

Management of Partner Ecosystems in the Enterprise Software Industry

Abilio Avila, Orestis Terzidis

¹ Institute of Entrepreneurship, Technology-Management & Innovation
Karlsruhe Institute of Technology
Karlsruhe, Germany
abilio.avila@kit.edu, orestis.terzidis@kit.edu

Abstract. Partner ecosystems are responsible for a significant percentage of the value creation of many companies in the enterprise software industry. Consequently, the focus of competition has moved from the management of internal resources to the management of complementary assets that are beyond the own companies' borders. Nevertheless, many software vendors are still trying to understand the management of partner ecosystems. This paper introduces a framework for the management of partner ecosystems in the enterprise software industry. The framework consists of four management levels: (i) selection of suitable partners, (ii) management of the individual partner relationships, (iii) management of a partner program, and (iv) management of a partner network. However, the focus of this paper is on a level (ii)-(iv). The development of the overall framework is based on the analysis of about 360.000 words of 33 semi-structured interviews with experts from the enterprise software industry.

Keywords: Software Ecosystems, Partner Ecosystems, Business Ecosystems, Enterprise Software Industry, SECO

1 Introduction

The enterprise software industry belongs to the network economy and is shaped by complementary and network effects. Thus, this industry behaves like a massively interconnected network of organizations, technologies, consumers and products. [1].

In the past, companies that commercialized products did not give too much attention to „*innovation coming from the side roads*“ [2]. In the early stages of the software industry, the value proposition for the customers was the result of independent software companies through the development of monolithic software products. [3, 4] The execution focus was on developing customer insight, building core competencies and beating the competition. Thus, companies devoted less attention to external companies that were neither competitors nor customers. [2, 5].

However, in the enterprise software industry, this centralized and vertical perspective has changed significantly. Today's landscape is highly fragmented and specialized software companies have emerged offering complementary services and

products. [3, 4] Management disciplines like customer development and competitive analysis are still vital. However, the management of dependencies to a multitude of external complementary companies is equally important when it come to determining success and failure. [5]

Today, the success of a software company depends not only on its own quality but also on its ability to manage a landscape of multiple partners. Customers do not longer decide for a single software product, but for a software ecosystem, where a software vendor and its partners create value for the customer. [6–8] Consequently, network industries such as the enterprise software industry require a different perspective on management and demand a partner ecosystem perspective.

This paper introduces a framework for the management of partner ecosystems in the enterprise software industry.

2 Related Work

The research field of software ecosystems is relatively young and still in a formative phase. The concept of software ecosystem was arguably coined in 2003 and the first papers have appeared in 2007 [9, 10]. Since then a broad variety of different phenomena and aspects are being explored. Thus, there exist a wide variety of research that studied software ecosystems in different contexts and with different perspectives. For example, studies have been conducted related to the software architecture and technical platform of software ecosystems ([11–17]), measuring the health and performance of software ecosystems [18–22], focusing on the business perspective [3, 23–25] or examining the modeling of software ecosystems ([26–31]).

However, the majority of papers have as contribution knowledge and experience obtained, rules of thumb or interesting observation but they are lacking a systematic approach. This is mainly related to the fact that a significant number of papers is based on single studies that are difficult to generalize. Consequently, the research field is lacking specific theories, methods, and tools. [9, 10]

This paper contributes to closing this gap in two ways. First, this study analyzed 33 semi-structured interviews with 27 different experts from the enterprise software industry (about 1.300 pages). Thus, the results reflect not a single case, but decades of experience of a multitude of different experts. Second, this research results in the development of a systematic framework for the management of partner ecosystems in the enterprise software industry. Herby, it contributes by a systematic approach, which is generalizable within the enterprise software industry. The result is a framework for the management of partner ecosystems in the enterprise software industry that consists of four core management levels.

In a previous study, the researchers concentrated the analysis of the 33 semi-structured interviews on the first of four management levels [8]. In order to offer a comprehensive result, this paper shortly summarizes the results of this previous study (management level 1) and subsequently unfolds in detail the results of the analysis of the other management levels (management level 2 - management level 4).

3 Research Question and Objective

This prospective study was conducted in order to address the following research question: What are the critical success factors (CSF) for the management of a partner ecosystem?

Based on this research question, the purpose of this paper is to introduce a framework for the management of partner ecosystems in the enterprise software industry.

4 Method

With the purpose to develop an understanding of the nature of the enterprise software industry, without getting influenced by existing theories, the researchers decided to use the research approach Grounded Theory (GT). GT is based on the inductive generation of theory, well grounded in empirical data and avoids an intensive literature review before the own theory emerges. [32–34] This section provides a summary of our GT research approach, which is described in more details in our previous study on partner selection in the enterprise software industry (management level 1) [8]. An overview of our grounded theory research approach is shown in Figure 1.

