systems

MDPI

Article

Open-Source Collaboration for Industrial Software Innovation Catch-Up: A Digital–Real Integration Approach

Xiaohong Chen [1], Qigang Zhu [2] and Yuntao Long [3,4,*]

1 School of Humanities and Social Science, University of Science and Technology Beijing, Beijing 100083, China; chenxiaohong@ustb.edu.cn
2 Global Open Source Collaboration Research Center, University of International Business and Economics, Beijing 100105, China; zhuqigang@nsoss.cc
3 Institutes of Science and Development, Chinese Academy of Sciences, Beijing 100190, China
4 School of Public Policy and Management, University of Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: yuntaol@casisd.cn

Abstract

In the era of digital–real integration, open-source collaboration has become a strategic pathway for accelerating the innovation catch-up of China's industrial software. This study employs an exploratory multi-case design, focusing on the China Automotive Operating System open-source project and the FastCAE open-source domestic CAE software integrated development platform to examine how open-source strategies shape collaborative mechanisms and innovation outcomes. The analysis reveals that firms adopt both formal (behavioral and outcome coordination) and informal (relationship and empowerment coordination) strategies, fostering high-level complementary collaboration in data, technology, institution, and human resources. These mechanisms significantly enhance R&D efficiency and quality, drive technological innovation, and create new market innovation, thereby improving collaborative performance. The study contributes to theory by linking open-source-driven digital–real integration with industrial software innovation catch-up and offers practical governance recommendations for strengthening China's industrial software autonomy and ecosystem sustainability.

Keywords: open-source collaboration; industrial software; digital–real integration; innovation catch-up

check for updates

Academic Editor: Mitsuru Kodama

Received: 9 July 2025
Revised: 8 August 2025
Accepted: 20 August 2025
Published: 24 August 2025

Citation: Chen, X.; Zhu, Q.; Long, Y. Open-Source Collaboration for Industrial Software Innovation Catch-Up: A Digital–Real Integration Approach. *Systems* **2025**, *13*, 733. https://doi.org/10.3390/systems13090733

1. Introduction

"Digital–real integration" is a major driving force for developing new-quality productivity and promoting high-quality economic development. It can be viewed as the pervasive introduction of digital technologies into every link and life-cycle stage of the real-economy value chain, with the aim of building a high-quality industrial ecosystem that revolves around those technologies (Huang & Wang, 2025) [1]. The Report to the 20th National Congress of the Communist Party of China emphasizes "accelerating the development of the digital economy, promoting the deep integration of the digital economy and the real economy, and creating globally competitive digital industrial clusters" [2]. The "digital economy" mainly refers to the segment of economic output generated entirely or predominantly by digital technologies, underpinned by a business model focused on digital goods or services (Bukht & Heeks, 2017) [3]. In contrast, broadly interpreted, the "real economy" encompasses the production, distribution, and sale of goods and services, and a more restricted view confines the real economy to tangible goods alone, covering industries such as manufacturing, commerce, agriculture, shipping, retail, and construction (Pirounakis, 1997) [4]. On the one hand, the integration of the real economy and the digital economy ("digital–real integration") helps traditional enterprises transform and upgrade, driving iterative innovation in industrial digitalization. On the other hand, viewing data assets as a core element drives the optimization of data value and fosters the vigorous development of digital industrialization.

Open source, abbreviated from "open-source code", primarily refers to code sharing and a collaborative communication model for building virtual systems. Open-source collaboration refers to a mode of cooperative software development among multiple actors, including developers, companies, communities, or foundations (Baldwin & Von Hippel, 2011) [5]. Open-source collaboration promotes the digital transformation which is the process of harnessing digital technologies—such as analytics, cloud computing, the Internet of Things, mobile platforms, and social media—to drive substantial change aimed at improving customer experience and innovating business models (Fitzgerald et al., 2014) [6]. Additionally, the technological sharing and cross-domain collaboration driven by open-source efforts further enrich the innovation ecosystem, enhancing the overall innovation capacity of society and providing robust support for the construction of next-generation digital infrastructure.

