36>
23. Cysneiros G, Zisman A (2008) Traceability and completeness checking for agent-oriented systems. In: 2008 ACM Symposium on Applied Computing. Association for Computing Machinery, New York, NY, USA, pp 71–77. <https://doi.org/10.1145/1363686.1363706>
24. Damian D, Linåker J, Johnson D et al. (2021) Challenges and strategies for managing requirements selection in software ecosystems. *IEEE Softw* 38(6):76–87. <https://doi.org/10.1109/MS.2021.3105044>
25. Dermeval D, Vilela J, Bittencourt II et al. (2016) Applications of ontologies in requirements engineering: a systematic review of



- the literature. *Requir Eng* 21:405–437. <https://doi.org/10.1007/s00766-015-0222-6>
26. Feiler P, Humphrey W (1993) Software process development and enactment: concepts and definitions. In: Second International Conference on the Software Process-Continuous Software Process Improvement, Berlin, Germany, pp 28–40. <https://doi.org/10.1109/SPCON.1993.236824>
  27. Fernández DM, Wagner S, Kalinowski M et al. (2017) Naming the pain in requirements engineering. *Empir Softw Eng* 22(5):1–41. <https://doi.org/10.1007/s10664-016-9451-7>
  28. Fernandez S, Svensson RB (2017) A survey of practitioners use of open innovation. In: 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Vienna, Austria, pp 305–312. <https://doi.org/10.1109/SEAA.2017.52>
  29. Figalíst I, Elsner C, Bosch J et al. (2019) Scaling agile beyond organizational boundaries: coordination challenges in software ecosystems. In: Kruchten P, Fraser S, Coallier F (eds) *Agile processes in software engineering and extreme programming*. Springer International Publishing, Cham, pp 189–206. [https://doi.org/10.1007/978-3-030-19034-7\\_12](https://doi.org/10.1007/978-3-030-19034-7_12)
  30. Fontão AL, Santos RP, Dias-Neto AC (2015) Mobile software ecosystem (mseco): a systematic mapping study. In: 39th Annual Computer Software and Applications Conference, IEEE, Taichung, Taiwan, pp 653–658. <https://doi.org/10.1109/COMPSAC.2015.121>
  31. Franco-Bedoya O, Ameller D, Costal D et al. (2017) Open source software ecosystems: a systematic mapping. *Inf Softw Technol* 91:160–185. <https://doi.org/10.1016/j.infsof.2017.07.007>
  32. Fricker S (2009) Specification and analysis of requirements negotiation strategy in software ecosystems. In: 2009 International Workshop on Software Ecosystems (IWSECO'09), Falls Church, VA, USA, pp 19–33. <https://doi.org/10.5167/uzh-28289>
  33. Fricker S, Gorschek T, Myllyperkiö P (2007) Handshaking between software projects and stakeholders using implementation proposals. In: Sawyer P, Paech B, Heymans P (eds) *Requirements engineering: foundation for software quality*. Springer, Berlin Heidelberg, pp 144–159. [https://doi.org/10.1007/978-3-540-73031-6\\_11](https://doi.org/10.1007/978-3-540-73031-6_11)
  34. García YM, Montes Á, Lira J, et al. (2019) Requirements management techniques and tools in small and medium enterprises (smes): a systematic review. In: IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), IEEE, Ixtapa, Mexico, pp 1–7. <https://doi.org/10.1109/ROPEC48299.2019.9057050>
  35. Glinz M (2020) Standard glossary for the certified professional for requirements engineering (CPRE) studies and exam v2. Tech. rep., International Requirements Engineering Board eV
  36. Groen EC, Seyff N, Ali R et al. (2017) The crowd in requirements engineering: the landscape and challenges. *IEEE Softw* 34(2):44–52. <https://doi.org/10.1109/MS.2017.33>
  37. Gómez M, Adams B, Maalej W et al. (2017) App store 2.0: from crowdsourced information to actionable feedback in mobile ecosystems. *IEEE Softw* 34(2):81–89. <https://doi.org/10.1109/MS.2017.46>
  38. Hanssen GK, Dybå T (2012) Theoretical foundations of software ecosystems. In: Fourth International Workshop on Software Ecosystems (IWSECO). CEUR-WS.org, pp 6–17
  39. Harland PE, Wüst S, Dedehayir O (2014) Innovation processes in business ecosystems: creating a common understanding by requirements. In: Portland International Center for Management of Engineering and Technology (PICMET '14); Infrastructure and Service Integration. Kanazawa, Japan, pp 723–729