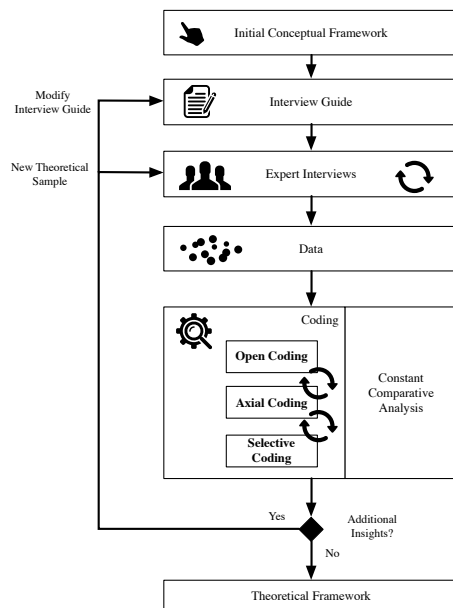


Fig. 1. Structure of the Grounded Theory Study.

5.1 Initial Conceptual Framework and Interview Guide

In order to identify and aggregate an initial cross-case pattern of success, the researchers analyzed 15 secondary case studies. These 15 secondary cases represent the initial theoretical sample (theoretical sample zero) and served the researchers as a starting point for further research activities.

It enabled the researcher to develop a preliminary structure of the research domain based on real business cases and resulted in the design of an initial conceptual framework. This framework built the basis for the interview guide.

Secondary Case Studies

The analysis of secondary case studies can be described as the use of existing case studies in order to address a research question that is in line with but differs from the purpose of the original case study [35]. Case studies are conducted in real-world settings. Consequent, they have a high degree of realism and offer insights in real business situations [36]. Thus, case studies are a suitable source to explore complex situations in the area of the development and commercialization of software products in the enterprise software industry.

In this study, the researchers selected secondary case studies primarily based on three simultaneously applicable main criteria:

- Companies in the enterprise software industry (B2B market).
- Companies that offered complex software products to solve complex business problems for their customers.
- Companies that needed to provide complementary business services in order to offer their customers a satisfactory solution for their business need. [8]

In addition, the completeness and quality of the secondary cases were evaluated in accordance with [35]. An overview of the selected case studies is given in Table 1.

Table 1. Selected Case Studies.

ID	Case Title	Company	Country
01	Scrum, Sprints, Spikes and Poker	Telerik	Bulgaria
02	Beas Systems, Inc. In 2013: Reaching for the Next Level	Beas Systems	USA
03	Precise Software Solutions	Precise Software Solutions	USA
04	PremiumSoft: Managing Creative People	PremiumSoft	China
05	Oracle Corporation	Oracle Corporation	USA
06	Product Development at OPOWER	OPOWER	USA
07	Nuway Software	Nuway Software	USA
08	SAP AG: Orchestrating the Ecosystem	SAP AG	Germany
09	Salesforce: The Evolution of Marketing Systems	Salesforce.com	USA
10	Siebel Systems	Siebel Systems	USA
11	WebSpective Software, Inc.	WebSpective Software, Inc.	USA
12	Customer Value-Based Pricing	Trilogy Corporation	USA
13	Lean at Wipro Technologies	Wipro Technologies	India

Initial Conceptual Framework

As a result of the analysis of these case studies through multiple iterations of data coding based on GT, the researchers developed an initial conceptual framework. This framework (Figure 2) served as the foundation for the design of the interview guide.

Company Related Factors <ul style="list-style-type: none"> ▪ Understand and manage the product value chain ▪ Cultivate distinctive competencies along the product value chain ▪ Determine the scope of the software company ▪ Understand dependencies and co-innovation risk ▪ Complementary value enhance services ▪ Complementary Platform Strategy ▪ Intellectual Property (IP) ▪ Organizational structure, processes and internal communication ▪ Company Culture ▪ Human Resources Management ▪ Finance and Investment ▪ Common company vision and business strategy ▪ Software Product Management ▪ Sales organization and sales process 	Product Related Factors <ul style="list-style-type: none"> ▪ Product Platform Strategy ▪ Product Platform Architecture ▪ Product Technology ▪ Pricing / Revenue Model ▪ Unique Superior Product ▪ Complementarity and Interoperability New Product Development Process <ul style="list-style-type: none"> ▪ Opportunity Identification and Evaluation ▪ Product Development ▪ Product Commercialization 	Partner Network Related Factors <ul style="list-style-type: none"> ▪ Leverage the ecosystem to accelerate innovations, create business values for the customer and scale ▪ Establish and leverage a partner network around the product ▪ Establish a procedure for partner select ▪ Low barriers for partnership ▪ Offer incentives and a clear partner business proposition for partners ▪ Ensure high quality of partner solutions ▪ Education and service offering for partners ▪ Stimulate complementary innovation for the product ▪ Establish strategic alliances ▪ Establish routines for developing and maintaining relationships with partner ▪ Common product innovation roadmap with partners
Market Related Factors <ul style="list-style-type: none"> ▪ Competition ▪ Target Customers ▪ Market Barriers ▪ Market Opportunity ▪ Community Management ▪ Company reputation and credibility 	Environmental Influences <ul style="list-style-type: none"> ▪ Technology Trend ▪ Market Trends ▪ Government and Regulation 	

