

Article

# How to Create a Software Ecosystem? A Partnership Meta-Model and Strategic Patterns

Ítalo Belo \* and Carina Alves \*

Informatic Center (Cin), Federal University of Pernambuco, Recife 50740-560, Brazil

\* Correspondence: icsb@cin.ufpe.br (Í.B.); cfa@cin.ufpe.br (C.A.)

**Abstract:** Large keystone organizations use partnership models to manage their software ecosystem partners. Although several partnership models have been developed by platform owners, smaller companies willing to create a new ecosystem may experience difficulties to define the appropriate features of partnership models when switching from an independent software product to an ecosystem. This study proposes a partnership meta-model and four strategic patterns to operationalize it. We adopted the Design Science Research (DSR) method. The partnership meta-model was built in the first cycle of DSR, using a Systematic Mapping Study, and validated through case studies of SAP, Eclipse, and Microsoft Azure ecosystems. In the second cycle of DSR, the strategic patterns were defined through a Multivocal Literature Review and validated by using interviews with professionals. The meta-model presents the key characteristics to define partnership models for emerging software ecosystems. The strategic patterns aim to operationalize the meta-model and, consequently, assist the keystone in defining the features that the partnership model will have and select suitable strategies. The meta-model and the strategic patterns help managers creating and evolving software ecosystems from a software product considering the impact of that transition on the partnership model.

**Keywords:** software ecosystems; partnership models; meta-model; strategic patterns



**Citation:** Belo, Í.; Alves, C. How to Create a Software Ecosystem? A Partnership Meta-Model and Strategic Patterns. *Information* **2021**, *12*, 240. <https://doi.org/10.3390/info12060240>

Academic Editor: Enrico Denti

Received: 11 May 2021

Accepted: 29 May 2021

Published: 3 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The platform business model has made companies such as Google, Amazon, Microsoft, Uber, SAP, and Airbnb global leaders. That model gave these companies rapid growth and made them powerfully disruptive. In this scenario of platforms, the software ecosystem has become essential to support organizational collaboration and competition. It enabled the creation of successful platforms such as Google G Suite, Microsoft Azure, SAP, and Facebook [1,2].

Software ecosystems can be defined as “the software and actor interaction in relation to a common technological infrastructure, that results in a set of contributions and influences directly or indirectly the ecosystem” [3]. Such actors can be classified as clients (consumers), external developers (complementors), and platform providers (keystone). The platform provider is not always present in the ecosystem as a single organization and can be represented as an open-source software community [4–6]. The generation and sharing of value among actors of an ecosystem are affected by partnerships among keystone and complementors. The alliance formation allows partners to acquire news assets, create innovative solutions, and share risks and costs [7]. These alliances are orchestrated through partnership models.

Partnership models organize the relationships between the actors of the ecosystem, structuring rights, responsibilities, and rules, such as partner access to the source code of the ecosystem platform, restrictions of the affiliation contract, and disciplinary sanctions for treat negative attitudes [8]. Without an adequate understanding of how to create partnership models, organizations may fail in establishing mechanisms to attract and retain partners, as well as survive in the competitive market.

More specifically, designing a partnership model for software ecosystems is not a trivial task. For example, defining the entry requirements for new partners in the ecosystem, establishing actors' roles, assessing partners' quality, and resolving conflicts between actors are usual challenges involved in the definition of appropriate partnership models [9,10]. Another challenge faced by actors when structuring a partnership model is the management of factors that affect that model when creating an ecosystem from a software product, such as selecting prosperous partners, enabling suitable communication within the network, developing trust among stakeholders, ensuring the quality of solutions available on the platform, defining subsidy strategies to attract participants, and identifying strategies to prevent ecosystem participants from being "taken over" by competitors [1,2,10,11].

Although there is an increasing tendency for companies to migrate from an independent software product approach to a software ecosystem, the difficulties in defining suitable partnership models challenge many organizations, and they may fail to create and develop their ecosystems [1,12,13]. Currently, in the literature, important partnership models have been analyzed. However, they do not make it possible to have a holistic and integrated view of the keys characteristic of the partnership model (e.g., conflict management and partners quality) and its relationships, as well as mechanisms (strategies) to address the challenges of defining the partnership model when moving from independent software to an ecosystem [1,2,10,12]. To address this research gap, we propose a meta-model that indicates the features and management areas that companies need to address to define specific partnership models. We also propose the following strategic patterns: Platform Quality, Support for the Partner, Participant's Attraction and Maintenance, and Participant's Profile. These patterns address key challenges (e.g., entry requirements definition for new partners, Platform Quality, and establishment of subsidy policies) that influence partnership model definition, enabling the creation and maintenance of alliances when creating an ecosystem from a software product.

The proposed partnership meta-model and strategic patterns enable us to understand the characteristics, relations, and strategies needed to define partnership models when creating a new ecosystem from a software product. That understanding supports the effective governance of alliances between keystone and complementors. It enables the development of successful collaborations, thus nurturing the creation and sharing of value in the new ecosystem.

## 2. Literature Review

This section discusses key concepts of software ecosystems: definitions, types, and lifecycle. In addition, it presents an analysis of partnerships in software ecosystems.

### 2.1. Software Ecosystems

In Plakidas [14], a software ecosystem is defined as "a common software base upon which a network or community of users with shared interests or values has built up a collection of derivative software products (a software market) in order to satisfy certain needs and accrue benefits, whether monetary or otherwise". Meanwhile, Manikas [6] considers a software ecosystem to be "the interaction of a set of actors on a common technological platform, which results in several software solutions or services".

There are several other definitions of software ecosystems that address the technical, social, or business perspectives [3,15]. The technical perspective focuses on the definition and maintenance of the platform architecture. The social view addresses aspects related to the interaction between the actors in the software ecosystem. The business perspective deals with the position of the ecosystem in the competitive market, seeking its competitive advantage and prosperity [3,16].

Regardless of the technical, business, or social viewpoint, the solutions generated in the software ecosystems are created through partnerships between the keystone (ecosystem orchestrator) and the complementors (responsible for developing software solutions or

services to complement the ecosystem platform). The complementors are also known as “niche player”, “external actor”, and “reseller with added value” [15,17–20].

It is essential to highlight that a software ecosystem can be “open source” or “proprietary” [9,21–23]. Open-source ecosystems are generally controlled by a foundation (or association) representing the community’s interests, which aims to increase the member’s contribution to projects. In open source ecosystems, the entry barriers tend to be lower than in the proprietary, and their members also aim to obtain non-financial benefits, such as fame, knowledge, and ideology [6]. Proprietary ecosystems belong to private companies and are governed by considering the performance of each actor from a commercial perspective. Compared to open source, proprietary ecosystems are more centralized [9]. It is important to note that there are hybrid ecosystems, such as Eclipse, which is mainly open source but has private companies as partners.

## 2.2. Software Ecosystem Lifecycle

Understanding the current stage of the software ecosystem lifecycle is essential for the keystone to make strategic decisions when governing the ecosystem. That lifecycle indicates the evolution of the ecosystem, considering the partner network focus, how the network is organized, and the market situation [24,25].

According to Moore [24], a software ecosystem lifecycle has the following steps: birth, expansion, leadership, and self-renewal. At the birth phase, the focus is to define what customers want and how to make the envisioned product or service available to the market [24]. At the expansion phase, rival ecosystems may choose to compete in the same market. Therefore, the business concept of the ecosystem must obtain a large customer base to expand more than its competitors. In the leadership phase, the keystone needs to maintain the authority exercised in the ecosystem and increase the ability to attract investments from key customers and suppliers [24,26]. Partners must acknowledge that only large innovations can fuel the large-scale growth of the ecosystem. Finally, at the self-renewal stage, the focus is to avoid obsolescence. It is important to be aware that the ecosystem will prosper in the long-term, with adequate renewal capacity, if it is capable to lead continuous innovation for a long time [24].

In a different way, Rong [25] proposed a lifecycle with the following phases: emerging, diversifying, converging, consolidating, and renewing. In the first phase, a new solution is launched to the market; the keystone and partners provide a simple supply chain. In the second phase, called diversifying, the focus is to promote diverse solutions in the ecosystem, and the network of partners organizes itself as a cohesive set of allies. The third phase of the lifecycle is characterized by the partner’s focus on specialized niche markets. In the fourth phase, the emphasis is on the mass production of a dominant design solution through partners that maintain close alliances. Finally, in the last step (renewal), the partners seek to renew the “original market”, and new market niches have emerged. In that context, there may be a substitution of the original market for an emerging one, which will take organizations to the beginning of the life cycle to supply the new market’s demand. In this paper, we focus on the initial stage of a software ecosystem that is called birth or emerging phase.

## 2.3. Partnerships in Software Ecosystems

Historically, there is a dependency between organizations that own platforms and companies that develop complementary solutions, since the success of both depends on the joint innovation of their products or services [27]. In the case of software vendors such as Eclipse and SAP, that dependency is characterized by the formation of partnerships between platform owners (i.e., keystones) and complementors [9].

The keystone is responsible for the orchestration of the entire software ecosystem, which includes providing the platform for that ecosystem, while complementors provide complementary products and services [4–7]. Together, orchestrators of software ecosystems

(keystones) and complementors operate in the market forming a partnerships network between them. That network is formed to obtain mutual benefits [4–7].

Through alliances, the keystone seeks benefits such as strengthening the market presence and obtaining innovative solutions [28], while complementors try to expand their business and better meet customer demands [20]. In the context of alliances in ecosystems, the keystone needs to determine which products and services should be made in-house and the ones that should be created by complementors. However, although the keystone decides to leave the development of some products or services (complements) to third parties, it must exert influence on the complements design and production, using resources such as API, marketing support, and professional assistance for the partner to develop and sell the complements [1,27].

The keystone must maintain control over the ecosystem architecture evolution and leave the architecture with adequate modularity to reduce innovation costs through complements. Moreover, the keystone must protect its intellectual property and promote the partners' copyright protection as well [1,27]. One way to do this is by using documents such as policies and conduct codes, as these documents guide alliances in the ecosystem. The alliances formed between keystone and complementors are defined in contracts and have different characteristics, depending on the type of ecosystem (open source or proprietary) [9,22,23]. The main difference between these ecosystems is the orchestrator type. A private company orchestrates the proprietary ecosystem. In contrast, the open source ecosystem is controlled by a foundation (or association) that represents the community's interests [9].

In that scope of alliances between ecosystem owners and complementors, partnership models are essential governance mechanisms to enable ecosystems' sustainable growth and, consequently, managing the software ecosystem partners cluster (attracting, grouping, and maintaining the ecosystem's affiliates) [29]. The partnership models define the roles, responsibilities, and benefits of the keystone and its partners. These models are influenced by factors such as monetization policy, partners' selection, ecosystem platform solutions quality, conflict management, and platform evolution [1,2,9,10].

#### 2.4. Related Works

In the context of partnership models as a governance mechanism to manage the keystone's cluster partners, studies were carried out with different focuses. The characterization of the software ecosystem partnership model was addressed, presenting the properties of the partnerships between the keystone and its partners signed through a contract [S1]. In addition, relationships between business strategies and revenue streams from application developers have been described [S18].

Some studies have investigated the creation of partnerships. The influence of the number of applications that an actor develops in his insertion in the ecosystem (ratio between the number of relationships that an actor has with others and the total number of relationships that is theoretically possible) was verified [S10]. Moreover, the following topics were addressed: the relationship between intellectual property rights or downstream capabilities (trademarks and consulting services) and the complementor's decision to join the platform [S13], as well as factors that influence the entry of ISVs (independent software resellers) in complementary markets [S14]. Reference [S8] investigated the impact of keystone's capabilities on complementors' motivation to create new partnerships. In addition, factors that influence the time for ISVs to form partnerships [S7] and bridges and barriers that affect the players' participation in software ecosystems [S15] were analyzed.

Within that scope of partnerships, the following aspects were studied, too: influence of autonomy and control mechanisms on the viscosity of the platform (complementors' intention of remain on a mobile application development platform) [S19], factors that influence developers satisfaction [S22], and managing the tension between control and trust in hub-and-spokes networks [S17]. In addition, the interactions between barriers and facilitators involved in the relationships between partners [S23] were studied besides the

relationship between the organizational coupling sought by the complementors, keystone's opportunistic behavior, and partners' capacities [S20].