Using open-source software (OSS) technology R&D communities (hereafter referred to as "OSS communities" or "open-source communities") as an example, this study explores the theoretical mechanisms of the complementary collaboration that underpins "digital–real integration" between enterprises and these virtual communities. The reason for choosing OSS communities as the research object lies, first and foremost, in their efficacy and strategic importance for resolving "choke-point" problems and achieving innovation catch-up. At present, China is undergoing rapid digital transformation and high-speed development of the digital economy. While technological breakthroughs continue to emerge, foundational R&D in various industries faces tremendous pressures and challenges. With Huawei's 5G chips under blockade, and institutions such as the Harbin Institute of Technology and Harbin Engineering University included on the U.S. Entity List, frequent instances of R&D "choke points" highlight the importance of strengthening foundational research and reinforcing industrial-chain and supply-chain resilience; this is key to controlling core technologies in our own hands. OSS is a critical infrastructure element on a global scale; it forms the core components of fundamental "choke-point" software such as operating systems and databases, and it is equally vital for competitiveness in key areas such as chips [7]. Furthermore, new infrastructure areas including 5G, artificial intelligence, blockchain, big data, and industrial internet also broadly adopt open-source models for development and sharing. Therefore, using open-source collaboration as the primary form and OSS communities as a specific collaboration platform is a crucial strategic proposition for constructing a Digital China and taking the initiative in economic competition.

Second, the complementary collaboration between enterprises and virtual OSS communities is both widespread and of significant importance. Globally, 97% of software developers and 99% of enterprises use open-source software, and more than 70% of newly initiated software projects adopt open-source models. To further promote enterprise–community collaboration, China's 14th Five-Year Plan and its 2035 Visionary Goals outline "support for digital technology open-source communities and other innovation alliances, the improvement of open-source intellectual property and legal systems, and encouragement for enterprises to open software source code, hardware designs, and application services." Simultaneously, an increasing number of enterprises encourage employees to participate in open-source communities, jointly pursuing innovation and value co-creation. For instance, Red Hat assigns full-time employees to the GNOME open-source community [8];

most platform maintainers for the Eclipse open-source community are IBM employees; and Huawei, Tencent, ByteDance, and others have each established their own open-source divisions to engage in OSS community development at the corporate level. Some recent examples include Red Hat collaborating with the open-source community on the Fedora project to advance the Linux ecosystem and enterprise-level operating systems; and OpenAI's GPT (Generative Pre-trained Transformer) models, which are large-scale language models based on the Transformer architecture for natural language processing. OpenAI open-sourced the GPT-1 and GPT-2 codes [9], making them freely available for use, improvement, and application by community developers. These various cases indicate that collaboration between physical enterprises and virtual open-source communities has become commonplace.

Industrial software refers to software that is mainly utilized or specifically developed for the industrial sector, with the purpose of improving industrial enterprises' research and development, manufacturing processes, operational management, and equipment performance (Zhou et al., 2023) [10]. Common examples include R&D and design-oriented industrial software such as Computer-Aided Design (CAD), Computer-Aided Engineering (CAE), and Product Lifecycle Management (PLM) systems. The existing challenges in the R&D and application of domestic industrial software mainly arise from the fact that, compared with leading international industrial software companies, domestic counterparts possess relatively weak technological R&D capabilities, face prominent bottlenecks in critical core technologies, and lack continuous refinement through real-world industrial application scenarios. In addition, the accumulation of application data, industrial know-how, and craftsmanship remains limited, resulting in generally low product maturity. At the same time, the industrial ecosystem for domestic industrial software has yet to be established, with insufficient capabilities for software–hardware integration. Consequently, the question of how to facilitate the innovation catch-up of domestic industrial software has become a focal concern for academia, industry, and government policymakers.

Existing research suggests that open-source collaboration plays a critical role in fostering technological innovation and, according to the resource-based view, can enable enterprises to gain competitive advantages (Barney & Clark, 2007) [11]. Based on transaction cost theory, it helps to obtain higher-quality code at a faster pace, thereby improving software development quality (Haefliger et al., 2008) [12]. Based on institutional theory, an increasing number of companies choose to collaborate with open-source communities by assigning internal members to participate in community contributions, thereby gaining legitimacy for their engagement in open source and enhancing the value of their collaboration (Dahlander & Wallin, 2006) [13]. Open-source collaboration can serve as a key driver for promoting the innovation catch-up of domestic industrial software. However, existing studies still have certain limitations, as they do not explain how open-source collaboration facilitates industrial software innovation catch-up. In particular, in the current context of digital–real integration, there remains substantial research space, with no exploratory studies analyzing the underlying process mechanisms.