Fig. 2. Six identified main categories and corresponding sub-categories

Initial Interview Guide

The initial interview guide was designed according to the discovered components of the conceptual framework. It allowed the researchers to start the expert interviews with a set of well-prepared questions.

The initial catalog consisted of 186 modular questions. These questions were structured into six discrete segments that address the subject of interest and were extended by two introductory sections. Thus, the interview guide comprised eight building blocks: Introduction, questions regarding the interview participant, company related factors, partner network related factors, market-related factors, product-related factors, product development process, environmental influences.

5.2 Data Collection

The researchers collected the data through the conduction of 33 semi-structured interviews with 27 different experts from the enterprise software industry (about 2300 minutes). These interviews were transcribed and the resulting textual representation of about 360.000 words was subsequently coded based on the coding approach of GT.

The participants of the interviews were selected according to the emerging theory and driven by data. All the participants were experts with significant experience in the

enterprise software. The researchers assembled diverse clusters of experts (theoretical samples) from different business areas (e.g. sales, partner management, marketing, consulting). This was in order to gain a comprehensive understanding of the domain and maximize the possibility to identify novel categories and insights. Furthermore, this enabled the research to develop categories of overall importance for the software enterprise industry across individual business disciplines.

5.3 Data Analysis

The categorization and conceptualization of the data through a structured coding approach represent the keystone of GT.

“Coding gets the analyst off the empirical level by fracturing the data, then conceptually grouping it into codes that then become the theory that explains what is happening in the data.” [37]

Based on Strauss’s GT version of GT [38] we used the following coding approach composed of three main elements: open coding, axial coding, and selective coding.

Open Coding

The objective of open coding is to identify categories in data and developed them in terms of their related characteristics. The researchers broken down the data into discrete components, examined them, and compared them for similarities and differences. For this purpose, the researchers identified phenomena in the data and label them with codes in order to develop categories. The detailed analysis of data allowed the differentiation among the categories. Furthermore, categories that were similar in nature, related to each other, or represent characteristics were grouped into more abstract categories resulting in a structure of categories and subcategories. [38]

Axial Coding

Axial coding contributed to the development of the theory by reassembling the data that were fractured through open coding. The process relates the categories to their subcategories on a conceptual level. Thus, the researchers reviewed the data on a conceptual level for evidence how the categories were related to its subcategories and specified the nature of their relationships. This allowed the analysts to form more exact and comprehensive descriptions of the identified phenomena and to developed a dense conceptual structure of the relationships around the axis of the respective categories. [38]

Selective Coding

Selective coding integrates and refines the identified categories with the purpose to form a larger theoretical scheme that results in a solid theory.

The integration begins with the identification of a central category that emerged from the research. This central category represents the main topic of the research and connects all the other categories with the purpose to obtain an integrated theory.

The researchers specified the relationships between the major categories obtained through open and selective coding (selection of suitable partners, management of the

individual partner relationships, management of a partner program, management of a partner network) and the core category (management of partner ecosystems in the enterprise software industry). Through this process, the researchers decided which of the initial categories contribute to the core theory, and refined and rearranged the relevant categories. The major categories were integrated and refined to form a larger theoretical scheme. [38]

The attention of the analytic procedure moved from open coding to axial coding and ultimately to selective coding. However, the coding procedure is not a linear process. It is an iterative and incremental approach based on constant comparison and close examination of the data, where the three coding modes were applied iteratively and often simultaneously.

6 Results: The Four Management Levels

We identified four core categories that represent the central management levels that a software vendor needs to address in order to manage a partner ecosystem: (i) selection of suitable partners, (ii) management of the individual partner relationships, (iii) management of a partner program, and (iv) management of a partner network. These are the main areas for the management of partner ecosystems. Each of these core categories consists of further subcategories.

6.1 Level 1: Selection of Suitable Partners

The selection of appropriate partners represents the fundament for every successful partnership. The researchers identified in a previous analysis of the collected data eight distinct selection categories: fundamental fit, culture fit, organization fit, strategy fit, commitment, ecosystem fit, complementarity, and market access [8]. The following paragraph represents a summary of the results of this previous study.