It is important to note that value co-creation was studied too. The model proposed by Reference [S5] addressed the complementors' perception of elements present in the value co-creation process; meanwhile, Reference [S16] approached the relationship among co-creation value modes, resources provided by alliance participants, and the types of factors (facilitators or inhibitors) associated with these co-creation value modes.

It was also analyzed the number of inter-firm relationships initiated by a complementor and the impact of the partners joining the partnership model on their respective productivities [S4]. Furthermore, the keystone's partner network influence on the value of its firm was investigated [S21], and the impact of the partnership on the business performance of small independent software vendors [S2]. The effect of control mechanisms on ecosystem leaders' objectives was assessed by Reference [S3].

Finally, in Reference [S12], the authors presented strategies adopted by software vendors (complementors) and their respective trade-offs. Reference [S11] described a framework for managing partnerships in the enterprise software industry (which has areas and levels for partner management), and Reference [S9] proposed a framework for assessing the ecosystem governance state. That framework has, among others, concepts that allow the current state assessment of the partnership governance aspects. In Reference [S6], the authors presented the "Software Platform Management Competency" based on the changes executed on the "Competency Model in Software Product Management". They identified areas and competencies for managing software ecosystems, including partnerships.

It can be noted that the studies presented addressed different important perspectives of the partnership models. However, they do not present in an integrated and global way key characteristics of these models, how these characteristics are related, and how to operationalize them when changing from an independent software product to an ecosystem.

### 3. Research Method

The research method used in this study was Design Science Research (DSR). The method proposes the creation and evaluation of artifacts to solve an organizational problem or propose improvements in the treatment of that problem [30–32]. The following research questions guided the process of creating, refining, and evaluating the artifacts generated in this study, using the DSR method:

**RQ1.** What are the key elements of partnership models in software ecosystems?

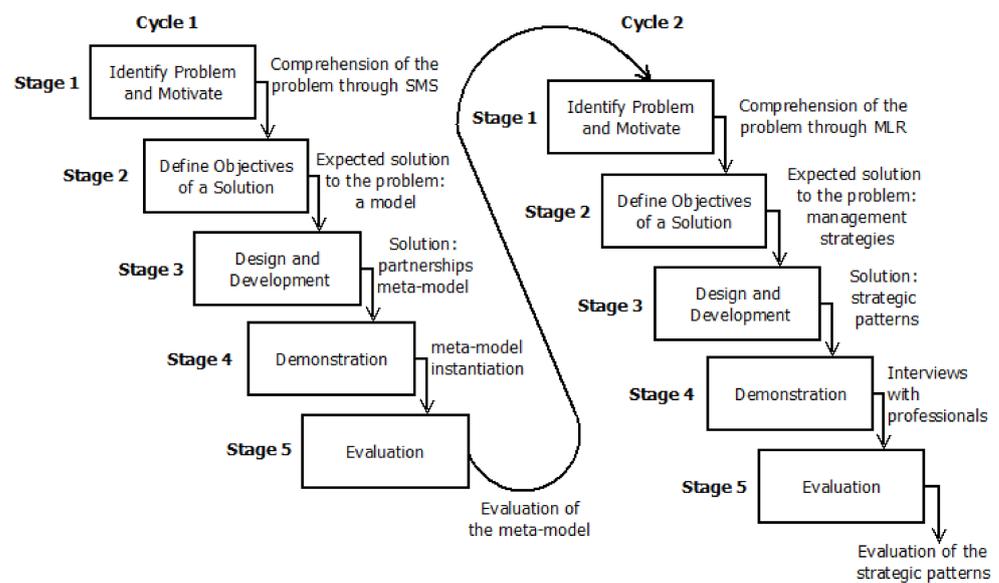
**RQ2.** How are the elements of partnership models related?

**RQ3.** What strategies can be adopted to define partnership models during the birthstage of a software ecosystem?

We adopted the DSR process proposed by Peffers [33]. That process was chosen because it focuses on the execution of Design Science Research in the Information Systems area and is widely referenced in the literature. Figure 1 describes the stages of Design Science Research as described in Peffers [33] and presents how we conducted each stage of the research. In this paper, we report the results of two DSR cycles. We explain how each stage was performed in the following sections.

#### 3.1. Identify Problem and Motivate

In this stage, a Systematic Mapping Study (SMS) and a Multivocal Literature Review (MLR) were performed in the first and second DSR cycles, respectively. An SMS synthesizes available peer-reviewed primary studies, characterizing relevant concepts on themes and identifying the maturity and comprehensiveness of a research area [34]. In contrast, an MLR aims to aggregate studies from the gray literature and peer-reviewed papers to obtain knowledge from academic and professional perspectives, closing the gap between academic research and professional practices [35].



**Figure 1.** Stages executed of Design Science Research process and respective results.

The SMS was previously published in Reference [10], and its results were used as input in this study for creating the partnership meta-model proposed. The goal of the SMS was to investigate the literature about partnership models in software ecosystems. The MLR was executed to identify strategies that can be used by organizations willing to create an ecosystem from a software product, considering the impact of that transition in the definition of partnership models. The primary studies of the SMS and the MLR are listed in the Appendix A.

The SMS was conducted by using a hybrid search strategy [36,37]. We performed an automatic search at Scopus Digital Library. Then, we used snowballing to complement the set of identified primary studies. After applying the inclusion and exclusion criteria, we selected 23 papers published between 2006 and 2018. Finally, we used open coding [38] to identify the partnership models' objectives and benefits described in the primary studies. After completing the mapping, we observed a lack of theoretical understanding about which strategies can support the partnership models definition for software ecosystems created from independent software products.

Furthermore, the SMS served as a basis for understanding the studied phenomenon, the main stakeholders, the problem, the causes of the problem, the envisaged effects of the problem resolution, and the problem-solving goals. The analyzed phenomenon consists of partnership models in software ecosystems. In that context, the main stakeholders are keystones and complementors who are the actors of the ecosystem.

The research problem is the need to understand the characteristics and relationships involved in creating partnership models switching from a software product to an ecosystem. Moreover, we aim to investigate strategies needed to tackle the impact of that change in the definition of partnership models. The problem source is the lack of an artifact that enables, through a global and integrated vision, defining suitable partnership models and managing the factors that impact those models during the birth stage of an ecosystem. An artifact of that nature enables the management of partnerships more efficiently, enabling the development of prosperous alliances, thus fostering the creation and sharing of value in the ecosystem.

During the switch from an independent software product to an ecosystem, companies should manage factors that can generate problems in the partnership model, such as conflicts between partners, platform quality, and partners' maintenance and attraction. Therefore, we run the MLR to identify practical strategies that can be adopted by organizations to conduct the switch from a software product to an ecosystem and address these

factors that affect the partnership models. The MLR followed the steps shown in Figure 2 and was guided by RQ3.

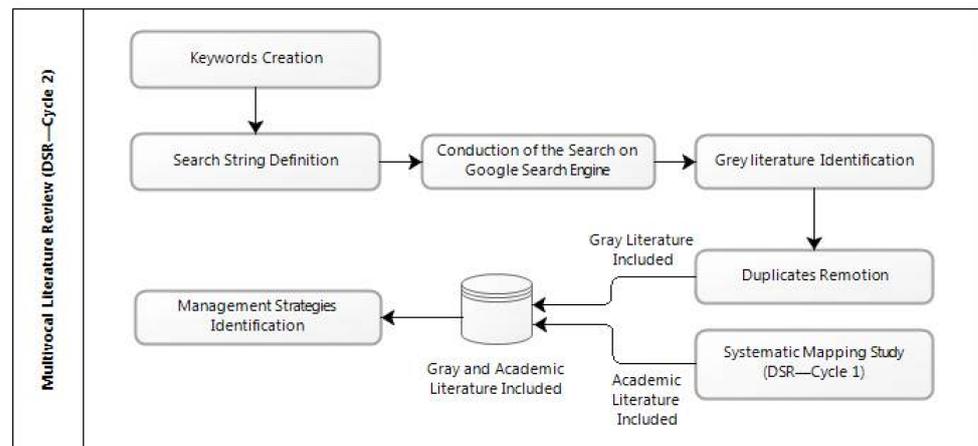


Figure 2. Multivocal Literature Review steps.

The keywords used to create the search string were obtained from previously known gray and academic literature. The search strategy was performed on Google Search Engine, similar to other Multivocal Literature Reviews [39,40]. To restrict the search scope, it was taken into account that Google Search Engine presents the most relevant results on the first pages [39–41]. Therefore, as in Garousi and Islam [39–41], we analyzed the first 10 pages returned, and the analysis continued to the next pages only if it was necessary.

Different strings were tested on Google Search Engine. For each one, the first 10 results obtained in the search were analyzed, and the string that made it possible to obtain more relevant search results was chosen. The selected string was the following: ((platform) AND (sided OR digital OR business OR product OR ecosystem OR market)). We chose that search string because software ecosystems are implementations of the business model based on platforms, also known as “multi-sided platforms”.

After the search, which comprised the period from 2016 to 2020 and returned 130 results, the duplicate studies were removed. Then, the inclusion and exclusion criteria presented in Table 1 were applied, and 23 papers were included. It is important to emphasize that the exclusion of studies was peer-reviewed and agreed between the authors.

Table 1. Inclusion and exclusion criteria.

Inclusion Criteria	Description
IC1	URL works.
IC2	Indicate strategies that positively influence the creation and maintenance of ecosystems.
EC1	If the webpage is only videos, audio, or images without text, it should be excluded.
EC2	Article from Quora, Slideshare, or LinkedIn, because on Quora and LinkedIn, the discussion happens without a detailed reflection. Similarly, Slideshare provides presentation material on several subjects but also lacks detailed reflection.

After applying the inclusion and exclusion criteria in the gray literature, 23 academic papers from SMS were added to the included studies set of the Multivocal Literature Review, thus totaling 46 included papers in the MLR. Finally, the studies metadata (study title, URL, and year of publication) and description of strategies for creating a platform ecosystem (through open coding [38]) were extracted. Both authors performed the extraction and discussed the differences that arose. We emphasize that the synthesis of the

extracted information was also carried out. We explain the results in the next stage of the DSR cycle (Design and Development), as described in Section 3.3.

### 3.2. Define Objectives of a Solution

This stage was carried out by analyzing the problem definition established in Stage 1 (Identify Problem and Motivate) of the Design Science Research cycle. The desired solution is a model and management strategies that have the objective to support the creation of partnership models when changing from a software product to an ecosystem, taking into account the impact of that change in the partnership model definition.

The artifacts to be developed in this study (the model and the management strategies) should help keystones and complementors to understand, from a holistic and integrated perspective, how partnerships in software ecosystems work and what factors (platform quality, for example) need to be managed when forming alliances. Moreover, the desired solution needs to support the keystone in adopting strategies to treating these factors and creating an ecosystem from a software product, as well as helping complementors to understand what they need to do to thrive in the alliance formed between them and the keystone.

### 3.3. Design and Development

This stage involves the generation of the solution for the studied problem, which can be a model, method, or instantiation [30,42]. The proposed solution is a partnership meta-model, together with strategic patterns to support the migration from a software product to an ecosystem, addressing the impact of that change in the partnership model definition. The meta-model was created through a 2-step approach proposed by Brambilla [43] and based on Object Management Group, Inc. (Needham, MA, USA) [44].

In Step 1, the concepts of the partnership meta-model were identified in the 23 primary studies investigated in the Systematic Mapping Study. The definition of each concept of the meta-model was obtained from primary studies and other sources available in the literature because some concepts were not defined precisely by the primary studies. It is important to highlight that each concept was represented as a class of the meta-model.

In Step 2, the primary studies were analyzed, and the relations among characteristics of partnership models were defined and represented in the meta-model. These relationships were structured as propositions, as proposed by Sjøberg [45]. These propositions can be accepted or refuted when analyzing the investigated context, so they are empirically traceable and testable.

After defining the partnership meta-model, strategic patterns were created. These patterns present strategies to help the keystone migrating from a software product to an ecosystem, treating factors that impact partnership model's definition (e.g., complements and platform quality). The proposed patterns were obtained from the synthesis of the Multivocal Literature Review. We chose to use patterns because they are widely used in Software Engineering and Business literature to document reusable knowledge [46–49].