Based on this, this study proposes the following core research question: from the perspective of digital–real integration, through what mechanisms does open-source collaboration influence industrial software innovation catch-up? From a theoretical perspective, this work enriches the foundational theory of innovation-focused collaboration between enterprises and virtual communities in the context of digital–real integration, offering deeper insights into the underlying mechanisms and operational patterns of multi-actor complementary cooperation. From a practical perspective, it clarifies the institutional logic underpinning the offshoots of digital–real integration, thereby providing new theoretical resources and analytical angles for advancing industrial digitalization, digital industrialization, and the synergy between them. It further provides an analytical lens for understanding how open-source strategies can enable domestic industrial software to achieve innovation catch-up.

2. Literature Review

2.1. Connotations of Complementary Collaboration

Complementary collaboration, derived from Teece's (1986) concept of complementary assets, emphasizes how organizations acquire external complementary resources beyond their core assets to gain competitive advantages as efficiently as possible [14]. Traditional complementary collaboration has typically involved shared physical resources and capabilities among physical enterprises. For example, Monteiro et al. (2016) studied the complementary collaboration among Samsung Electronics in Korea, a graduate program in electrical engineering, and an electronics information research center. They found that Samsung Electronics gained human resources through collaboration, significantly improving digital television production and processes; graduate students received scholarships; and the research center and professors obtained funding to build new labs, classrooms, and facilities—all showing substantial results from their collaboration [15].

In the era of digital transformation and artificial intelligence, complementary collaboration emphasizes sharing intangible assets such as data, technology, and knowledge. For instance, Microsoft's acquisition of GitHub provides a prototypical example of enterprise–virtual OSS community complementary collaboration. GitHub is the world's largest open-source software development platform, boasting a vast developer community, whereas Microsoft is a leading global software company with formidable development and market resources. By acquiring GitHub in 2018, Microsoft not only gained deeper engagement with and support for the open-source community but also accessed an enormous developer ecosystem endowed with innovation capabilities. This facilitated rapid development in Microsoft's cloud computing and development tools business. Meanwhile, GitHub also benefited from Microsoft's resources and technical backing, expanding its influence and user base. This complementary partnership capitalized on each party's advantages, yielding a synergistic effect of more than the sum of its parts [16].

2.2. Reasons for Complementary Collaboration Between Enterprises and OSS Communities

An open-source community refers to a platform composed of members sharing common interests who voluntarily participate in free exchange and learning within a digital platform [17]. Internationally, well-known communities include GitHub and HuggingFace, while, in China, there are platforms such as Gitee, "Open Source China," and "Open Source Society"etc. Enterprises actively collaborate with OSS communities for the following key reasons: first, reducing transaction costs and improving R&D efficiency. Transaction cost theory posits that the value of inter-organizational collaboration lies in obtaining competitive advantages at relatively low costs [18]. Transaction costs include the time cost of searching for information as well as learning costs. On the one hand, when enterprise members join an OSS community, they can quickly access needed information, code, knowledge, or solutions, largely avoiding "reinventing the wheel" [19]. On the other hand, enterprise members can also subscribe to real-time updates on project dynamics, business models, and future directions—sometimes even obtaining insider information—thus speeding up their learning curves and skill enhancement [20].

The second reason involves obtaining complementary assets to achieve business goals. Chen et al. (2021) analyzed OSS community governance mechanisms in terms of organizational structure, leadership, and intellectual property management, examining the motivations behind enterprise participation in OSS communities [21]. Drawing on motivation theory, an enterprise can glean feedback and contributions from community developers, embedding itself within collaborative projects to gain permission to download non-copyrighted works or use software products with proper licenses. Additionally, an enterprise can expand its product pipeline, attracting more community developers to its projects, enhancing product competitiveness, and increasing market share.

The third reason pertains to gaining legitimacy and enhancing corporate image and brand recognition. Using institutional theory, Wei and Chen (2021) depicted the binary isomorphism of enterprises transitioning into OSS communities from the perspectives of governance mechanisms and business models, finding that cross-organizational isomorphism bolsters enterprise–community coupling through a mediating effect of cognitive legitimacy [22]. Drawing on institutional theory, firms can only gain community recognition and attain legitimate participation rights by contributing to open-source communities. Although open-source community source code is freely available and documentation is publicly accessible, in accordance with the principle that greater contributions yield greater returns, enterprises seeking to derive continuous value from open-source communities must consistently contribute in order to establish legitimate participation. Furthermore, a growing number of firms aspire to collaborate with these communities to demonstrate their commitment to open-source culture and values, thereby enhancing their corporate brand reputation. Consequently, an increasing number of enterprises have adopted collaboration with OSS communities as part of their corporate strategy [23].