  40. Heistracher T, Kurz T, Marcon G, et al. (2006) Collaborative software engineering with a digital ecosystem. In: IEEE International Conference on Global Software Engineering (ICGSE'06), IEEE Computer Society, Florianópolis, Brazil, pp 119–126. <https://doi.org/10.1109/ICGSE.2006.261224>
  41. ISO/IEC/IEEE (2011) ISO/IEC/IEEE international standard—systems and software engineering—life cycle processes—requirements engineering. ISO/IEC/IEEE 29148:2011(E) pp 1–94. <https://doi.org/10.1109/IEEESTD.2011.6146379>
  42. Jung A, Carbonell J, Marchezan L et al. (2020) Systematic mapping study on domain-specific language development tools. *Empir Softw Eng* 25:4205–4249. <https://doi.org/10.1007/s10664-020-09872-1>
  43. Iyer DG, Lytinen K (2019) Requirements engineering (re) effectiveness in open source software: the role of social network configurations and requirements properties. In: 27th European Conference on Information Systems (ECIS). Association for Computing Machinery, New York, NY, USA, ESEM '12, pp 8–14
  44. Jansen S (2020) A focus area maturity model for software ecosystem governance. *Inf Softw Technol* 118(106):219. <https://doi.org/10.1016/j.infsof.2019.106219>
  45. Jansen S, Peeters S, Brinkkemper S (2013) Software ecosystems: from software product management to software platform management. In: *Software Business. From Start-ups to SaaS Conglomerate: Life Cycles of Software Products Workshop (IW-LCSP 2013)*. Potsdam, Germany, pp 5–18
  46. Jayatilake S, Lai R (2018) A systematic review of requirements change management. *Inf Softw Technol* 93:163–185. <https://doi.org/10.1016/j.infsof.2017.09.004>
  47. Johnson D, Tizard J, Damian D, et al. (2020) Open crowdre challenges in software ecosystems. In: 4th International Workshop on Crowd-Based Requirements Engineering (CrowdRE), Zurich, Switzerland, pp 1–4. <https://doi.org/10.1109/CrowdRE51214.2020.00007>
  48. Kamalrudin M, Sidek S (2015) A review on software requirements validation and consistency management. *Int J Softw Eng Appl* 9(10):39–58. <https://doi.org/10.14257/ijseia.2015.9.10.05>
  49. Kazman R, Chen HM (2010) The metropolis model and its implications for the engineering of software ecosystems. In: *FSE/SDP Workshop on Future of Software Engineering Research*. Association for Computing Machinery, New York, NY, USA, p 187–190. <https://doi.org/10.1145/1882362.1882402>
  50. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Evidence-Based Software Engineering (EBSE) Project
  51. Kitchenham BA, Budgen D, Brereton P (2015) Evidence-based software engineering and systematic reviews, vol 4. CRC Press, Boca Raton
  52. Knauss A, Borici A, Knauss E, et al. (2012) Towards understanding requirements engineering in it ecosystems. In: Second IEEE International Workshop on Empirical Requirements Engineering (EmpiRE), Chicago, IL, USA, pp 33–36. <https://doi.org/10.1109/EmpiRE.2012.6347679>
  53. Knauss E, Yussuf A, Blincoe K et al. (2018) Continuous clarification and emergent requirements flows in open-commercial software ecosystems. *Requir Eng* 23(1):97–117. <https://doi.org/10.1007/s00766-016-0259-1>
  54. Lewellen S (2020) Identifying key stakeholders as part of requirements elicitation in software ecosystems. In: 24th ACM International Systems and Software Product Line Conference—Volume B. Association for Computing Machinery, New York, NY, USA, pp 88–95. <https://doi.org/10.1145/3382026.3431249>
  55. Lewellen S (2021) A comprehensive approach to identifying key stakeholders in complicated software ecosystems. In: IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, pp 492–497. <https://doi.org/10.1109/RE51729.2021.00074>

56. Lim S, Henriksson A, Zdravkovic J (2021) Data-driven requirements elicitation: a systematic literature review. *SN Comput Sci* 2:1–35. <https://doi.org/10.1007/s42979-020-00416-4>