### 3.4. Demonstration

In this stage of DSR, the researcher must demonstrate the proposed artifact by applying it to the problem context. The validation can use experimentation, simulation, case study, or another suitable approach [32,33]. In this study, two demonstrations were carried out, one with the meta-model and the other with the strategic patterns.

#### 3.4.1. Meta-Model Demonstration

In the first demonstration, the application of the artifact in the problem context was accomplished by instantiating the meta-model in three successful software ecosystems: SAP, Microsoft Azure, and Eclipse. These ecosystems have organizations as partners (complementors), use mechanisms to attract and maintain partners, and have as keystone an organization or a consortium. That vision is in line with the target audience of the

proposed meta-model: organizations that wish to create a software ecosystem individually or through a consortium, whose partners will also be companies that need to be attracted and maintained in the ecosystem. It is also important to highlight that the partnership models adopted by these ecosystems to promote the attraction and maintenance of affiliates are essential for companies to succeed during the birth phase of software ecosystems [1,2,24,26].

### 3.4.2. Strategic Patterns Demonstration

In the second demonstration, we conducted interviews with 6 professionals. The goal of the interviews was to compare the proposed strategic patterns with industrial professionals' perceptions about strategies that organizations can use during the birth phase of software ecosystems. It is important to highlight that the creation of the patterns considered the industry perspective since they were defined based on gray literature through the Multivocal Literature Review.

As the strategic patterns were designed to help companies migrating from a software product approach to an ecosystem, six interviews were conducted with professionals who participated in that change type. The analysis of these interviews was carried out according to guidelines by Seaman [50] and Cruzes [51]. First, the audios of the six interviews were verbatim transcribed. Then, the analysis of these transcripts started with open coding. The codes created were associated with the text excerpts, as presented in Figure 3.

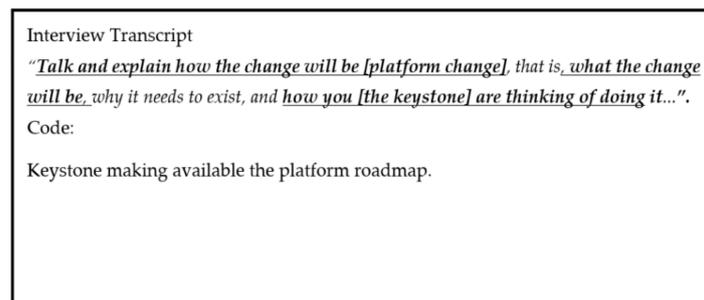


Figure 3. Open coding: creating codes.

After the constant comparison approach was used, that is, the codes generated in each interview were compared with the codes in the same interview and others. From these comparisons, the codes were grouped into the following categories: keystone’s actions and partners’ actions for switching from a software product to a platform ecosystem, treating the impact on partnership models (see Figure 4). After that, we identified relationships among these categories and the platform quality, as well as the support for partners, as described in Figure 5.

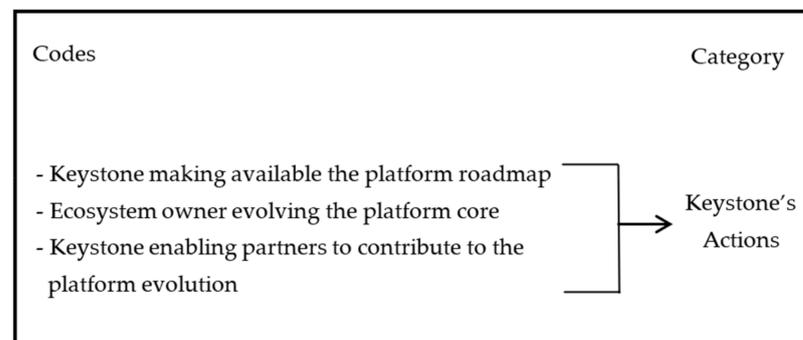


Figure 4. Open coding: creating categories.

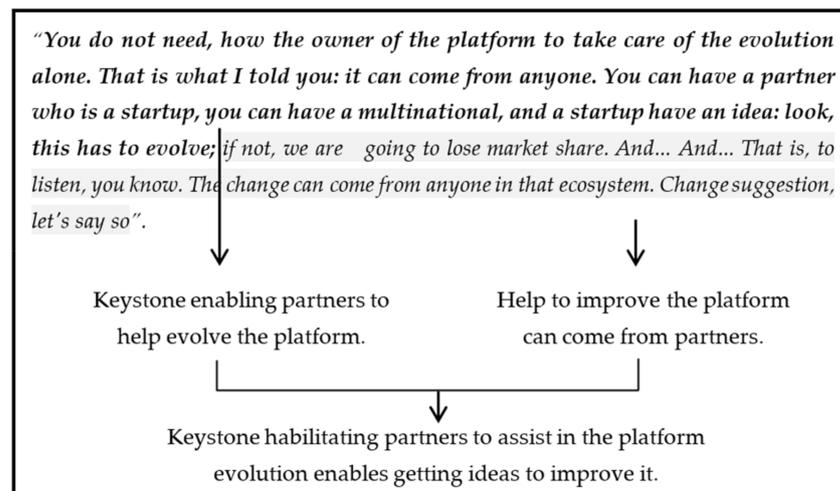


Figure 5. Open coding: creating relations.

### 3.5. Evaluation

In this stage of DSR, the researcher must evaluate how effectively the proposed artifacts support the problem solution, comparing the results of the previous stage (Demonstration) with those obtained in Stage 2, i.e., Define Objectives of a Solution of the DSR process [33].

The demonstration of the proposed partnership meta-model and the strategic patterns was compared with the objectives defined for the solution. As part of the proposed solution, we developed a meta-model to guide the creation of specific partnership models for software ecosystems. Therefore, we analyzed the meta-model to verify if its elements and relationships are present in SAP, Eclipse, and Microsoft Azure ecosystems. The presence of these elements and relations in these ecosystems indicates whether the proposed meta-model can be instantiated to define partnership models for novel software ecosystems.

The second part of the proposed solution is the strategic patterns for creating an ecosystem from an independent software product. The patterns aim to operationalize the concepts of the partnership meta-model. Moreover, the patterns aim to deal with aspects that affect the definition of partnership models, such as selecting adequate partners, enabling effective communication, and establishing subsidies to attract participants. Therefore, it was verified whether the strategic patterns are compatible with the management practices identified after analyzing industry professionals’ interviews. The profile of the interviewees is shown in Table 2.

Table 2. Profiles of the interviewed.

Interviewee No.	Role	Company	Ecosystem Launched	Company Information
1	Integration Manager	A	2020	Market area: ERP; Annual billing: USD 5,369,897.18.
2	Marketing Director	B	2016	Market area: Financial sector; Market Capitalization: USD 1,795,654,999.18. (Source: Ibovespa, the main stock trading market for publicly traded companies in Brazil).
3	Latin America General Manager	C	2016	Market area: Digital Experience Platform (DXP). It is among the 3 leaders in the Magic Quadrant for DXPs, according to Gartner.
4	QA Engineer			
5	QA Engineer and Consultant			
6	Software Engineer			

## 4. Results

In this section, the results of the following stages of DSR are presented: Design and Development, Demonstration, and Evaluation. In these stages, the partnership meta-model and the strategic patterns were designed and evaluated. Section 4.1 presents the elements and attributes of partnership models for software ecosystems. Section 4.2 displays the propositions of the proposed meta-model. Section 4.3 shows the complete partnership meta-model. In Section 4.4, the meta-model demonstration is shown. Section 4.5 presents the evaluation of the partnership meta-model. In Section 4.6, we present the strategic patterns. Section 4.7 shows the demonstration of these patterns. Finally, in Section 4.8, we present the evaluation of the strategic patterns.

For the sake of brevity, the elements and attributes definition of the partnership meta-model was made available at the following link: <https://sites.google.com/view/meta-model> (accessed on 8 May 2021). In addition, in this paper, we show an instantiation of the meta-model. The others instantiations are in that link, which also contains additional information on the patterns validation.

### 4.1. Partnerships Elements and Attributes in Software Ecosystems

In this section, we answer **RQ1: What are the key elements of partnership models in software ecosystems?** We found 14 elements that have 18 attributes, which are constructs that represent characteristics of partnership models. These elements and attributes were identified in the primary studies of the Systematic Mapping Study previously published in Reference [10]. They are the basis for creating the proposed partnership meta-model for software ecosystems.

The elements and attributes describe characteristics (concepts) present in partnerships that need to be governed by the keystones or complementors. The goal element, for example, is a concept that represents the central purpose of the partnership creation, and the risk attribute represents the risks of the partnership between the keystone and its partners. Therefore, keystone and complementors (partners) need governing mechanisms (benefits and requirements, for example) that promote the achievement of their objectives and treating the partnership risks. The elements and attributes, together with their origin studies, are presented in Tables 3 and 4, respectively. According to References [43,44], a meta-model is the abstract representation of some observed phenomenon (such as partnership models) and can be used to instantiate that phenomenon in some context.

**Table 3.** Elements of the proposed partnership meta-model and primary studies of origin.

Elements		Studies
	Actor	All
	Complementor	All
	Keystone	All
	Role	[S1,S11,S16]
Goal	Of Complementor	[S1,S5,S11,S15,S16,S18]
	Of keystone	[S1,S3,S6,S11]
Partnership Management	Platform management	[S6,S7,S11,S22,S23]
	Document management	[S1,S6,S9,S11,S16,S17,S19]
	Risk management	[S12,S15,S17,S20]
	Conflict management	[S6,S11,S16,S17,S23]
	Partner Cluster Management	[S1,S6,S8,S9,S11,S16,S17,S20,S21]
	Requirement	[S1,S6,S9,S11,S17]
	Benefit	[S1,S6,S11,S15]
	Partnership Level	[S1,S6,S11]

**Table 4.** Attributes of the proposed partnership meta-model and studies of origin.

Attributes	Studies
Partner Training	[S1,S5,S6,S11]
Partner Performance	[S1,S2,S4,S6,S9–S11]
Marketing Support Strategy	[S1,S5,S8,S11,S16,S15]
Communication Strategy	[S5,S6,S11,S15–S17,S23]
Sales Channels and Distribution	[S1,S5,S6,S8,S11,S15,S16,S18,S22]
Monetization Strategy	[S1,S6,S16,S18,S23]
Network Effect Strategy	[S4–S6]
Partners Profile Definition	[S1,S6,S11]
Keystone Performance	[S3,S8,S9,S21]
Platform Evolution Strategy	[S5,S6,S23]
Platform Integration Strategy	[S5,S7,S11,S22,S23]
Product and Service Homologation	[S1,S5,S6,S11,S16,S23]
Document	[S1,S6,S9,S16,S17,S19]
Document Management Strategy	[S1,S6,S9,S11,S16,S17,S19]
Risk	[S12,S15,S17,S20]
Risk Treatment	[S15,S17]
Conflict	[S6,S11,S16,S17,S23]
Conflict Treatment	[S6,S11,S16,S17,S23]

Therefore, after conducting the SMS, the elements and attributes found in the primary studies were represented in a meta-model enabling that companies to create (instantiate) their specific partnership models in the context of the birth phase of the software ecosystem creation.

It is essential to highlight that the primary studies address partnerships in software ecosystems and, of course, consider the existence of complementors and keystone. Therefore, as indicated in Table 3, the elements actor, keystone, and complementor were identified in all primary studies.

#### 4.2. Propositions of the Meta-Model Proposed

In this section, we answer **RQ2: How are the elements of partnership models related?** These relationships are represented as *Propositions 1–4 (P1–4)* between the elements (see Table 3) of the meta-model that were extracted from the primary studies. Each proposition is supported by evidence from the literature and information from case studies conducted at the SAP, Eclipse, and Microsoft Azure ecosystems. In the following paragraphs, we discuss that evidences.