2.3. Characteristics of Complementary Collaboration Between Enterprises and OSS Communities

The development of the digital economy has substantially reshaped production relations in the domain of collaborative governance. Compared with complementary collaboration among traditional physical enterprises, collaboration under digital–real integration diverges in three main aspects—namely, the degree of collaboration constraints, value-chain relationships, and patterns of dynamic evolution (Table 1).

Table 1. Characteristics of complementary collaboration between enterprises and virtual open-source communities.

Comparison Category	Characteristics of Complementary Cooperation among Physical Enterprises	Characteristics of Complementary Cooperation Between Physical Enterprises and Virtual Open-Source Communities
Degree of Cooperative Constraint	• Governed by legally binding contractual agreements. • Partners who withdraw arbitrarily are liable for legal consequences.	• Not enforced by legally binding contracts. • Participation in the community is voluntary, and withdrawal does not incur legal consequences.
Value Chain Relationships	• Clear delineation of upstream and downstream value chain relationships. • A defined transactional relationship exists between producers and consumers.	• The traditional dichotomy between enterprises and communities becomes increasingly blurred. • The roles of producers and consumers converge.
Dynamic Evolution Characteristics	• Exhibits relatively lower levels of flexibility and complexity.	• Characterized by a phased development process, with greater flexibility and complexity.

First, there are differences in collaboration constraints. Collaboration between enterprises and OSS communities does not exhibit the same level of enforceability as that between physical enterprises [24]. Although an enterprise and an OSS community may set certain rules and regulations for joint R&D projects, these arrangements differ markedly from legally binding agreements among physical enterprises in terms of the binding strength and the consequences of voluntary exit. This divergence stems from the fact that participation within OSS communities is largely voluntary [25] and, owing to the open and flexible nature of OSS communities, developers can freely enter or leave both the community and specific projects. Consequently, the collaboration environment is more relaxed and flexible. If a participant withdraws from an OSS community project, they generally do not face severe "penalties" or liabilities comparable to those in inter-enterprise contracts.

Second, there are differences in value-chain relationships. Collaboration between enterprises and OSS communities involves a distinctive merging of producers and consumers. According to Porter's value-chain theory, a firm is connected with its suppliers and buyers—parties positioned up- and downstream within the industry chain [26]—and it realizes value enhancement by transacting with its partners (e.g., suppliers or substitute products) [27]. Nonetheless, traditional value-chain analysis does not fully apply to collaboration between enterprises and OSS communities. Conventionally, complementary assets are regarded as part of a company's asset portfolio with a clearly defined status in the value chain; however, collaboration with OSS communities breaks this traditional value-chain structure [28]. This is because, in an OSS project involving both volunteer community developers and enterprise employees, any individual can play the role of producer by creating value or act as a consumer who participates directly in value creation. This scenario entails a merger of producer and consumer roles, blurring the traditional two-way distinction between enterprises and OSS communities [29].

Third, there are differences in dynamic evolution. At different stages, the collaboration relationships between enterprises and OSS communities exhibit distinct characteristics. Germonpre et al. propose that, once an enterprise joins an OSS community, its collaborative relationship evolves through three phases, based on the contextual features of the OSS community and the enterprise's open-source collaboration strategy: communal markets, interdependent ideologies, and distributed relationships. In the first phase, the enterprise primarily aims to enhance or promote its commercial closed-source software by leveraging OSS community resources to consolidate and expand its commercial model [30]. During this stage, the firm's interaction with the OSS community is largely market-driven, seeking additional market opportunities by tapping into community collaboration. In the second phase, to implement a coordinated strategy with the OSS community, the enterprise gradually modifies its behaviors and institutional culture to align with community culture, ensuring legitimacy and continuous participation in the OSS community [31]. By the third phase, as the enterprise–community collaboration stabilizes, both sides establish relatively fixed paradigms in community governance, resource sharing, and collaborative mechanisms. The collaborative relationship then tends toward long-term stability. By contrast, the flexibility and complexity of dynamic evolution among physical enterprises are not on par with those seen in enterprise–OSS community collaborations under digital–real integration.