57. Lima T, Werner C, Santos R (2019) Identifying architecture attributes in the context of software ecosystems based on a mapping study. In: Hyrynsalmi S, Suoranta M, Nguyen-Duc A et al. (eds) *Software business*. Springer International Publishing, Cham, pp 55–70. [https://doi.org/10.1007/978-3-030-33742-1\\_6](https://doi.org/10.1007/978-3-030-33742-1_6)
58. Linåker J, Rempel P, Regnell B et al. (2016) How firms adapt and interact in open source ecosystems: analyzing stakeholder influence and collaboration patterns. In: Daneva M, Pastor O (eds) *Requirements engineering: foundation for software quality*. Springer International Publishing, Cham, pp 63–81. [https://doi.org/10.1007/978-3-319-30282-9\\_5](https://doi.org/10.1007/978-3-319-30282-9_5)
59. Linåker J, Regnell B, Damian D (2020) A method for analyzing stakeholders' influence on an open source software ecosystem's requirements engineering process. *Requir Eng* 25(1):115–130. <https://doi.org/10.1007/s00766-019-00310-3>
60. Linåker J, Runeson P (2020) Public sector platforms going open: creating and growing an ecosystem with open collaborative development. In: 16th International Symposium on Open Collaboration. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3412569.3412572>
61. Linåker J, Regnell B, Munir H (2015) Requirements engineering in open innovation: a research agenda. In: 2015 International Conference on Software and System Process. Association for Computing Machinery, New York, NY, USA, pp 208–212. <https://doi.org/10.1145/2785592.2795370>
62. Linåker J, Wnuk K (2016) Requirements analysis and management for benefiting openness. In: IEEE 24th International Requirements Engineering Conference Workshops (REW), IEEE, Beijing, China, pp 344–349. <https://doi.org/10.1109/REW.2016.062>
63. Linåker J, Munir H, Wnuk K et al. (2018) Motivating the contributions: an open innovation perspective on what to share as open source software. *J Syst Softw* 135:17–36. <https://doi.org/10.1016/j.jss.2017.09.032>
64. Luz PBV, Fernandes J, Valença G et al. (2020) Exploring sustainability in real cases of emerging small-to-medium enterprises ecosystems. In: Santos R, Maciel C, Viterbo J (eds) *Software ecosystems, sustainability and human values in the social web*. Springer International Publishing, Berlin, pp 42–59. [https://doi.org/10.1007/978-3-030-46130-0\\_3](https://doi.org/10.1007/978-3-030-46130-0_3)
65. Manikas K (2016) Revisiting software ecosystems research: a longitudinal literature study. *J Syst Softw* 117:84–103. <https://doi.org/10.1016/j.jss.2016.02.003>
66. Messerschmitt DG, Szyperski C (2003) *Software ecosystem: understanding an indispensable technology and industry*. MIT Press, Cambridge
67. Mourão E, Pimentel JF, Murta L et al. (2020) On the performance of hybrid search strategies for systematic literature reviews in software engineering. *Inf Softw Technol* 123(106):294. <https://doi.org/10.1016/j.infsof.2020.106294>
68. Napoleão B, Felizardo K, Souza E, et al. (2021) Establishing a search string to detect secondary studies in software engineering. In: 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy, pp 9–16. <https://doi.org/10.1109/SEAA53835.2021.00010>
69. Ochodek M, Koczyńska S (2018) Perceived importance of agile requirements engineering practices—A survey. *J Syst Softw* 143:29–43. <https://doi.org/10.1016/j.jss.2018.05.012>
70. Oriol M, Müller C, Marco J et al. (2023) Comprehensive assessment of open source software ecosystem health. *Internet Things* 22(100):808. <https://doi.org/10.1016/j.iot.2023.100808>
71. Paternoster N, Giardino C, Unterkalmsteiner M et al. (2014) Software development in startup companies: a systematic mapping study. *Inf Softw Technol* 56(10):1200–1218. <https://doi.org/10.1016/j.infsof.2014.04.014>
72. Petersen K, Feldt R, Mujtaba S et al. (2008) Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12. BCS Learning & Development Ltd., Swindon, GBR, pp 68–77