##### **P1: Actor fulfills a role, which defines how it will act in the ecosystem.**

**Evidence:** Software ecosystem partnership models guide the execution of various activities, such as making the ecosystem platform available, developing solutions, and selling these solutions. These activities are executed by actors in accord with the role that fulfills. The keystone plays the role of ecosystem orchestrator, which is responsible for providing and maintaining the software ecosystem platform, as well as the governance of the entire ecosystem. In this context, complementors (the keystone's partners) develop solutions for the ecosystem and sell these solutions, filling roles such as value-added reseller, solution developer, or software service provider.

##### **P2: Role defines the actor's level in the partnership, its responsibilities, and the benefits that it can receive.**

**Evidence:** Keystone and its partners (complementors) fulfill roles in the software ecosystem. However, only complementors have some partnership level associated with their role, for example, gold, silver, or bronze. These roles and levels define the partners' benefits, responsibilities (requirements to be satisfied), and progression in the partnerships [S1,S6,S11].

According to role and partnership level, the complementors receive benefits as marketing assistance, revenue flow, access to the keystone consumer base, access to specific ecosystem information, and permission to use the source code of software products [S1,S6,S11]. Moreover, the partners (according to their role and partnership level) need to satisfy the following requirements defined by the keystone: completion of a certification process (for services, tools, or software solutions), not participating in a competing ecosystem, and allocating a specific number of full-time developers to create software for the platform, for example [S1,S9,S11].

In this context, the keystone has no partnership level. Its role defines its benefits in the alliances, such as extending the cluster of partners, co-innovate with affiliates, and obtaining complementary products on the platform [S1,S6,S11]. In addition, the role defines the requirements for the keystone, such as providing means of conflict resolution, information sharing, and effective communication between partners [S6,S11]. The satisfaction of these requirements positively impacts the creation and maintenance of alliances.

**P3: Actor has goals that are met through the management of the partnerships.**

**Evidence:** Keystone and complementors (ecosystem actors) form partnerships to achieve different objectives. The achievement of these objectives indicates whether the management mechanisms employed by keystone are being effective [S1,S3,S5,S11].

The keystone seeks to protect the relationship with clients, access complementary resources, extend the partner ecosystem, and co-innovate with affiliates [S1,S3]. While complementors seek better access to the consumer market, better satisfy customer demands, and integrate their products into the ecosystem software platform [S5].

**P4: The partnerships management is executed through a set of strategies that enable the keystone and its partners to achieve their goals.**

**Evidence:** To achieve its goals (as well as its partners), the keystone uses different management strategies to attract and maintain affiliates and promote the ecosystem platform quality, making the software ecosystem thrive.

Therefore, the keystone uses management mechanisms that range from partner support to developing and selling solutions in the ecosystem [S1,S5,S6,S11] to partner evaluation, revenue models, and network effects promotion [S4–S6,S15].

Another way a keystone manages its partnerships is to use and manage documents such as privacy policies, by-laws, and performance reports [S1,S6,S9,S11,S16,S17,S19]. In addition, a keystone can employ strategies for organizing platform evolution [S5,S6,S23], maintain partner solutions' quality, manage risk [S12,S15,S17,S20], and handle conflicts in partnerships [S6,S11,S16,S17,S23].

#### 4.3. Partnership Meta-Model for Software Ecosystems

This section presents the proposed partnership meta-model, shown in Figure 6. The meta-model enables a global and integrated view of the elements and attributes as well as relations present in partnership models. These elements are represented by boxes (e.g., role and actor) with attributes, and the propositions are represented by lines indicating relations between elements, for example, the preposition P1 (that indicates the relation between actor and role).

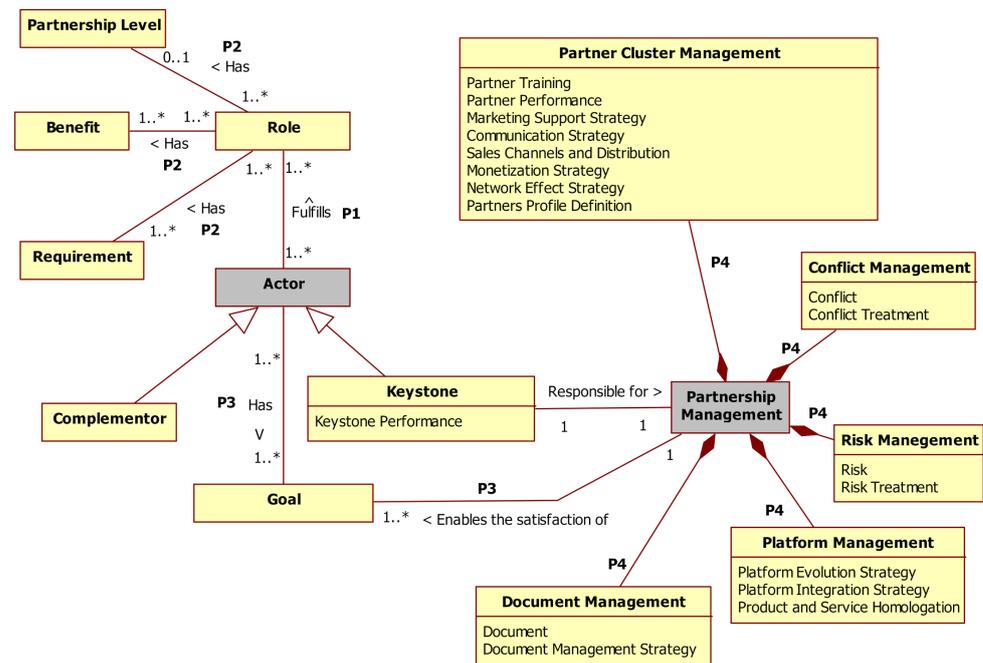


Figure 6. Partnership meta-model.

In the meta-model, the central elements are actor and partnership management. The first is considered a key element of software ecosystems because it represents the keystone and the complementors (the two actors that form alliances governed by the partnership model of any software ecosystem). Similarly, partnership management deals with essential mechanisms for orchestrating alliances.

It is important to emphasize that all meta-model elements and attributes are necessary to take into account to instantiate partnership models, since it is their joint orchestration that enables the successful creation and evolution of the software ecosystem [1,2,20].

#### 4.4. Demonstration of the Partnership Meta-Model for Software Ecosystems

This section presents the proposed meta-model application based on Wieringa [32] and Peffer [33]. The goal of the meta-model is to be a reference guide to create specific partnership models for companies initiating software ecosystems from a software product.

Therefore, it must be possible to instantiate the meta-model for ecosystems owned by a company or consortium that uses mechanisms to attract and maintain partners.

In our study, the meta-model was instantiated for the Eclipse, SAP, and Microsoft Azure ecosystems. These ecosystems were chosen because they are owned by a company or consortium. Moreover, although they are mature, they use the partner’s attraction and maintenance strategies that are essential to migrate from a software product to an ecosystem [1,2,9,10]. Table 5 presents details of these ecosystems, and Figure 7 shows the meta-model instantiation for the SAP ecosystem.

In Figure 7, we highlight the central elements of the SAP partnership model. They are keystone (SAP Company, Walldorf, Germany), the complementor (Partner Organization of the SAP), and partnership management (SAP’s partnerships management areas), which is composed of conflict management, document management, risk management, platform management, and partner cluster management.

Table 5. Eclipse, SAP, and Microsoft Azure ecosystems overview.

Eclipse (Open Source)	SAP (Proprietary)	Microsoft Azure (Proprietary)
Partnership Model	Partnership Model	Partnership Model
<b>Name:</b> Eclipse Membership-at-Large.	<b>Name:</b> PowerEdge Partner Program.	<b>Name:</b> Microsoft Partner Network.
<b>Partners Roles:</b> Strategic, Enterprise, Solutions, Committer, and Associate.	<b>Partners Roles:</b> Build Solutions, Sell Solutions, Service Solutions, and Run Solutions.	<b>Partners Roles:</b> Microsoft Partner Network, Microsoft Action Pack, and Competency.
<b>Roles Meaning:</b> Partnership levels that reflect the involvement degree of affiliates in the ecosystem, from an initial level with less influence to a strong influence on governance.	<b>Roles Meaning:</b> Functions performed by the partners.	<b>Roles Meaning:</b> Partners classification according to benefits received of the keystone and requirements to be met.
<b>Observation:</b> Eclipse’s partnership model is used to manage alliances with private and open source companies [52].	<b>Observation:</b> Partners can occupy more than one role, and their evolution is done through levels within each role (silver, gold, or platinum) [53].	<b>Observation:</b> The Microsoft Partner Network is the gateway to form a partnership and does not have a division in levels. The Microsoft Action Pack, on the other hand, has more benefits and does not have any division into levels, while the role with more resources is the “competency”, which has two levels: silver and gold [54,55].

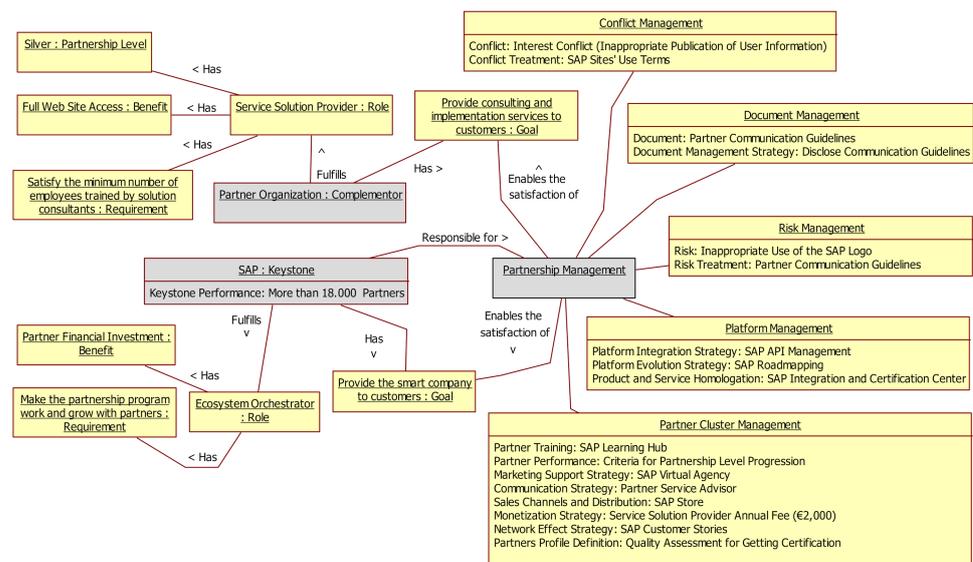


Figure 7. Instantiation of the partnership meta-model for the SAP ecosystem.

The keystone (SAP Company) has more than 18,000 partners and, in its role as orchestrator of the ecosystem, it has objective, benefit, and requirement. SAP satisfies its objectives through partnership management. Each part of the partnership management shows a way to carry out that task. For example, partner cluster management indicates, among others, a strategy for partner training (SAP Learning Hub), monetization (Annual Fee charged to the partner who fulfills the role Service Solution Provider), and marketing support for partners (SAP Virtual Agency). Finally, the complementor (Partner Organization) has the role (Service Solution Provider), in which it has a level (Silver), benefits (Full Website Access), and requirement to be satisfied (Satisfy the Minimum Number of Employees Trained by Solution Consultants).

It is important to highlight that the meta-model has a level of abstraction that allows its instantiation in different ways. Figure 7 shows one instantiation possibility, but there are others.

For example, the element of risk management has one risk and one mechanism for its treatment, but it could have several risks and several treatment mechanisms for them (including a risk management process). That logic applies to the instantiation of all elements of the meta-model.

#### 4.5. Evaluation of the Partnership Meta-Model for Software Ecosystems

In this section, in conformity with Peffers [33], we verify if the objective defined for the solution, in Stage 2 of Cycle 1 of the DSR process, was achieved. That is, we verify if, in the demonstration described in Section 4.4, the meta-model elements, attributes, and relationships are present in existing ecosystems, which shows whether companies can use the meta-model proposed to create specific partnership models for their software ecosystems.

In the meta-model demonstration, all the elements, attributes, and relationships of the proposed meta-model were found in different software ecosystems' partnership models. That result shows that the meta-model can be used to create specific partnership models, which indicate that objective defined for the solution, specified in Stage 2 of Cycle 1 of the Design Science Research process, was achieved.