2.4. Theoretical Proposition Based on Complementary Collaboration in Industrial Software Innovation Catch-Up Under the Digital–Real Integration Approach

This study is positioned within the paradigm of exploratory qualitative research, grounded in open innovation theory, the resource-based view, and digital–real integration theory; it seeks to explain how open-source collaboration influences industrial software innovation catch-up through complementary collaboration mechanisms. Existing research indicates that strategy not only establishes the course of action and decision-making logic for an organization but also guides members' attention allocation and behavioral patterns. These behavioral patterns, in turn, influence performance outcomes by affecting resource integration, collaboration efficiency, and innovation capability (Ocasio, 1997) [32]. In the context of digital–real integration, the strategic formulation of physical enterprises influences their complementary collaboration behaviors with virtual communities, which in turn affects collaborative performance, thereby exerting a significant impact on the innovation catch-up of the industrial software sector. On this basis, this study makes the theoretical proposition that an enterprise's open-source collaboration strategy shapes complementary collaboration behaviors, which subsequently influence collaborative performance.

3. Research Design

3.1. Research Method

This research adopts an interpretivist paradigm with an exploratory, multiple-case study design. The primary reason for employing an exploratory case study method lies in its suitability for research questions related to "how," enabling the investigation of process-oriented and mechanism-related issues through case analysis [33]. It is well-suited for this study, which examines how open-source collaboration promotes industrial software innovation catch-up from the perspective of digital–real integration. Furthermore, compared with a single-case approach, a multiple-case study enables the observation of the impact pathways of different open-source collaboration strategies through theoretical replication, facilitating the identification of relevant relationships and causal links, thereby enhancing the accuracy and generalizability of the research findings [34].

3.2. Case Selection

This study follows the principle of theoretical sampling in case selection, focusing primarily on two cases: the China Automotive Operating System open-source project and the FastCAE open-source domestic CAE software integrated development platform. The case selection is based on three criteria. First, case representativeness: both cases prominently reflect collaboration between physical enterprises and diverse actors such as open-source communities, and each holds a certain degree of influence in its respective field. Second, content relevance: the two cases differ in their open-source strategies, complementary collaboration behaviors, and resulting impacts, allowing for meaningful comparative analyses. Third, data accessibility: given that the overall market scale of Chinese industrial software projects is relatively limited, these two cases offer advantageous conditions for obtaining the necessary materials and data for this study.

3.3. Data Collection and Analysis

The data collection and analysis in this study were conducted in three stages, as outlined below:

(1) Preliminary preparation stage. Since 2023, the author has engaged in extensive literature reading on "open source" and "industrial software" and has published related articles, gaining deeper insights into the field. For example, based on bibliometric methods, the authors analyzed 138 articles in the field of management science published between 1999 and 2022. Using a combination of quantitative and qualitative approaches, they conducted an in-depth examination of the current state, conceptual essence, theoretical breakthroughs, and theoretical evolution of open-source software under mainstream research perspectives from both cooperation and competition viewpoints (Chen & Zhou, 2024) [35]. In addition, the author collaborated with the research team to draft policy advisory reports on industrial software, analyzing the current development status, characteristics, and existing challenges of China's industrial software, and recognizing the crucial role of open-source collaboration in promoting China's industrial software innovation catch-up.

(2) Field research stage. Formal data collection began in July 2024, with literature, company profiles, and other secondary materials as the main data sources. Keywords such as "open source," "industrial software," "complementary collaboration," and "digital–real integration" guided the extensive literature review and data collation.

Primary data—such as interview transcripts and participatory observation records—were used to corroborate secondary data sources, with data triangulation employed to enhance the reliability and validity of the research. For instance, in October 2024, the author attended an open-source academic conference at the Shanghai University of International Business and Economics; in December 2024, an open-source and AI conference in Guangzhou; and, in July 2025, the OpenAtom Open-Source Ecology Conference in Beijing. These events focused on collecting case evidence of the impact of open source on industrial software enterprises. Additionally, interviews were conducted with more than ten industry experts, gathering relevant transcripts and further case data.

(3) Data analysis stage. Rather than using coding software such as NVivo for structured coding, the study applied a theory-guided inductive logic for manual synthesis, drawing on the researcher's understanding of theoretical literature, official documents, interviews, and other sources to gradually construct a conceptual framework. This approach was chosen because the study is exploratory in nature, with the aim being not to exhaustively code all case materials or conduct frequency analysis, but rather to explore—through continuous dialogue and abstraction among theory, literature, cases, and observational data—how open-source collaboration promotes industrial software innovation catch-up from a digital–real integration perspective. The theoretical construction process adhered to a balance of flexibility and coherence, respecting the boundaries of the case materials while employing abductive reasoning to drive the formation of the theoretical framework.