Fundamental fit represents prerequisites related to fundamental characteristics of a potential partner, essential to carry out its role within a partnership in a stable way and comprise industry expertise, reputation, financial stability, and company size. The partner's company culture should be compatible with the software vendor's culture (*culture fit*). This culture fit includes its values, behavioral principles, business practices, service standards and its overall business philosophy. A mismatch of cultures is a potential conflict area for a partnership. *Organizational fit* reflects the availability of an appropriate organizational structure that allows the partner to address the necessary disciplines of the business such as sales, professional services, consulting or customer support. *Strategy fit* refers in the present context to the degree to which the partner has complementary business goals and a compatible vision regarding the software vendor's core product. *Commitment* refers to the assurance of resources and managerial dedication of a partner to complement the software vendor's core product as a building block of the partner's own business. *Ecosystem fit*

addresses the evaluation of the partner's fit according to the partner portfolio of the software vendor. The position of a potential partner in the partner ecosystem has to be evaluated. *Complementarity* refers to the partner's ability to offer complementary business services, products or components on top of the software vendor's own core software product. *Market access* refers to the partner's potential to enter a market with the software vendor's product. This assessment should address the regional presence of a partner, its existing customer bases and experience in the target market and how these attributes may contribute to benefit the software vendor's market access and the competitiveness of joint offerings. [8]

6.2 Level 2: Manage the Individual Partner Relationships

Five management areas have been identified, necessary to systematically manage the individual relationship that a software vendor has with each of his partners: design, enablement, ramp up, operation, and revision.

Design

This management area is crucial for the development of a mutual understanding of the partnership. It defines the nature of the relationship. The main results of this stage should be a clear understanding of the partnership and its objectives, the commitment of the partner and the assets the partner will contribute to the partnership. Thus, the key deliverables are the definition of the operational scope of the partnership, a set of defined business objectives for the partner, and a partner specific business plan. This comprises also a defined development path for the partner including specific activities such as the participation in trainings or the realization of marketing activities. Furthermore, it is important to define a clear set of criteria for evaluating the achievement of the partner's objectives.

Enablement

The software vendor needs to ensure that the partner develops the skill-base necessary to offer complementary services, components, and/or products on top of the software vendor's product. These skills represent the keystone for the partner to build a complementary business. Depending on the determined scope of the partnership, the partner has to cover different aspects of the software vendor's product value chain. This means, that a partner may fully understand the product and its functionalities, the technical foundation, how to market and sell the product, how to customize or extend the product, how to implement the product within the customer environment or how to manage software projects. Thus, the software vendor has to train the partner in the corresponding disciplines. Consequently, the enablement can be differentiated into product enablement, sales enablement, and implementation enablement. The focus of the enablement depends on the defined operational scope of the partnership. For instance, implementation enablement includes the training of best practice project management specific for the implementation of the software product, training of activities necessary to integrate the product within the customer environment, as well

as trainings that address requirements management to identify and capture the customer needs, or how to plan and perform trainings for the end-users of the product.

Ramp up

The first steps in the partnership are crucial and demand intensive attention in order to ramp up the partnership successfully. A key aspect of the initial stage of the partnership implementation is to provide the necessary conditions and assistance to enable the partner to reach a quick win. This is important to develop early confidence in the partnership and to keep the partner motivated. This can be addressed through intensive assistance and close collaboration for the first customer project(s). Joint projects and sales activities may lead to a shared understanding of the business, effective knowledge transfer, reduction of uncertainties and the development of confidence.

Operation

From the moment the partnership has been established a continuous management of the relationship is required. The formation of the partnership is just the beginning. Different aspects such as partner assistance and communication have to be taken into consideration by the software vendor.

Partner Assistance

In order to be able to conduct its business, the partner will need access to assistance on a regular basis. For instance, a partner may need access to the software vendor's product support to obtain technical information or to professional services such as consulting to get assistance for the implementation of a customer project. Furthermore, a partner may need support through joint sales or marketing activities. Thus, the software vendor needs to provide the partner an easy access to support personnel and documentation.

Communication

The establishment of strong communication linkages between the software vendor and the partner are required in order to develop a beneficial relationship. Communication on a regular basis keeps the parties aligned and contributes to building trust. Thus, it represents a vital management instrument and builds the fundament for a well-operating partnership.

Regular communication allows the partner to keep up to date regarding new developments and changes and provides insights to the partner about the software vendor and its business. The establishment of an effective communication is the basis for knowledge and experience transfer. It fosters collaboration and contributes to building a mutual understanding of the business and the expectations of the parties.