#### 4.6. Strategic Patterns

In this section, we answered **RQ3: What strategies can be adopted to define partnership models during the birth stage of a software ecosystem?** We identified strategies through the analysis of 46 studies from a Multivocal Literature Review (MLR). These strategies were analyzed and classified through the codification process [50,51] in the following patterns:

- **Pattern 1**—Platform Quality.
- **Pattern 2**—Support for the Partner.
- **Pattern 3**—Participant's Attraction and Maintenance.
- **Pattern 4**—Participant's Profile.

The strategic patterns were created according to the template in Table 6, based on Hoffmann [56] and Xuan [57]. After creating the patterns through the MLR studies analysis, they were validated and refined through interviews with six industry professionals. After the analysis of the interviews, we generated the final version of each pattern. These patterns serve to operationalize the meta-model proposed in this study. In this way, they serve as reference for organizations to define their partnership model when changing from an independent software product to an ecosystem. Table 7 shows the purpose of each pattern.

**Table 6.** Template of the strategic patterns.

<b>Name:</b> Pattern identification.	<b>Objective:</b> What the pattern does.
<b>Source:</b> Origin of the strategies and practical applications.	
<b>Strategy:</b> Action that must be taken to achieve the goal.	
<b>Practical Application:</b> Practical examples of the strategy.	

**Table 7.** Strategic pattern's purpose.

Pattern	Objective
Platform Quality	Enable the maintenance of the software ecosystem Platform Quality.
Support for the Partner	Support the partners to develop and sell solutions in the ecosystem.
Participant's Attraction and Maintenance	Enable the attraction and maintenance of partners (complementors) and consumers.
Participant's Profile	Enable the participation of partners and customers in the ecosystem, according to keystone's need.

It is relevant to note that two patterns were concluded (Platform Quality and Support for the Partner), and two are in the validation phase (Participant's Profile and Participant's Attraction and Maintenance). In the following section, we present the pattern Platform Quality.

#### Pattern 1: Platform Quality

Creating an ecosystem from an independent software product requires the software platform to be complemented with solutions from various partners. These solutions need to have the adequate quality to avoid platform depreciation and, consequently, the degradation of the entire ecosystem [1,2,58]. Therefore, we proposed the Pattern Platform Quality. That pattern has the knowledge to maintain the software ecosystem platform's quality. Table 8 shows part of the pattern with its three strategies and practical applications.

**Table 8.** Pattern 1—Platform Quality.

<b>Name:</b> Platform Quality.	<b>Objective:</b> Enable the maintenance of the software ecosystem Platform Quality.
<b>Source: References</b> [S5,S7,S11,S15,S16,S22,S23]; References [G1,G2,G4,G8,G9,G13,G14,G16,G17,G19,G21,G22]; Interviewees 1, 2, 3, 4, 5, and 6.	
<b>Strategy 1: Define the platform mechanisms for integration with partners' solutions.</b>	
<b>Practical Application</b>	
The keystone provides documentation on how to integrate partner solutions into the ecosystem platform, using API (Application Programming Interface) and SDK (Software Development Kit). That documentation can be an integration manual.	
<b>Strategy 2: Establish the platform's evolution approaches.</b>	
<b>Practical Application</b>	
The keystone and complementors make available their product roadmap for collaboration in defining the future evolution of their products, but without revealing critical information about their business strategy.	
<b>Strategy 3: Define mechanisms to have partner solutions with adequate quality.</b>	
<b>Practical Application</b>	
The keystone conducts training and provides a certification program for partners.	

#### 4.7. Demonstration of the Strategic Patterns

This section shows the strategic patterns validation based on Wieringa [32] and Pefers [33]. The strategic patterns' goal is to operationalize the meta-model proposed in this study (as defined in Figure 6) to support organizations in defining specific partnership models when creating an ecosystem. Therefore, these strategic patterns must be in line with the professionals' perspective on strategies for moving from a software product to an ecosystem. Given this, interviews were carried out with six industry professionals who participated in this type of transition to identify how to do it properly. After that, the results of these interviews were compared with the strategic patterns, which were proposed based on the Multivocal Literature Review. More details on the validation (demonstration) of the patterns are shown in the following paragraphs.

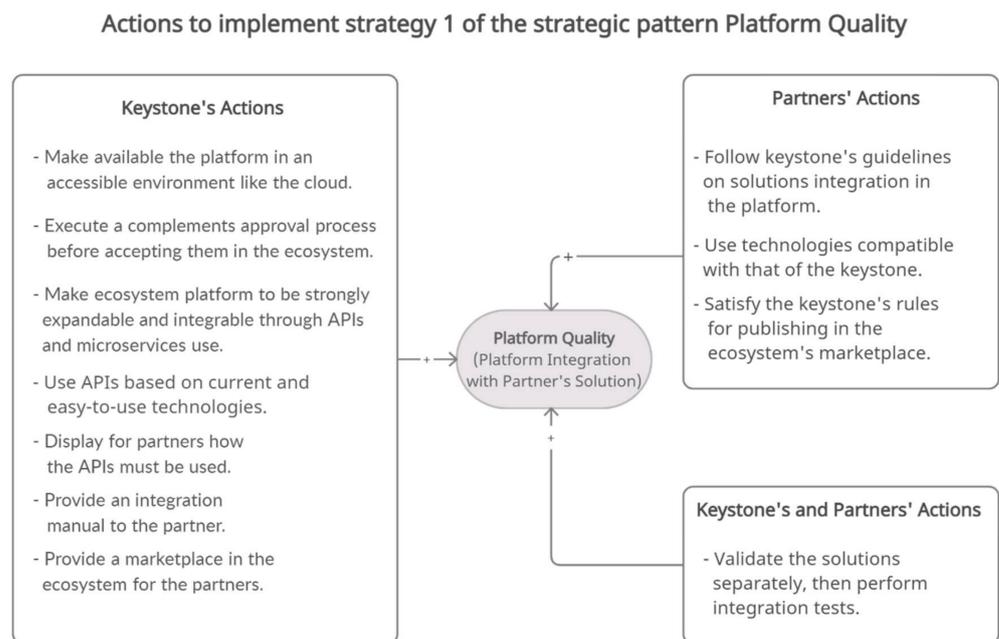
We conducted the interviews following the script shown in Appendix B. The script was developed based on the four strategic patterns proposed. The interviews' objective was to verify the interviewees' perception about how the migration from a software product to an ecosystem can be performed. After conducting the interviews, the audios were transcribed, and the transcriptions were analyzed by using codification [50,51]. From the analysis, we identified a set of actions that the keystone or its partners can perform to conduct a migration from an independent software product to an ecosystem, considering the decisions to be made on the partnership model.

After that, we checked if the actions identified in the interviews were present in the proposed strategic patterns. Some actions were in these patterns, and others not. Those that were not initially included, we added to the patterns.

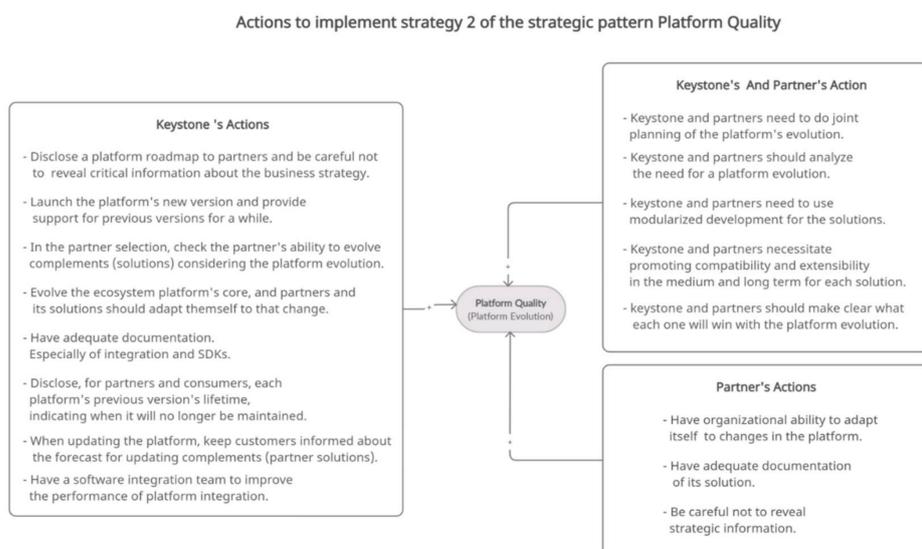
Figures 8–10 show the coding process results regarding the interviews about the quality of the platform. That process was used to analyze and synthesize the interview transcripts [50,51]. We present in Appendix C some transcripts of these interviews that served as input for the coding process explained in Section 3.4.2. (Strategic Patterns Demonstration).

The interviews’ analysis and synthesis, through the coding process, identified actions that the keystone or its partners can perform and impact the quality of the platform.

Figure 8 shows a set of actions identified in the interviews, and that positively impact the platform quality, specifically the platform integration with complements (partners’ solutions), and are necessary to implement Strategy 1 (define the platform mechanisms for integration with partners’ solutions) of the pattern Platform Quality.

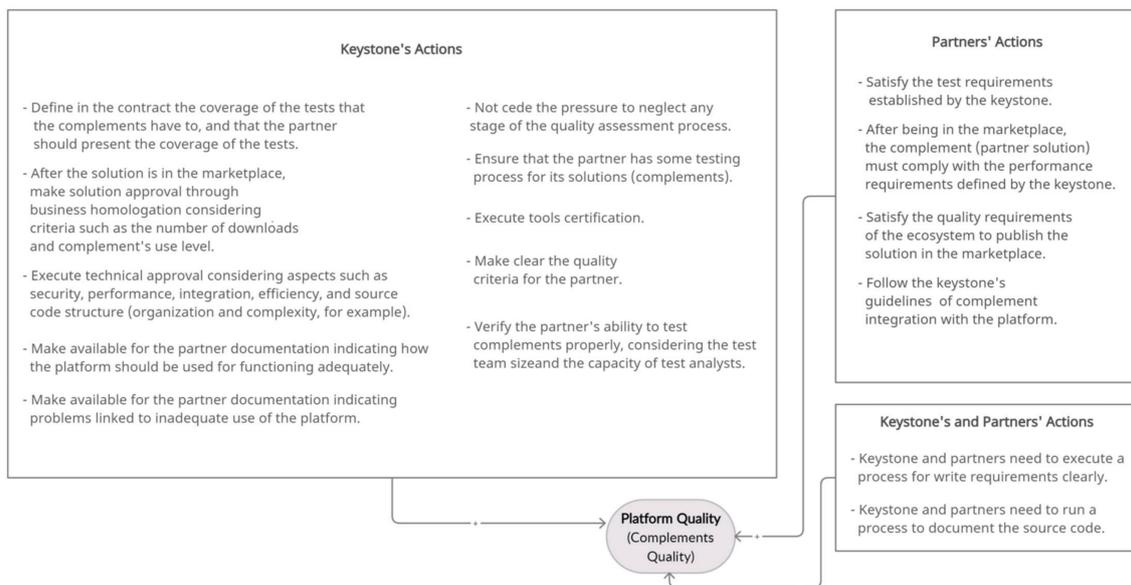


**Figure 8.** Result of the analysis of the interviews about the quality of the platform: actions that positively impact the platform quality (platform integration with partners’ solution).



**Figure 9.** Result of the analysis of the interviews about the quality of the platform: actions that positively impact the platform quality (platform evolution).

Actions to implement strategy 3 of the strategic pattern Platform Quality



**Figure 10.** Result of the analysis of the interviews about the quality of the platform: actions that positively impact the platform quality (complements quality).

Figure 9 shows several actions identified in the interviews that positively impact the platform quality (more precisely, the platform evolution) and are necessary to implement Strategy 2 (establish the platform's evolution approaches) of the pattern Platform Quality.

Figure 10 shows actions identified in the interviews that positively impact the platform quality (that is, the complements quality) and are necessary to implement Strategy 3 (define mechanisms to have partner solutions with adequate quality) of the pattern Platform Quality.

#### 4.8. Evaluation of the Patterns Proposed

In this section, following the indications in Peffers [33], we verified if the objective defined for the solution in Stage 2 of Cycle 2 of the Design Science Research was achieved. That is, we checked if the demonstration (validation) described in Section 4.7 indicates the following: whether the strategic patterns can be used to operationalize the meta-model proposed and are in line with industry professionals' view on how to switch from a software product to an ecosystem.