4. Analytical Framework Based on Case Studies

4.1. Open-Source Collaboration Strategy

Guided by corporate strategy, enterprises determine whether to collaborate with open-source communities and how extensively they seek to shape or influence these communities. On the one hand, because OSS communities are inherently free, open, and collaborative, enterprises typically have a strong willingness to actively engage with and influence these communities. On the other hand, from a managerial perspective, although enterprises participating in OSS communities do not own these communities, this does not prevent them from joining and conducting collaborative software R&D projects. This study argues that enterprises participate in OSS communities and engage in complementary collaboration primarily on the basis of an open-source collaboration strategy. Depending on the mode of influence and degree of formality, such strategies can be categorized as either formal or informal. The former mainly refers to coordinating mechanisms related to rules, behavioral outcomes, and processes, thereby shaping complementary collaboration behaviors for both parties. A key advantage of this approach is that work objectives and performance requirements are relatively explicit, facilitating supervision and evaluation. In contrast, informal collaboration strategies leverage intangible resources such as culture, values, and interpersonal relationships—utilizing social pressure, role modeling, and intrinsic motivation to guide behaviors in collaborative projects. Drawing on existing research, this study divides formal collaboration strategies into behavioral coordination and outcome coordination, whereas informal collaboration encompasses relationship coordination and empowerment coordination [36]. A detailed analysis is provided below:

First, behavioral coordination. This refers to an enterprise establishing clear plans, processes, and standards to regulate and manage individual or collective behaviors, thereby ensuring the achievement of objectives [37]. Such a coordination strategy can significantly influence behavior by, for example, reducing errors and improving work quality through detailed rules and standards. It can also standardize the innovation process to ensure effective implementation of innovative projects and risk control. When enterprises cooperate with virtual open-source communities to develop projects, both sides may jointly formulate rules and standards for code contributions, project management, and communication, thereby ensuring explicit behavioral norms for both parties during collaboration. For instance, Red Hat, by providing comprehensive documentation and online training courses [38], helps community developers become familiar with the development processes and quality requirements of its open-source projects, guiding complementary collaboration behaviors for both sides.

Second, outcome coordination. This involves managing and motivating individuals and teams by setting clear objectives and evaluating performance outcomes. Typically, outcome coordination strategies focus on result, relying heavily on explicit definitions, specified performance indicators, and the measurement and feedback of results. Through reviewing completed work, enterprises can effectively offer corrective measures and improvements, thereby ensuring that collaborative projects are delivered on schedule [39]. Meanwhile, such strategies also help enhance efficiency in selecting, planning, and executing tasks, improving the likelihood of achieving performance targets. In enterprise collaboration with a virtual open-source community, result coordination is mainly reflected in the following ways: (1) setting explicit collaborative objectives and performance indicators to ensure that both parties' complementary behaviors proceed with a clear focus and standardized guidelines. For example, when collaborating with Hyperledger Fabric, IBM aims to develop enterprise-ready solutions that overcome existing technological limitations in areas such as privacy, confidentiality, auditability, performance, and measurability [40]. (2) Providing timely feedback and evaluation: based on the project's phased progress, adjustments are made to critical actions during collaboration. (3) Implementing appropriate incentive mechanisms. Under Huawei's leadership, the OpenEuler open-source community attracted participation from more than 1300 leading enterprises, research institutions, and universities. Among these participants, Tongtech donated its proprietary mature software R&D platform and the latest results in cloud-native technology [41], significantly empowering the community. Leveraging this platform has simultaneously amplified the company's market influence, thus creating a win–win outcome for both the physical enterprise and the virtual open-source community.

Third, relationship coordination. This refers to influencing and managing behaviors by establishing and maintaining strong cooperative relationships; it relies on informal mechanisms such as trust, commitment, reciprocity, and long-term collaboration rather than on formal rules. Stressing the role of social capital and network relationships in cooperation, this approach seeks to achieve shared objectives by emphasizing effective relationship management. Existing research has shown that social ties between partners play a crucial role in joint tasks and exert a more significant effect on collaborative behaviors [42]. When a physical enterprise collaborates with a virtual open-source community, relationship coordination may shape their complementary collaboration in several ways: (1) through building trust. By maintaining transparency in the collaboration process and sharing information openly, an enterprise can strengthen and sustain its trust-based relationship with an open-source community, thereby increasing community participation and contributions. (2) By promoting open communication, information can be promptly circulated between the enterprise and the open-source community, thereby enhancing project coordination efficiency. (3) By reinforcing cooperative commitment: through ongoing resource input and support—including, but not limited to, technical and human resources—the enterprise demonstrates its long-term dedication to open-source community collaboration.