Furthermore, an established communication path to partners allows the software vendor to leverage the partner's market insights. Partners that offer complementary activities for customers such as consulting, product implementation and sales efforts have continuous access to the target market. Thus, they are an ideal source to gain valuable cross-customer information regarding current and future customer needs. For

this purpose, software vendors may establish formal systems, e.g. web-portals that allow partners to report and rank customer needs and partner requests.

Evaluation

A continuous performance measurement is a vital management task for partnerships. The maintenance and evolvement of a partnership represents a significant investment for a software vendor. Thus, the software vendor has to regularly evaluate if the partnership is still beneficial. The degree of accomplishment of annually agreed objectives as well as performance metrics is the base for the evaluation of the partner performance. The most important metric is the generated revenue by the partner. However, a comprehensive evaluation should not be reduced on revenue, but includes multiple aspects. A proper evaluation of the partner performance may include the assessment of the engagement level of the partner (e.g. sales activities or event participation), customer satisfaction, service quality, lead conversion rate, continuity, sustainability of the partner activities, new customer acquisition, and the participation in the software vendor's trainings. A systematic evaluation of the partner's performance, based on a well-defined set of metrics, provides the software vendor with the required information to assess if the partnership generates the expected results. Furthermore, it allows the software vendor, if necessary, to take appropriate measures to improve the partnership outcome and to assist the partner with well-aimed activities. However, the obtained insights may also lead to the conclusion to end the partnership.

6.3 Level 3: Manage a Partner Program

In order to streamline and scale partner activities, level 2 needs to be complemented through a standardized partner program. The goal is to manage a multitude of partners simultaneously and reach a consistent level of quality across them. We specified four main company areas that need to be aligned to successfully implement a partner program: structure, culture, strategy and core capabilities.

Structure

According to [39], the organizational structure describes the company's approach of dividing labor into various tasks and achieves coordination among them. The company's organizational structure is perceived and interpreted by employees as a guiding fundament for their behavior. Consequently, the organizational structure of a company influences employee behaviors. For organizational structures, there are various approaches [40]. Organizational structures that are not aligned with the strategy will be counterproductive.

The interview data revealed that the organizational structure needs to provide the appropriate framework required for the collaboration with a multitude of partners. The development of a partner program and the achievement of its objectives require the development of a supporting structure.

The main objective of a partner program is to reach a homogenous quality across the partner portfolio. Thus, a dedicated organizational unit (the partner academy) that

is operationally capable of training and certifying a multitude of partners is a key building block of a partner program.

In addition, to scale the partner assistance, dedicated organizational units and resources need to be assigned. These organizational units include product support, professional services, sales assistance and marketing assistance.

The organizational unit for the governance, management, and orchestration of all activities regarding the partner portfolio within and beyond the company borders is the partner management organization.

Culture

Culture is a complex concept analyzed and described by numerous researchers and authors [41]. For our purpose, we describe culture as the pattern of shared values, beliefs and norms of an organizational unit which shapes the behaviors of its members in order to succeed. Or as expressed by [42] „*the way we do things around here*“ with the purpose of success. Aligned with this perspective, [43] described culture as the essential way of an organizational unit to success. [44]

Culture provides constancy for an organization and works as a guiding system for people`s behavior. It supports people by telling them what kind of activities are within the boundaries and which are out of the boundaries. Over time, culture establishes rules of behavior and communication patterns. In the organization`s context, it defines what effective and ineffective performance means. [43]

The software vendor has to understand the fact that its success depends not solely on his internal execution but depends significantly on the willingness and ability of his partners to succeed as well. This requires company culture that shifts its focus from an internal execution perspective to a comprehensive view of a partner ecosystem. The decision to develop a partner program is the decision to develop an ecosystem focus. „*Choosing the focus on the ecosystem, rather than simply on the immediate environment of innovation, changes everything - from how you prioritize opportunities and threats, to how you think about market timing and positioning, to how you define and measure success. This new paradigm asks innovators to consider the entire ecosystem by broadening their lens to develop a clearer view of their full set of dependencies.*„, [5]

Success in such a context depends significantly on the degree of alignment of the software vendor with a multitude of complementary partners. Thus, the development of a partner program requires a strong collaborative culture, beyond the software vendors own company borders. The software vendor has to foster a company culture that encourages internal and external collaboration. According to [43] (and in line with our findings), synergy represents the core element of such a collaboration culture. The dynamics in a collaborative culture enables people to empower one another and deliver something as result of their cooperation that is more than the sum of the ingredients. Interaction and involvement, as well as harmony and cooperation, are essential elements in this culture. This culture strives for win-win situations. This culture has to be highly adaptive and able to make fast adjustments. The organization evolves and grows through the collective experience and knowledge of people inside and outside the organization. [43].