These results show that the strategic patterns are aligned with professionals' perspectives about what strategies can be used to move from an independent software product approach to an ecosystem.

### 5. Discussion

The alliances in the software ecosystem between keystone and partners are orchestrated through partnership models, which enable the healthy growth of the ecosystem. In that context, different perspectives of these models were studied, such as characteristics of the partnership model (considering the definition of roles, benefits, and responsibilities of the keystone and its partners) [S1] [59], incentives to attract partners [S5,S8,S9] [59], mechanisms of affiliate retention [S17,S19,S20,S22], and impact of the partnerships on the software ecosystem health [S2,S3,S21].

There are several relevant studies about different aspects of partnership models of software ecosystems. However, it is hard for researchers and managers to obtain a holistic and integrated view of key characteristics that partnership models need to have and factors that affect these models when creating an ecosystem from a software product [9,10].

Moreover, it is difficult for them (researchers and managers) to treat that factors in that changing context, which makes many organizations fail when migrating from a software product to an ecosystem. Examples of these factors are subsidies to attract participants, selection of suitable partners, assessment of the ecosystem solutions quality, risk analysis associated with opportunistic behavior, and stimulus of the network effects to attract consumers and partners [1,2,10,11].

This work aims to tackle these problems by proposing a partnership meta-model for software ecosystems that provide a global and integrated vision of the partnership models properties. Moreover, we propose strategic patterns to operationalize the meta-model proposed and guide the migration from a software product to an ecosystem, treating the impact of that change on the partnership model definition. Together, the partnership meta-model and the strategic patterns proposed in this work have practical and academic implications.

*Practical implications:* The meta-model enables managers and practitioners to understand in a comprehensive and integrated manner the characteristics of the partnership models and factors that impact the creation of that models and need to be addressed, such as platform governance, risk management, and partners' administration, considering their role, benefits, and responsibilities. Moreover, the meta-model can be used to analyze different contexts of the organization that want to switch from a software product to an ecosystem, thus verifying which aspects of that change were treated or not. That applicability is possible because the meta-model has a high level of abstraction and is based on the academic and industry perspectives through different information sources (literature review and case studies in different ecosystems).

Another practical implication is related to the strategic patterns proposed to operationalize the meta-model. These patterns have a set of strategies with respective practical applications that can be used by keystone or partners.

Managers can use the Platform Quality pattern as a source of strategies to manage the ecosystem platform quality. More specifically, the pattern Platform Quality can be used by keystone's or partners' managers to define strategies that will be used to make complements integration with the ecosystem's software platform feasible, carry out the evolution of that platform and have complements with adequate quality.

Regarding the pattern Support for the Partner, it can be used by the keystone's or partners' managers to set up a support structure for the partner to have success in the complements development and commercialization. That pattern has strategies that can be used to help partners reach new consumers and markets, be qualified to provide complements, and have adequate communication with the keystone.

*Academic implications:* The meta-model is an advance in the understanding of partnership models. It enables researchers to understand, in a global and integrated way, key characteristics of these models and their relationships in the context of structuring an ecosystem from a software product. Regarding strategic patterns, researchers can use them as a reference to add more strategies and practical applications needed to move from an independent software product to an ecosystem.

## 6. Conclusions

This research was carried out by using Design Science Research. We conducted several studies during two cycles of DSR: a Systematic Mapping Study; a Multivocal Literature Review; case studies with the SAP, Eclipse, and Microsoft Azure ecosystems; and interviews with industry professionals. Through the SMS, we identified the lack of an artifact in the literature that enables a broad and integrated perspective of partnership models for software ecosystems. To address this gap, we proposed the partnership meta-model, which has practical and academic implications.

Under the practical perspective, the meta-model enables the keystone's and partners' managers to understand, in a comprehensive and integrated manner, the characteristics that partnership models need to have and factors that impact the creation of these models during migration from a software product to an ecosystem. Examples of these factors are

platform quality, conflict management, and partner administration, considering monetization strategy, support in marketing, and network effects. Moreover, the meta-model enables the keystone's and partners' managers to decide which characteristics of the partnership model to address during the ecosystem creation, considering the reality faced. We emphasize that the meta-model can be used in other phases of the software ecosystem's life cycle in addition to the initial one, because it has characteristics that need to be managed in other phases of that life cycle.

Another practical implication of this research is related to the proposed strategic patterns to operationalize the partnership meta-model. These patterns have a set of strategies, with respective practical applications, that support companies in moving from a software product to an ecosystem and define their partnership model in that challenging shift context. It is important to note that keystone's and partners' managers of these companies can choose which patterns to use according to the current challenges that they are facing. Moreover, the patterns are flexible; they have varied strategies and practical applications that the managers can select according to the context and problems they are experiencing.

From an academic perspective, the meta-model enables researchers to understand, holistically, the partnership models' properties and the impact of the ecosystem creation from a software product on these models. Regarding the strategic patterns, it is also important to highlight that they have several practical applications of the strategies not identified in the literature analyzed through MLR. However, the professionals interviewed indicated these practical applications as possible actions to manage the transition from a software product to an ecosystem, taking into account the partnership model definition. Therefore, these patterns can serve as a reference for researchers to add more strategies and practical applications to make that transition.

### 6.1. Limitations and Threats to Validity

In this work, a limitation was the number of companies from which practitioners were chosen to participate in the interview to validate the strategic patterns. These practitioners work at three companies. To address that limitation, we chose companies that operate in different markets and professionals playing different roles. In future works, more companies can be selected as a source of professionals for interviews. We address potential threats to validity from the perspective of interpretive, qualitative, non-positivist research. We follow the perspective indicated in Merriam [60].

*Construct validity:* Construct validity is related to the precise definition of the studied concepts. The partnership meta-model was created through a Systematic Mapping Study. The well-defined review process ensured its creation was based on peer-reviewed academic literature. Moreover, the meta-model was verified through case studies of three distinct ecosystems. Regarding the strategic patterns, they were created based on gray and academic literature, obtained with a Multivocal Literature Review. After that, interviews with professionals from different functions and companies (that operate in distinct markets) were carried out. These interviews were realized to verify if our proposals of the patterns conformed with professionals' views regarding the strategies that can be used to migrate from an independent software product for an ecosystem, covering the partnership model's definition.

*Internal validity:* Internal validity refers to how much the researcher captured the reality and expressed it in the study results. To increase internal validity, different data sources were used. Regarding the partnership meta-model, data were obtained from the academic literature that investigated real ecosystems of worldwide scope and from case studies on the partnership model of three software ecosystems (SAP, Eclipse, and Microsoft Azure). Concerning the strategic patterns, we obtained data from the academic literature on real ecosystems and from the industry perspective by analyzing studies of gray literature through a Multivocal Literature Review (thus reducing the knowledge gap between academia and industry). After, we used member checking to validate the strategic

patterns and verify their pertinence and completeness. That was accomplished through interviews with industry professionals.

*External validity:* In this work, external validity was approached based on the concept of transferability, which indicates that it is possible to learn from the results and decide which contexts they are transferable or applicable.

To increase transferability, we used two approaches: (a) We sought to describe in a detailed and rich manner the research method, its context of execution, and the results obtained. We believe that such a description is a limitation of this article since the space limitation impacts the elaboration of detailed descriptions. To alleviate this issue, we included extra information regarding the generated artifacts on an external website. (b) We consulted the academic and gray literature, conducted three case studies of software ecosystems, and conducted interviews with professionals from different positions and companies of different markets. That diversity of data sources and contexts provided richer data and, therefore, more substantial results with broader applicability.

*Reliability:* According to Merriam [60], as human feelings and perceptions change, in qualitative research, the researcher does not expect to obtain replicability of the results. In this context, the most important is consistency, that is, that the results come from the research data consistently, meaning that the researchers do not make any inference that the research data cannot support. To increase consistency concerning the partnership meta-model, data collection and analysis were carried out by the first author and reviewed by the second, and the case studies followed the protocol indicated in Merriam [60].

Regarding the strategic patterns, the MLR that supported their creation was carried out by the two authors to increase consistency. Besides, member checking was carried out with professionals through interviews to validate the strategic patterns created.

It is essential to highlight that the codification of the interviews was carried out by the first author and reviewed by the second.

## 6.2. Future Works

This study proposed four strategic patterns, in which two patterns, Participant's Attraction and Maintenance and Participant's Profile, are in the validation phase. Therefore, a plan for future work is the conclusion of that validation.

We believe that further research can involve new interviews considering the perspective of professionals working in ecosystems that operate in different market domains in order to provide more strategies and respective practical applications for the proposed patterns. Another relevant line for future research consists of conducting longitudinal studies with emerging ecosystems to understand how they create their partnership models and establish strategies to foster their growth and evolution.

**Author Contributions:** Conceptualization, Í.B.; methodology, Í.B. and C.A.; writing—original draft, Í.B.; writing—review & editing, Í.B. and C.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data about search results can be found at: <https://sites.google.com/view/meta-model> (accessed on 1 June 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Studies Included in the Systematic Mapping Study (S1–S23) and Multivocal Literature Review (G1–G23).

- [S1] Angeren, J. V.; Kabbedijk, J.; Jansen, S.; Popp, K.M. A survey of associate models used within large software ecosystems. In: Proceedings of the Third Workshop on Software Ecosystems, Springer, 2011.
- [S2] Ceccagnoli, M.; Forman, C.; Huang, P.; Wu, D. J. Cocreation of value in a platform ecosystem: the case of enterprise software, *MIS Quarter*, 2012, 36(1), 263–290.
- [S3] Mukhopadhyay, S.; Reuver, M.; Bouwman, H. Effectiveness of control mechanisms in mobile platform ecosystem, *Telemat and Informat*, 2016, 33(3), 848–859. <https://doi.org/10.1016/j.tele.2015.12.008>.
- [S4] Angeren, J. v.; Jansen, S.; Brinkkemper, S. Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the Office365 ecosystem. In: International Conference of Software Business, Springer, Paphos, Cyprus, 16–18, 1–15, June 2014. [https://doi.org/10.1007/978-3-319-08738-2\\_1](https://doi.org/10.1007/978-3-319-08738-2_1).
- [S5] Rickmann, T.; Wenzel, S.; Fischbach, K. Software ecosystem orchestration: The perspective of complementors. In: Twentieth Americas Conference on Information Systems, Association for information systems, Savannah, GA, 7–10 August 2014.
- [S6] Jansen, S.; Peeters, S.; Brinkkemper, S. Software ecosystems: From software product management to software platform management. In: From Start-ups to SaaS Conglomerate: Life Cycles of Software Products Workshop (IW-LCSP 2013), Potsdam, Germany, 11 June 2013.
- [S7] Liu, X.; Lee, C.; Iyer, B. The impact of design moves on platform adoption: The case of Microsoft Windows OS. In: Proceedings of the 39th Hawaii International Conference on System Sciences, IEEE, Kauai, HI, USA, 4–7 Jan 2006. <https://doi.org/10.1109/HICSS.2006.466>.
- [S8] Kude, T.; Dibbern, J.; Heinzl, A. Why do complementors participate an analysis of partnership networks in the enterprise software industry, *IEEE Transac on EngManag*, 2012, 59 (2), 250–265. <https://doi.org/10.1109/TEM.2011.2111421>.
- [S9] Baars, A.; Jansen, S. A framework for software ecosystem governance. In: International Conference of Software Business, Springer, Cambridge, MA, USA, 18–20 June 2012. [https://doi.org/10.1007/978-3-642-30746-1\\_14](https://doi.org/10.1007/978-3-642-30746-1_14).
- [S10] Angeren, J. V.; Blijleven, V.; Jansen, S.; Brinkkemper, S. Complementor embeddedness in platform ecosystems: The case of Google Apps. In: 7th IEEE International Conference on Digital Ecosystems and Technologies, IEEE, Menlo Park, CA, USA, 24–26 July 2013. <https://doi.org/10.1109/DEST.2013.6611326>.
- [S11] Avila, A.; Terzidis, O. Management of partner ecosystems in the enterprise software industry. In: 8th International Workshop on Software Ecosystem, Dublin, Ireland, 10 December 2016.
- [S12] Angeren, J. v.; Blijleven, V.; Jansen, S. Relationship intimacy in software ecosystems: A survey of the Dutch software industry. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, ACM, November, 2011. <https://doi.org/10.1145/2077489.2077502>.
- [S13] Huang, P.; Ceccagnoli, M.; Forman, C.; Wu, D. J. Appropriability mechanisms and the platform partnership decision: Evidence from enterprise software, *ManagScien*, 2013, 59 (1), 102–121. <https://doi.org/10.1287/mnsc.1120.1618>.
- [S14] Huang, P.; Ceccagnoli, M.; Forman, C.; Wu, D. J. When do ISVs join a platform ecosystem? Evidence from the enterprise software industry. In: International Conference on Information Systems, Phoenix, Arizona, USA, 15–18 December 2009.
- [S15] Wnuk, K.; Runeson, P.; Lantz, M.; Weijden, O. Bridges and barriers to hardware-centric software ecosystem participation—A case study, *Informat and Soft Technol*, 2014, 56 (11), 1493–1507.
- [S16] Sarker, S.; Sarker, S.; Sahaym, A.; Bjorn-Andersen, N. Exploring value cocreation in relationships between an ERP vendor and its partners: A revelatory case study, *MIS Quart*, 2012, 36 (1), 317–338. <https://doi.org/10.2307/41410419>.
- [S17] Huber, T.; Kude, T.; Dibbern, J. Resolving tensions in hub-and-spoke networks of the enterprise application software industry—An exploratory case study. In: Proceedings of