Fourth, empowerment coordination. This involves granting individuals or teams increased autonomy and decision-making authority as a means of facilitating collaboration.

Such a coordination strategy places emphasis on encouragement and support, enabling organizational members to exercise greater independence and creativity in accomplishing tasks and achieving objectives. Generally, empowerment coordination significantly heightens members' sense of responsibility and engagement, thereby boosting their intrinsic motivation and job satisfaction. Studies have found that empowerment can improve participants' capacity for cooperation and enhance management efficacy [43], fostering improved levels of satisfaction, adaptability, and engagement in collaboration. When enterprises collaborate with a virtual open-source community, they may grant project members more autonomy, encouraging them to set their own objectives and codes of conduct for specific tasks, thereby reinforcing complementary behaviors among participants.

In the China Automotive Operating System Open-Source Project, the open-source strategy demonstrates a dual characteristic of both formal and informal coordination. In formal coordination, behavioral coordination is reflected in the joint efforts of the China Association of Automobile Manufacturers and the OpenAtom Open-Source Foundation to build an upstream–downstream collaboration platform, foster open-source communities and independent ecosystems, and achieve industry-wide co-construction and sharing through the open-sourcing of foundational original technologies. This also drives the formulation and implementation of technical requirements and testing/evaluation standards for automotive operating systems. Outcome coordination is reflected in the rapid adaptation and deployment of project results. For example, the easyAda kernel open-sourced by Puhua Basic Software was successfully adapted to the SemiDrive G9X chip, and the "Xiaoman" secure vehicle-control operating system was adapted to multiple mainstream domestic and international MCU controllers, with proven mass production experience. In informal coordination, relationship coordination is reflected in the long-term national-level support for the construction of open-source communities and platforms for independent core technologies, emphasizing the role of domestic open-source foundations and industry organizations in ecosystem cultivation. Empowerment coordination is reflected in hosting the project under the OpenAtom Open-Source Foundation and adopting open governance mechanisms such as the Mulan Public License, providing institutional guarantees and empowerment for community participants.

In the FastCAE Project, the open-source strategy also embodies both formal and informal coordination characteristics. In formal coordination, behavioral coordination is reflected in opening up the core three-dimensional modeling and simulation platform capabilities through open source, thereby building a cross-industry, cross-disciplinary joint R&D and testing environment. This promotes full-process capability sharing from geometric modeling and mesh generation to multiphysics computation and establishes a project docking and task collaboration mechanism with upstream and downstream enterprises in the industrial chain. Outcome coordination is reflected in the platform's continuous iteration and rapid adaptation—for example, customized functional extensions have been developed for fields such as new energy vehicles, ship design, and pressure vessels, with stable deployment and performance verification achieved in multiple engineering cases. In informal coordination, relationship coordination is reflected in the establishment of long-term cooperative relationships between the core team and domestic and international universities, research institutes, and industry users, forming a stable network of users and developers through community-based operations. Empowerment coordination is reflected in the use of open-source licenses to standardize code sharing and usage, ensuring institutional rights and interests for different participants regarding intellectual property and the use of project outcomes.

4.2. Complementary Collaboration Behaviors Between Enterprises and OSS Communities

According to resource-based theory, an enterprise consists of a collection of resources that it owns or controls. The purpose of acquiring complementary resources is to integrate them into products or services to maximize returns [44]. Some scholars have further extended this view, suggesting that enterprises can transcend their boundaries to access external complementary assets—such as capabilities, knowledge, products, or services that facilitate innovation—to gain competitive advantages [45]. Drawing on existing theories and practices, this study posits that complementary collaboration behaviors between enterprises and open-source software (OSS) communities primarily focus on four areas (Figure 1).

Figure 1. Complementary collaboration behaviors between enterprises and OSS communities.

First, data-factor complementarity. This refers to partners sharing and integrating each other's data resources in order to strengthen their own innovation capabilities and competitiveness. Such complementary behavior is grounded in resource-based theory and dynamic capabilities theory. The former underscores the heterogeneity and inimitability of resources, suggesting that acquiring complementary resources through collaboration can yield a competitive advantage. The latter focuses on how enterprises integrate internal and external data to adapt to rapidly changing technological environments and challenges. For physical enterprises and virtual OSS communities, data-factor complementarity may be realized by sharing, integrating, and leveraging resources such as code, documentation, architectures, communication records, audio, video, and images. This approach significantly enhances both parties' R&D efficiency and ecosystem development.