Strategy

The software vendor has to understand that the development of a partner program and the ecosystem perspective are vital parts of the strategy. The partner channel represents a vehicle to reach competitive advantage and to develop a strategic position in order to achieve above-average performance in the industry. [45, 46] Consequently, aspects of the software vendor's strategy need to be aligned to address the development of the strategic position of an ecosystem leader. The required alignment has an impact mainly on two strategic disciplines: channel strategy and product strategy.

Channel Strategy

The channel strategy of the software vendor must integrate the building of an indirect channel structure through partners that offer complementary services and products along the value chain of the software vendor's core products. This has enormous strategic and operational implications. The software vendor needs to consciously decide which element of its business value chain should be addressed by partners and to which extend, and which elements should be kept in-house. This implies, that the software vendor has to evaluate which of the business elements represents core competencies and are considered to be central to sustain and extend the software vendors market position and thus should be kept within the company borders. The company should cooperate with partners that complement the value chain through complementary services and products and are crucial to complete its value chain. Products, services, and competencies that are vital elements may be built and maintained within the company. Typically, the software vendor retains the sensitive core elements of its business such as the product source code in-house. Furthermore, despite the shift to partner channels, it is still important to cultivate distinctive skills such as product development and consulting within the own company borders. The maintenance and development of distinctive internal capabilities and the ability to absorb new knowledge is important to maintain the market position as an ecosystem hub. In addition, it is crucial to stay in close contact with the market e.g through the direct implementation of customer projects. Without the direct access to the customer, the software vendor is likely to disconnect from the market and its needs. While the objective is to scale through partners, to rely too much on external partner products and services represents a risk and may end up in significant disadvantages. This balance between control and dependency has to be addressed through a well-defined multi-channel strategy. The channel strategy has to strive for a high degree of mutual complementarity between the software vendor and his partners.

Product Strategy

The decision to develop a core product on which partners offer complementary products and services has an important impact on the software vendor's *product strategy*. This impact arises mainly from the mutual dependency of the core product and the complements. The strategy for a core product that builds the platform for further value creation differs from traditional one-product strategy. Software vendors that rely heavily on the cooperation with complementors to scale and address markets,

need to approach the core product not only from the limited perspective of the own company's border but to extend the circle to external partners. Since the partners are an important element for the product launch and diffusion, it is necessary to evaluate if the product is aligned with the knowledge, skills, experience and resources of the company's extension - the partners. Moreover, the company needs to take into account that complementary partners expect benefits from the software vendor's core products. The product needs to offer the partners a solid basis for profitable complementary services and product enhancements. It is unlikely that partners are going to invest in complementary activities for products without sufficient incentives and financial prospects. This has as well technical as commercial implications for a product. Consequently, a company needs not only to identify and address the customer needs, but also to understand and fulfill the needs of their partners. As a result, the product needs to fit specific characteristics to be material for complementary partner business: market-oriented product, unique superior product, effective customizable, modular architecture, open interfaces, allows the development of integrated modules, offers standard connectors to common third-party software products, possesses effective development tools, offers the fundament for complementary partner services and products. These characteristics foster complementary innovation and facilitate the development of complements that increase the value of the platform.

Core Competency

A company's core competency is defined as an area of specialized expertise that is the result of harmonizing complex streams of individual technologies, production skills, and work activities. It arises from the company's ability to combine multiple key capabilities in which the company excels into a set of key areas of specialized expertise. Core competencies share three main characteristics: (1) they provide the capability to access to a variety of potential markets, (2) they represent a vital element to deliver customer value, (3) since they are a complex combination of multiple streams, they are difficult to imitate by competitors. The company's core competencies rely heavily on the ability to establish and synchronize cross-functional relationships within the company and are crucial elements of the company's overall identity. [35, 47–49] The shift toward the development of a partner program leads to the need to evolve and master a new core competency: partner portfolio management.

Partner Portfolio Management

Obviously, the alignment of the partner portfolio with the objectives of the channel strategy is a central management activity. The objective is to develop a balanced partner portfolio. Consequently, the portfolio should consist of partners that provide a high level of complementarity to the core products and competencies of the software vendor. Furthermore, the partners should have limited overlap with the software vendor's core business.

Furthermore, in order to reach a balanced partner portfolio, aspects such as partner market segmentation a balanced number of partners and partner categories, and the development and maintenance of a homogenous partner quality (certified services and products) belong to the component of partner portfolio management.

Partner Portfolio Management also includes the management of conflicts with partners. Conflicts within the partnership can arise in different forms such as overlapping customer segments or competing products. Thus, the reduction of overlaps and conflict management are important parts of the partner portfolio management.