- the Sixteenth Americas Conference on Information Systems, Association for Information Systems, Lima, Peru, 12–15 August 2010.
- [S18] Hyrynsalmi, S.; Suominen, A.; Mäkilä, T.; Järvi, A.; Knuutila, T. Revenue models of application developers in android market ecosystem. In: International Conference of Software Business, Springer, Cambridge, MA, USA, 18–20 June 2012. [https://doi.org/10.1007/978-3-642-30746-1\\_17](https://doi.org/10.1007/978-3-642-30746-1_17).
- [S19] Goldbach, T.; Kemper, V.; Should i stay or should i go? The effects of control mechanisms on app developers' intention to stick with a platform. In: 22st European Conference on Information Systems, Tel Aviv, Israel, 9–11 June 2014.
- [S20] Kude, T.; Dibbern, J. Tight versus loose organizational coupling within inter-firm networks in the enterprise software industry—the perspective of complementors. In: 15th Americas Conference on Information Systems, Association for Information Systems, San Francisco, California, USA, 6–9 August 2009.
- [S21] Basole, R. C.; Park, H. Interfirm collaboration and firm value in software ecosystems: Evidence from cloud computing, *IEEE Transact on EnginManag*, 2018, 66(3), 368–380. <https://doi.org/10.1109/TEM.2018.2855401>.
- [S22] Hilkert, D.; Benlian, A.; Sarstedt, M.; Hess, T. Perceived software platform openness: The scale and its impact on developer satisfaction. In: Thirty Second International Conference on Information Systems, Association for Information Systems, Shanghai, China, 4–7 December 2011.
- [S23] Valença, G.; Alves, C. We need to discuss the relationship: An analysis of facilitators & barriers of software ecosystem partnerships. In: Proceedings of the 19th International Conference on Enterprise Information Systems, Porto, Portugal, 26–29 April 2017. DOI: 10.5220/0006231900170028
- [G1] The Platform Economy. Available online: <https://innovator.news/the-platform-economy-3c09439b56> (accessed on October 2020).
- [G2] Platform Business Model—Definition | What is it? | Explanation. Available online: <https://www.applicoinc.com/blog/what-is-a-platform-business-model/> (accessed on October 2020).
- [G3] The ecosystem platform as business model. Available online: <https://www.ibm.com/blogs/think/be-en/2018/05/07/ecosystem-platform-business-model/> (accessed on October 2020).
- [G4] What is a digital platform? Available online: <https://enterprisersproject.com/article/2018/12/what-digital-platform> (accessed on October 2020).
- [G5] The Complete Guide to the Revolutionary Platform Business Model. Available online: <https://innovationtactics.com/platform-business-model-complete-guide/> (accessed on October 2020).
- [G6] Everything You Need to Know About Platform Product Management. Available online: <https://www.mindtree.com/blog/everything-you-need-know-about-platform-product-management> (accessed on October 2020).
- [G7] Platforms and Ecosystems: Enabling the Digital Economy. Available online: [http://www3.weforum.org/docs/WEF\\_Digital\\_Platforms\\_and\\_Ecosystems\\_2019.pdf](http://www3.weforum.org/docs/WEF_Digital_Platforms_and_Ecosystems_2019.pdf) (accessed on October 2020).
- [G8] Once They Were Companies, Now They Are Platform Businesses. Available online: <https://www.forbes.com/sites/joemckendrick/2019/01/23/once-they-were-companies-now-they-are-platform-businesses/> (accessed on October 2020).
- [G9] The Four Biggest Challenges Digital Platforms Need to Address. Available online: <https://medium.com/mit-initiative-on-the-digital-economy/the-four-biggest-challenges-digital-platforms-need-to-address-a842413ee3c7> (accessed on October 2020).
- [G10] What You Need to Know Before Starting a Platform Business. Available online: <https://sloanreview.mit.edu/article/what-you-need-to-know-before-starting-a-platform-business/> (accessed on October 2020).

- [G11] Platform marketing: how to harness the power of networks & communities. Available online: <https://www.the-gma.com/platform-marketing-networks-communities> (accessed on October 2020).
- [G12] How to Build a Successful Platform Business. Available online: <https://knowledge.insead.edu/blog/insead-blog/how-to-build-a-successful-platform-business-4728> (accessed on October 2020).
- [G13] Products to Platforms: Making the Leap. Available online: <https://hbr.org/2016/04/products-to-platforms-making-the-leap> (accessed on October 2020).
- [G14] Digital Transformation: Platform Business Model of the Digital Economy. Available online: <https://www.bptrends.com/digital-transformation-platform-business-model-of-the-digital-economy/> (accessed on October 2020).
- [G15] Digital Platforms—Building Digital Business Models Step by Step. Available online: <https://www.handelskraft.com/2018/11/digital-platforms-building-digital-business-models-step-by-step-part-2/> (accessed on October 2020).
- [G16] The Eight Key Dimensions of Platform Product Management. Available online: [https://www.accenture.com/\\_acnmedia/pdf-88/accenture-cmt-industry-x0-platform-product-management-pov-october-2018.pdf](https://www.accenture.com/_acnmedia/pdf-88/accenture-cmt-industry-x0-platform-product-management-pov-october-2018.pdf) (accessed on October 2020).
- [G17] The Power of Platform Businesses: How to Enable Value Creation. Available online: <https://www.imanagesystems.com/digital-marketing/power-platforms-enable-value-creation/> (accessed on October 2020).
- [G18] 10 Digital Marketing Strategies for Your Online Marketplace or Platform. Available online: <https://www.cyberclick.es/numericalblogen/10-digital-marketing-strategies-for-your-online-marketplace-or-platform> (accessed on October 2020).
- [G19] Platform Strategy. Available online: <https://www.thoughtworks.com/what-we-do/platform-strategy> (accessed on October 2020).
- [G20] Market size of global platform economy surpasses \$7 trillion mark. Available online: <https://www.consultancy.org/news/104/market-size-of-global-platform-economy-surpasses-7-trillion-mark> (accessed on October 2020).
- [G21] Platform Strategies for Insurtechs and Insurers. Available online: <https://sensedia.com/en/digital-business/insurance-platform-strategies-insurtechs-apis/> (accessed on October 2020).
- [G22] Why every company needs a digital business platform. Available online: <https://www.genpact.com/insight/blog/why-every-company-needs-a-digital-business-platform> (accessed on October 2020).
- [G23] How to Be a Digital Platform Leader. Available online: <https://hbswk.hbs.edu/item/how-to-be-a-business-platform-leader> (accessed on October 2020).

## Appendix B

### Interview Script

**Research objectives:** to obtain their perception, based on their professional experience, about what a company should do to change from an independent software product approach to a software ecosystem.

### Profile of the Participants

- (1) What information should be understood about customers, in addition to their needs (requirements)?
- (2) How to obtain partners with the desired profile?
- (3) How can the company expand the portfolio of partners to enable the expansion of the software ecosystem to different niche markets?

### Platform Quality

- (4) How to enable the integration of partner solutions with the ecosystem software platform?
- (5) What strategies can the company and its partners adopt to evolve the ecosystem software platform?

- (6) What strategies can be adopted to ensure that the company's partners maintain an adequate quality of products?

#### Attraction and Maintenance of Participants

- (7) What strategies can be used to create a number of solutions sufficient to launch the ecosystem platform on the market?
- (8) How should the monetization policy be defined in the software ecosystem?
- (9) What actions can be used to stimulate the attraction of new partners?
- (10) What actions can be used to stimulate the attraction of new customers?
- (11) How to avoid the participation of customers in the company's ecosystem and the ecosystem of competitors simultaneously?

#### Support for the Partner

- (12) In your opinion, what are the appropriate strategies to support partners in obtaining new customers and markets?
- (13) What actions can the company take to help partners develop the necessary skills to provide complementary solutions to the ecosystem platform?
- (14) How to promote effective communication with partners?

### Appendix C

Parts of the Interviews Transcripts Referents to the Quality of the Platform (Questions 4–6 of the interviews script).

The analysis results of these transcripts and the others, through the coding process, are presented in Section 4.7. More specifically, the results regarding Questions 4–6 can be visualized in Figures 8–10, respectively. We highlight that, in these figures, there are results from the coding process made in the complete transcripts, that is, in the parts presented here and in the other transcripts.

#### Interviewee 1

##### Question 4

*"I think that, first, you have to be in an environment that is accessible, be in the cloud, so you don't have too many environment restrictions, have only the security you need, right, but not have a penetration barrier".*

##### Question 5

*"Talk and explain how the change will be. That is, what the change will be, why it needs to exist, and how you are thinking of doing it ... As much as you own the platform, for it to continue evolving, you need partners, and for you to need partners, you need to have communication in order to make engagement remain. So when you make a change like this, you may have a problem with people's engagement".*

##### Question 6

*"That is very serious. That is very serious because, sometimes, because of a partner, you put at risk the entire ecosystem, the image [ecosystem image]. Here again, comes the question of the brand, the image. I think that: from the beginning, you make clear the quality criteria. You do not neglect any quality step; there is pressure for that [neglect quality] when the deadline is approaching".*

#### Interviewee 3

##### Question 4

*"It is like the idea of you developing a platform that is completely extensible and integrable, that is, the company x has a layer of exposure of the entire services of the technological stack... With the different digital channels, for example, like Mobile, you can make a call through API, through microservices, from Mobile, from your... From the framework that you are using (whether IOS, Android, hybrid, CORBA, whatever you want). You can make that call from the mobile directly on the portal, understand?"*

## Question 5

*“I think that too, very important in that process, is the documentation process, which is often neglected, and is being neglected, neglected based on the RUP, in the old structured analysis. So, you need to have documentation, especially of integration and the SDKs (that are involved in allowing that development). You need to have that documentation minimally adjusted so that the other tip [the partner] can interpret and can leave the other side with minimal interaction [interaction with the keystone], right?”*

## Question 6

*“You submit the code. You submit the project, and that project is evaluated according to the good practices defined by the platform [by the keystone] both in terms of performance and code structuring (from the point of view of organization and complexity, understand?)... The most appropriate way is you to implement the revision system”.*

## Interviewee 5

## Question 4

*“Each one [keystone and partners] tests with their testers, each one [keystone and partners] tests independently, and the moment they had this link [moment of integrating solutions] between the two, they would have to do new tests (of that integration). So, I think it would be interesting to validate individually (separately) and then everything integrated”.*

## Question 5

*“If I launch a new version of my product [of the platform], but the product of the partner that is inside [inside the platform] remains outdated, I think I need to at least give for the customer a notion of when it [the complement] will also be updated. So, that planning I think is important”.*

## Question 6

*“If I am going to partner with a company that I know do not have enough testers or do not have test analysts who are capacitated to check the quality and do quality validation too, I think it might not be a good deal to do that [make the partnership]”.*