73. Qureshi S, Khan SUR, Iqbal J et al. (2021) A study on mitigating the communication and coordination challenges during requirements change management in global software development. *IEEE Access* 9:88217–88242. <https://doi.org/10.1109/ACCESS.2021.3090098>
74. Roth S, Leydesdorff L, Kaivo-Oja J et al. (2020) Open cooperation: when multiple players and rivals team up. *J Bus Strategy* 41(6):31–38. <https://doi.org/10.1108/JBS-11-2018-0192>
75. Santos J, Martins L, Santiago Júnior V et al. (2020) Exploring the challenges and benefits for scaling agile project management to large projects: a review. *Requir Eng* 25:317–337. <https://doi.org/10.1007/s00766-019-00325-w>
76. Santos P, Carvalho M (2022) Exploring the challenges and benefits for scaling agile project management to large projects: a review. *Requir Eng* 27:117–134. <https://doi.org/10.1007/s00766-021-00363-3>
77. Santos R, Esteves M, Freitas G et al. (2014) Using social networks to support software ecosystems comprehension and evolution. *Soc Netw* 3(2):49–69. <https://doi.org/10.4236/sn.2014.32014>
78. Santos RP, Werner CML (2010) Revisiting the concept of components in software engineering from a software ecosystem perspective. In: Fourth European Conference on Software Architecture: Companion Volume. Association for Computing Machinery, New York, NY, USA, pp 135–142. <https://doi.org/10.1145/1842752.1842782>
79. Santos RP, Werner CML (2011) A proposal for software ecosystems engineering. In: Third International Workshop on Software Ecosystems (IWSECO-2011), pp 40–51
80. Santos RP, Werner CML (2013) On the impact of software ecosystems in requirements communication and management. In: Castro J, Alencar F, Lucena M, et al. (eds) *Requirements Engineering@Brazil 2013*, CEUR Workshop Proceedings, vol 1005. CEUR-WS.org, Rio de Janeiro, Brazil
81. Santos RP, Werner C, Finkelstein A (2018) Ecosystems effects on software-consuming organizations: an experience report on two observational studies. In: 12th European Conference on Software Architecture: Companion Proceedings. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3241403.3241428>
82. Santos RPd, Werner CML (2012) Reuseecos: an approach to support global software development through software ecosystems. In: IEEE Seventh International Conference on Global Software Engineering Workshops, IEEE, Porto Alegre, Brazil, pp 60–65. <https://doi.org/10.1109/ICGSEW.2012.16>
83. Scacchi W (2009) Understanding requirements for open source software. In: Lytinen K, Loucopoulos P, Mylopoulos J et al. (eds) *Design requirements engineering: a ten-year perspective*. Springer, Berlin, Heidelberg, pp 467–494. [https://doi.org/10.1007/978-3-540-92966-6\\_27](https://doi.org/10.1007/978-3-540-92966-6_27)
84. Schön EM, Thomaschewski J, Escalona MJ (2017) Agile requirements engineering: a systematic literature review. *Comput Stand Interfaces* 49:79–91. <https://doi.org/10.1016/j.csi.2016.08.011>
85. Shahzad B, Javed I, Shaikh A et al. (2021) Reliable requirements engineering practices for COVID-19 using blockchain. *Sustainability* 13(12):6748. <https://doi.org/10.3390/su13126748>
86. Shaw M (2003) Writing good software engineering research papers. In: 25th International Conference on Software Engineering, Portland, OR, USA, pp 726–736. <https://doi.org/10.1109/ICSE.2003.1201262>

87. Soltani M, Knauss E (2015) Cross-organizational challenges of requirements engineering in the autostar ecosystem: an exploratory case study. In: IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE), Ottawa, ON, Canada, pp 41–48. <https://doi.org/10.1109/EmpiRE.2015.7431306>
88. Song W (2017) Requirement management for product-service systems: status review and future trends. *Comput Ind* 85:11–22. <https://doi.org/10.1016/j.compind.2016.11.005>
89. Srinivasan J, Dobrin R, Lundqvist K (2009) 'State of the art' in using agile methods for embedded systems development. In: 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA, USA, pp 522–527. <https://doi.org/10.1109/COMPSAC.2009.186>