An additional management activity in which a software vendor has to excel, is the orchestration of activities across different organizational units such as product management, product development, sales, and marketing, with the purpose that their joint efforts serve not only the software vendor's direct channel but the entire partner ecosystem.

In addition, partner specific sales and marketing disciplines such as lead management, forecasting, incentive management (partner levels), and channel communication have to be addressed.

6.3 Level 4: Manage a Partner Network

This level includes attributes that are related to the management of business areas beyond the software vendor's own company borders. In order to foster innovation and collaboration among the partners, a software vendor needs to move into domains beyond his direct control (level 1-3) and develop new paradigms of more indirect influence. This level focuses consequently on the means and measures that address the influence area of the software vendor.

The objective of this management level is to set up the necessary conditions that enable and foster interconnections and collaboration among the partners. The software vendor aims to create an environment that facilitates communication, information exchange and the development of trust among the partners. For this purpose, the software vendor has to fulfill the role of a supporting hub for communication, mutual support, exchange of experiences and collaboration between the partners. This can be addressed by providing supporting elements such as partner conferences and events, a partner community platform, a partner portal or a partner board. Furthermore, in case of conflicts between partners, the software vendor may play an important role as mediator and handle the escalation (escalation management). However, the fundament is to provide a harmonized partner portfolio with limited overlaps among the partners.

According to [50] and complementing our data, the advantages of such an integrated network of partners can be categorized in information advantage, cooperation advantage, and power advantage. Information advantage reflects the ability of all partners to share common knowledge. Information that is known by one member of the network spreads rapidly among the other partners. Cooperation advantage results from the ability of the partner network as a whole to ensure proper conduction of the individual partners. This is because, in such an interconnected network, a partner can not misbehave to another partner without affecting the relationship to other partners of the network. Power advantage refers to the software vendor's ability to mobilize collective resources, for example, to respond to common competitors.

To fully unlock the benefits of a partner ecosystem and gain competitive advantage, a software vendor needs to address all management levels of the partner ecosystem.

7 Relevance

The results allow researchers and practitioners to draw from the complex and multi-dimensional activities of partner ecosystem management in the Enterprise Software Industry and concentrate on the essential management areas where high performance must be ensured. Furthermore, the framework supports researchers with a valuable blueprint for further research activities in the area of partner ecosystems. Therefore, this article has both profoundly practical as well as scientific relevance.

References

1. Iansiti, M., Levien, R.: *Keystones and dominators: framing operating and technology strategy in a business ecosystem*. Harvard Business School, Boston (2004)
2. Gawer, A., Cusumano, M.A.: *Platform leadership* (2002)
3. Popp, K., Meyer, R.: *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. BoD–Books on Demand (2010)
4. Buxmann, P., Diefenbach, H., Hess, T.: *The software industry. Economic principles, strategies, perspectives*. Springer, Heidelberg, New York (2013)
5. Adner, R.: *The wide lens*. Portfolio, New York (2012)
6. Jansen, S., Cusumano, M.A.: 1. Defining software ecosystems: a survey of software platforms and business network governance. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, 13 (2013)
7. Torrisi, S.: *Industrial organisation and innovation. An international study of the software industry. New horizons in the economics of innovation*. E. Elgar Pub., Cheltenham, Northampton, MA (1998)
8. Avila Albez, A.: *Management of Partner Ecosystems in the Enterprise Software Industry – The Partner Selection*. G-Forum - 20th Annual Interdisciplinary Conference on Entrepreneurship and Innovation. (2016)
9. Manikas, K., Hansen, K.M.: *Software ecosystems—a systematic literature review*. *Journal of Systems and Software* 86(5), 1294–1306 (2013)
10. Manikas, K.: *Revisiting software ecosystems research: a longitudinal literature study*. *Journal of Systems and Software* 117, 84–103 (2016)
11. Bosch, J. (ed.): *Architecture challenges for software ecosystems*. ACM (2010)
12. Cataldo, M., Herbsleb, J.D. (eds.): *Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration*. ACM (2010)
13. dos Santos, Rodrigo Pereira, Werner, Cláudia Maria Lima (eds.): *Revisiting the concept of components in software engineering from a software ecosystem perspective*. ACM (2010)
14. Kazman, R., Gagliardi, M., Wood, W.: *Scaling up software architecture analysis*. *Journal of Systems and Software* 85(7), 1511–1519 (2012)
15. Lungu, M., Robbes, R., Lanza, M. (eds.): *Recovering inter-project dependencies in software ecosystems*. ACM (2010)
16. Robbes, R., Lungu, M. (eds.): *A study of ripple effects in software ecosystems (nier track)*. ACM (2011)
17. Viljainen, M., Kauppinen, M. (eds.): *Software ecosystems: A set of management practices for platform integrators in the telecom industry*. Springer (2011)
18. Fotrousi, F., Fricker, S.A., Fiedler, M., Le-Gall, F. (eds.): *Kpis for software ecosystems: A systematic mapping study*. Springer (2014)
19. Hartigh, E.d., Tol, M., Visscher, W. (eds.): *The health measurement of a business ecosystem* (2006)
20. Jansen, S.: *Measuring the health of open source software ecosystems: Beyond the scope of project health*. *Information and Software Technology* 56(11), 1508–1519 (2014)
21. Manikas, K., Hansen, K.M. (eds.): *Reviewing the Health of Software Ecosystems-A Conceptual Framework Proposal* (2013)
22. van den Berk, Ivo, Jansen, S., Luinburg, L. (eds.): *Software ecosystems: a software ecosystem strategy assessment model*. ACM (2010)
23. Burkard, C., Widjaja, T., Buxmann, P.: *Software Ecosystems* (2012)
24. Popp, K.M. (ed.): *Hybrid Revenue Models of Software Companies and their Relationship to Hybrid Business Models* (2011)