*“In the contract, you should say: look, need to have a test’s coverage, your product needs to have a 90% test coverage, and then you have to present it [the test coverage] to us”.*

## References

1. Parker, G.G.; Alstyne, M.W.V.; Choudary, S.P. *Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You*, 1st ed.; W. W. Norton & Company: New York, NY, USA; London, UK, 2017.
2. Cusumano, M.A.; Gawer, A.; Yoffie, D.B. *The Business of Platforms: Strategy in the Age of Digital Competition, Innovation and Power*, 1st ed.; HarperCollins Publishers Ltd.: Toronto, ON, Canada, 2019.
3. Manikas, K. Revisiting software ecosystems Research: A longitudinal literature study. *J. Syst. Softw.* **2016**, *117*, 84–103. [[CrossRef](#)]
4. Bosch, J. Software ecosystems: Taking software development beyond the boundaries of the organization. *J. Syst. Softw.* **2012**, *85*, 1453–1454. [[CrossRef](#)]
5. Hanssen, G.K. A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *J. Syst. Softw.* **2012**, *85*, 1455–1466. [[CrossRef](#)]
6. Manikas, K.; Hansen, K.M. Software ecosystems—A systematic literature review. *J. Syst. Softw.* **2013**, *86*, 1294–1306. [[CrossRef](#)]
7. Haile, N.; Altmann, J. Value creation in software service platforms. *Future Gener. Comput. Syst.* **2016**, *55*, 495–509. [[CrossRef](#)]
8. Valença, G.; Alves, C.; Jansen, S. Strategies for managing power relationships in software ecosystems. *J. Syst. Softw.* **2018**, *144*, 478–500. [[CrossRef](#)]
9. Angeren, J.V.; Kabbedijk, J.; Popp, K.M.; Jansen, S. Managing software ecosystems through partnering. In *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*; Edward Elgar Publishing: Cheltenham, UK, 2013; pp. 85–102.
10. Belo, Í.; Alves, C. Partnership Models for Software Ecosystems: A Systematic Mapping Study. In Proceedings of the 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Kallithea, Chalkidiki, Greece, 28–30 August 2019.

11. Saarni, K.; Kauppinen, M. Activities and Challenges in the Planning Phase of a Software Ecosystem. In Proceedings of the Software Business, 10th International Conference on Software Business, Jyväskylä, Finland, 18–20 November 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 71–85.
12. Cusumano, M.A.; Gawer, A.; Yoffie, D.B.; MacDonald, A. How Digital Platforms Have Become Double-Edged Swords. *MIT Sloan Manag. Rev.* **2019**, *60*, 1–7.
13. Bosch, J. From software product lines to software ecosystems. In Proceedings of the 13th International Software Product Line Conference (SPLC'09), Carnegie Mellon University, Pittsburgh, PA, USA, 24–28 August 2009; pp. 111–119.
14. Plakidas, K.; Stevanetic, S.; Schall, D.; Ionescu, T.; Zdun, U. How do software ecosystems evolve? A quantitative assessment of the r ecosystem. In Proceedings of the 20th International Systems and Software Product Line Conference SPLC, Beijing, China, 16–23 September 2016; pp. 89–98.
15. Jansen, S.; Cusumano, M. Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance. In Proceedings of the 4th International Workshop on Software Ecosystems, Boston, MA, USA, 18 June 2012; pp. 41–58.
16. Dos Santos, R.P.; Werner, C. Treating business dimension in software ecosystems. In Proceedings of the International Conference on Management of Emergent Digital EcoSystems, San Francisco, CA, USA, 21–23 November 2011; pp. 197–201.
17. Hagi, A. Strategic Decisions for Multisided Platforms. *Top* **2015**, *10*, 4–13.
18. Dedehayir, O.; Seppänen, M. Birth and Expansion of Innovation Ecosystems: A Case Study of Copper Production. *J. Technol. Manag. Innov.* **2015**, *10*, 145–154. [[CrossRef](#)]
19. Walton, N. *The Internet as a Technology Based Ecosystem: A New Approach to the Analysis of Business, Markets and Industries*, 1st ed.; Palgrave Macmillan: London, UK, 2017; pp. 27–28.
20. Valença, G.; Alves, C.; Jansen, S. A power perspective on software ecosystem partnerships. In Proceedings of the 17th International Conference on Product-Focused Software Process Improvement, Trondheim, Norway, 22–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 69–85.
21. Guggenberger, T.; Möller, F.; Haarhaus, T.; Gür, I. Ecosystem Types in Information Systems. In Proceedings of the 28th European Conference on Information Systems, Marrakesh, Morocco, 15–17 June 2020.
22. Franco-Bedoya, O.; Ameller, D.; Costal, D.; Franch, X. Open source software ecosystems: A Systematic mapping. *Inf. Softw. Technol.* **2017**, *91*, 160–185. [[CrossRef](#)]
23. Wnuk, K.; Manikas, K.; Runeson, P.; Lantz, M.; Weijden, O.; Munir, H. Evaluating the governance model of hardware-dependent software ecosystems—A case study of the Axis ecosystem. In Proceedings of the 5th International Conference of Software Business, Paphos, Cyprus, 16–18 June 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 212–226.
24. Moore, J.F. Predators and prey: A new ecology of competition. *Harv. Bus. Rev.* **1993**, *71*, 75–86.
25. Rong, K.; Shi, Y. *Business Ecosystems Constructs, Configurations, and the Nurturing Process*, 3rd ed.; Palgrave Macmillan: New York, NY, USA, 2015; pp. 137–143.
26. Moore, J.E. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*; Book Review by Herb Rubenstein; Herb Rubenstein: New York, NY, USA, 1996.
27. Cusumano, M.; Gawer, A. The elements of platform leadership. *IEEE Eng. Manag. Rev.* **2003**, *31*, 8. [[CrossRef](#)]
28. Aarnoutse, F.; Renes, C.; Snijders, R.; Jansen, S. The reality of an associate model: Comparing partner activity in the eclipse ecosystem. In Proceedings of the European Conference on Software Architecture Workshops, Vienna, Austria, 1–6 August 2014.
29. Angeren, J.V.; Kabbedijk, J.; Jansen, S.; Popp, K.M. *Partnership Characteristics within Large Software Ecosystems*; Springer: Berlin/Heidelberg, Germany, 2011.
30. Belvner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75. [[CrossRef](#)]
31. Livari, J.; Venable, J. Action research and design science research—Seemingly similar but decisively dissimilar. In Proceedings of the 17th European Conference on Information Systems, Verona, Italy, 8–10 June 2009; pp. 1–13.
32. Wieringa, R.J. *Design Science Methodology for Information Systems and Software Engineering*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 3–11.
33. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [[CrossRef](#)]
34. Kitchenham, B.A.; Budgen, D.; Brereton, P. *Evidence-Based Software Engineering and Systematic Reviews*, 1st ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015.
35. Garousi, V.; Felderer, M.; Mäntylä, M. The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature. In Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, Limerick, Ireland, 1–3 June 2016; pp. 1–6.
36. Mourão, E.; Kalinowski, M.; Murta, L.; Mendes, E.; Wohlin, C. Investigating the use of a hybrid search strategy for systematic reviews. In Proceedings of the 11th International Symposium on Empirical Software Engineering and Measurement, Markham, ON, Canada, 9–10 November 2017; pp. 193–198.
37. Villamizar, H.; Kalinowski, M.; Viana, M.; Fernández, D.M. A systematic mapping study on security in agile requirements engineering. In Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications, Prague, Czech Republic, 29–31 August 2018.
38. Stol, K.J.; Ralph, P.; Fitzgerald, B. Grounded theory in software engineering research: A critical review and guidelines. In Proceedings of the 38th International Conference on Software Engineering (ICSE), Austin, TX, USA, 14–22 May 2016; pp. 120–131.

39. Garousi, V.; Felderer, M.; Hacaloğlu, T. Software test maturity assessment and test process improvement: A multivocal literature review. *Inf. Softw. Technol.* **2017**, *85*, 16–42. [CrossRef]
40. Garousi, V.; Mäntylä, M.V. When and what to automate in software testing? A multi-vocal literature review. *Inf. Softw. Technol.* **2016**, *76*, 92–117. [CrossRef]
41. Islam, C.; Babar, M.A.; Nepal, S. A Multi-Vocal Review of Security Orchestration. *ACM Comput. Surv.* **2019**, *52*, 1–45. [CrossRef]
42. Järvinen, P. Action Research is Similar to Design Science. *Qual. Quant.* **2007**, *41*, 37–54. [CrossRef]
43. Brambilla, M.; Cabot, J.; Wimmer, M. *Model-Driven Software Engineering in Practice*, 2nd ed.; Morgan & Claypool Publishers: San Rafael, CA, USA, 2017.
44. *OMG Meta Object Facility (MOF) Core Specification*; Object Management Group Inc.: Milford, CT, USA, 2019.
45. Sjøberg, D.I.K.; Dybå, T.; Anda, B.C.; Hannay, J.E. Building theories in software engineering. In *Guide to Advanced Empirical Software Engineering*; Springer: London, UK, 2008; pp. 312–336.
46. Farshidi, S.; Jansen, S.; Van Der Werf, J.M. Capturing software architecture knowledge for pattern-driven design. *J. Syst. Softw.* **2020**, *169*, 110714. [CrossRef]
47. Ya’u, B.I.; Nordin, A.; Salleh, N.; Aliyu, I. Requirements Patterns Structure for Specifying and Reusing Software Product Line Requirements. In Proceedings of the International Conference on Information and Communication Technology for the Muslim World, Kuala Lumpur, Malaysia, 23–25 July 2018.
48. Amshoff, B.; Dülme, C.; Echterfeld, J.; Gausemeier, J. Business Model Patterns for Disruptive Technologies. *Int. J. Innov. Manag.* **2015**, *19*, 1–22. [CrossRef]
49. Laue, R.; Kirchner, K. Patterns for Discussing and Modelling Variability in Business Processes. In Proceedings of the 23rd European Conference on Pattern Languages of Programs, Association for Computing Machinery, Irsee, Germany, 4–8 July 2018; pp. 1–10.
50. Seaman, C.B. Qualitative Methods. In *Guide to Advanced Empirical Software Engineering*; Springer: London, UK, 2008; pp. 35–62.
51. Cruzes, D.; Dybå, T. Recommended Steps for Thematic Synthesis in Software Engineering. In Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement, Banff, AB, Canada, 22–23 September 2011; pp. 275–284.
52. [Eclipse.org](https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf). Bylaws of Eclipse Foundation, Inc. Available online: [https://www.eclipse.org/org/documents/eclipse\\_foundation-bylaws.pdf](https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf) (accessed on 10 November 2019).
53. Become an SAP Partner. Available online: <https://www.sap.com/partner/become.html> (accessed on 11 November 2019).
54. Microsoft. Available online: <https://partner.microsoft.com/en-us/membership> (accessed on 15 November 2019).
55. Microsoft—See Which Partner Offer Is Right for You. Available online: <https://partner.microsoft.com/en-us/membership/compare-offers> (accessed on 15 November 2019).
56. Hoffmann, A.; Söllner, M.; Hoffmann, H.; Leimeister, J.M. Towards Trust-Based Software Requirement Patterns. In Proceedings of the Second IEEE International Workshop on Requirements Patterns, Chicago, IL, USA, 24 September 2012.
57. Xuan, X.; Wang, Y.; Li, S. Privacy Requirements Patterns for Mobile Operating Systems. In Proceedings of the IEEE 4th International Workshop on Requirements Patterns, Karlskrona, Sweden, 26–26 August 2014; IEEE: Karlskrona, Sweden, 2014; pp. 9–42.
58. Axelsson, J.; Skoglund, M. Quality assurance in software ecosystems: A systematic literature mapping and research agenda. *J. Syst. Softw.* **2016**, *114*, 69–81. [CrossRef]
59. Jansen, S.; Yang, Z. Source Data for the Focus Area Maturity Model for Software Ecosystem Governance. *Data Brief* **2020**, *31*. [CrossRef]
60. Merriam, S.B. *Qualitative Research: A Guide to Design and Implementation*, 3rd ed.; Jossey-Bass: San Francisco, CA, USA, 2009.