90. Tizard J, Rietz T, Liu X et al. (2022) Voice of the users: an extended study of software feedback engagement. *Requir Eng* 27(3):293–315. <https://doi.org/10.1007/s00766-021-00357-1>
91. Tuunanen T, Rossi M (2004) Engineering a method for wide audience requirements elicitation and integrating it to software development. In: 37th Annual Hawaii International Conference on System Sciences, IEEE, Big Island, HI, USA, pp 10–pp. <https://doi.org/10.1109/HICSS.2004.1265420>
92. Valença G, Alves C, Tedesco P, et al. (2013) Analysing requirements negotiation in software ecosystems with multi-agent systems techniques. VII WDDSS, Brasília pp 44–51
93. Valença G, Alves C (2016) Understanding how power influences business and requirements decisions in software ecosystems. In: 31st Annual ACM Symposium on Applied Computing. Association for Computing Machinery, New York, NY, USA, pp 1258–1263. <https://doi.org/10.1145/2851613.2851756>
94. Valença G (2013) Requirements negotiation model: a social oriented approach for software ecosystems evolution. In: IEEE 21st International Requirements Engineering Conference (RE), Rio de Janeiro, Brazil, pp 393–396. <https://doi.org/10.1109/RE.2013.6636763>
95. Valença G, Alves C, Heimann V, et al. (2014) Competition and collaboration in requirements engineering: a case study of an emerging software ecosystem. In: IEEE 22nd International Requirements Engineering Conference (RE), Karlskrona, Sweden, pp 384–393. <https://doi.org/10.1109/RE.2014.6912289>
96. Vegendla A, Duc AN, Gao S et al. (2018) A systematic mapping study on requirements engineering in software ecosystems. *J Inf Technol Res (JITR)* 11(1):49–69. <https://doi.org/10.4018/JITR.2018010104>
97. Villela K, Hess A, Koch M, et al. (2018) Towards ubiquitous RE: a perspective on requirements engineering in the era of digital transformation. In: IEEE 26th International Requirements Engineering Conference (RE), Banff, AB, Canada, pp 205–216. <https://doi.org/10.1109/RE.2018.00029>
98. Villela K, Groen EC, Doerr J (2019) Ubiquitous requirements engineering: a paradigm shift that affects everyone. *IEEE Softw* 36(2):8–12. <https://doi.org/10.1109/MS.2018.2883876>
99. Wiegers K, Beatty J (2013) *Software requirements*. Pearson Education, London
100. Wohlin C, Höst M, Henningsson K (2003) Empirical research methods in software engineering. In: Conradi R, Wang AI (eds) *Empirical methods and studies in software engineering: experiences from ESERNET*. Springer, Berlin, Heidelberg, pp 7–23. [https://doi.org/10.1007/978-3-540-45143-3\\_2](https://doi.org/10.1007/978-3-540-45143-3_2)
101. Wohlin C, Kalinowski M, Romero Felizardo K et al. (2022) Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. *Inf Softw Technol* 147(106):908. <https://doi.org/10.1016/j.infsof.2022.106908>
102. Wouters J, Menkveld A, Brinkkemper S et al. (2022) Crowd-based requirements elicitation via pull feedback: method and case studies. *Requir Eng*. <https://doi.org/10.1007/s00766-022-00384-6>
103. Yaseen M, Baseer S, Sherin S (2015) Critical challenges for requirement implementation in context of global software development: a systematic literature review. In: 2015 International Conference on Open Source Systems & Technologies (ICOSST), Lahore, Pakistan, pp 120–125. <https://doi.org/10.1109/ICOSST.2015.7396413>
104. Yaseen M, Ali Z, Humayoun M (2019) Requirements management model (rmm): A proposed model for successful delivery of software projects. *International Journal of Computer Applications* 178(17):32–36. <https://doi.org/10.5120/ijca2019918984>
105. Yin H, Pfahl D (2017) Open innovation in software requirements engineering: A mapping study. In: 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, pp 5–10. <https://doi.org/10.1109/ICSESS.2017.8342852>
106. Zaggl MA, Schweisfurth TG, Herstatt C (2020) The dynamics of openness and the role of user communities: A case study in the ecosystem of open source gaming handhelds. *IEEE Trans Eng Manage* 67(3):712–723. <https://doi.org/10.1109/TEM.2019.2897900>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.