25. Weiblen, T., Giessmann, A., Bonakdar, A., Eisert, U.: Leveraging the software ecosystem: Towards a business model framework for marketplaces (2012)
26. Boucharas, V., Jansen, S., Brinkkemper, S. (eds.): Formalizing software ecosystem modeling. ACM (2009)
27. Handoyo, E.: Software Ecosystem Modeling. *Software Business*, 227
28. Handoyo, E., Jansen, S., Brinkkemper, S.: Software Ecosystem Roles Classification. *Software Business*, 212
29. Handoyo, E., Jansen, S., Brinkkemper, S. (eds.): Software ecosystem modeling: the value chains. ACM (2013)
30. Handoyo, E., Jansen, S., Brinkkemper, S. (eds.): Software ecosystem roles classification. Springer (2013)
31. Pettersson, O., Svensson, M., Gil, D., Andersson, J., Milrad, M. (eds.): On the role of software process modeling in software ecosystem design. ACM (2010)
32. Glaser, B.G., Strauss, A.L.: The discovery of grounded theory: Strategies for qualitative research. Transaction Publishers (2009)
33. Strauss, A.L., Corbin, J.M.: Basics of qualitative research. Sage (2008)
34. Urquhart, C.: Grounded theory for qualitative research: A practical guide. Sage (2013)
35. Hinds, P.S., Vogel, R.J., Clarke-Steffen, L.: The possibilities and pitfalls of doing a secondary analysis of a qualitative data set. *Qualitative Health Research* 7(3), 408–424 (1997)
36. Runeson, P., Host, M., Rainer, A., Regnell, B.: Case study research in software engineering: Guidelines and examples. Wiley.com (2012)
37. Glaser, B.G., Holton, J. (eds.): Remodeling grounded theory (2004)
38. Strauss, A., Corbin, J.: Basics of qualitative research, vol. 15. Newbury Park, CA: Sage (1998)
39. Anderson, D.J.: Kanban. Successful evolutionary change in your software business. Blue Hole Press, Sequim, Wash (2010)
40. Kieser, A., Ebers, M.: Organisationstheorien. W. Kohlhammer Verlag (2006)
41. Campbell-Kelly, M.: Development and structure of the international software industry, 1950-1990. *Business and Economic History* 24(2) (1995)
42. Tukey, J.W.: The teaching of concrete mathematics. *The American Mathematical Monthly* 65(1), 1–9 (1958)
43. Messerschmitt, D.G., Szyperski, C.: Industrial and Economic Properties of Software: technology, processes and value
44. Tim McLaren, M.B., Buijs, P.: A DESIGN SCIENCE APPROACH FOR DEVELOPING INFORMATION SYSTEMS RESEARCH INSTRUMENTS
45. Ahn, J.-H., Lee, D.-J., Lee, S.-Y.: Balancing Business Performance and Knowledge Performance of New Product Development: Lessons from ITS Industry. *Long Range Planning* 39(5), 525–542 (2006). doi: 10.1016/j.lrp.2006.08.001
46. Cooper, R.G.: Winning at new products. Basic Books (2011)
47. Wallin, C.: A process approach for senior management involvement in software product development. Mälardalen University (2003)
48. Bonner, J.M., Ruekert, R.W., Walker, O.C.: Upper management control of new product development projects and project performance. *Journal of Product Innovation Management* 19(3), 233–245 (2002)
49. Jeffrey Thieme, R., Michael Song, X., Shin, G.: Project management characteristics and new product survival. *Journal of Product Innovation Management* 20(2), 104–119 (2003)
50. Greve, H., Rowley, T., Shipilov, A.: Network advantage: how to unlock value from your alliances and partnerships. John Wiley & Sons (2013)