Second, technological complementarity. This refers to different actors collaborating on technological aspects to achieve results superior to those that either party could attain independently. Butler et al. (2022) propose that, when software development faces technological bottlenecks, companies and OSS communities can offer mutual technical support [46], including but not limited to fields such as cloud computing, big data, artificial intelligence, machine learning, cybersecurity, blockchain, and the Internet of Things.

Third, institutional complementarity. This refers to collaboration between an enterprise and an OSS community in areas such as community governance, reporting structures, project management norms, and coordination mechanisms. New Institutional Economics posits that institutions play a pivotal role in economic activities and can substantially influence organizational behavior and performance; effective institutional arrangements can greatly reduce transaction costs and improve resource allocation efficiency and organizational outcomes. Because OSS communities are highly flexible in their organizational structure, project management often features certain informal elements. However, given the formality typical of corporate management, a company's participation in a community can often facilitate more standardized community governance. For instance, Microsoft plans to gradually open-source the Windows 11 user interface framework, WinUI, and will provide relevant documentation and guidance to help developers use the framework smoothly [47].

Fourth, human-resource-related complementarity. This denotes the mutual collaboration between an enterprise and an OSS community in terms of professional expertise, judgment, intelligence, and insight. Social capital theory asserts that social networks and relationships play a crucial role in the acquisition and utilization of resources. Through human resource complementarity, enterprises and communities can establish closer cooperative relationships. In OSS communities, developers often participate on a voluntary basis driven by interest or personal needs, typically dedicating spare time outside their regular employment, and the free flow of participants can sometimes disrupt development schedules. However, when enterprises join these communities for collaborative development, they effectively gain additional voluntary contributors beyond their in-house personnel, lowering both labor and financial costs. Human resource complementarity thus serves as a critical two-way driver of collaboration between physical enterprises and virtual OSS communities.

In the China Automotive Operating System Open-Source Project, regarding the complementary behaviors between enterprises and the community, data-factor complementarity is reflected in the sharing of testing and evaluation standards and the unification of industry data interfaces, thereby providing consistent reference benchmarks for upstream and downstream enterprises in the industrial chain. Technological complementarity is achieved through cross-platform adaptation between kernels and chips, as well as between operating systems and controllers, along with the integration of mass production experience from multiple parties, accelerating the engineering implementation of key technologies. Institutional complementarity is embodied in the joint formulation of industry technical standards and testing specifications, as well as the use of open-source license authorization mechanisms to standardize the management of code sharing and application. Human resource complementarity is realized by integrating the professional technical teams of OEMs such as FAW, SAIC, BAIC, Geely, and Li Auto, along with chip and tool vendors such as SemiDrive and Horizon Robotics, enabling cross-enterprise R&D resource sharing and knowledge complementarity.

In the FastCAE Project, regarding the complementary behaviors between enterprises and the community, data-factor complementarity is mainly reflected in the introduction of industry-shared test case libraries, material databases, and geometric model resources into the simulation platform, thereby reducing the cost of model preparation and verification in the early stages for all parties. Technological complementarity is demonstrated through cross-platform compatibility, the coupling of different solvers, and unification of secondary development interfaces, creating conditions for customized applications by users in different fields. Institutional complementarity is reflected in community-built development specifications, code submission and review mechanisms, as well as the alignment of simulation standards promoted by industry associations. Human resource complementarity is realized by integrating development and application experts from software companies, equipment manufacturers, and research institutions, who jointly participate in platform feature development, bug fixing, and performance optimization.

4.3. Performance of Complementary Collaboration

The performance of complementary collaboration between enterprises and OSS communities can be evaluated from multiple perspectives. This study categorizes performance into the following aspects: (1) R&D efficiency. This mainly refers to the ability of developers to efficiently conduct software development, function improvement, and defect correction with limited time and resources by leveraging collaborative sharing. Software development often encounters challenges related to development cycles; enterprises operating for commercial gain frequently favor closed-source, commercial software, but the top-down, centralized nature of closed-source models may significantly prolong the innovation cycle [48]. Compared to conventional closed-source software development, OSS community–based collaborative projects exhibit higher R&D efficiency in terms of reduced development cycles, more frequent contributions, and rapid response to issues. (2) R&D quality. This centers on the quality of products generated in software R&D, including functionality completeness, code maintainability, system stability, and security. Open-source software seamlessly embeds user innovation into the development process, enabling product designs to better reflect and fulfill user requirements [49]. Because the development process of collaborative projects is fully transparent, any issues that arise can be promptly detected and effectively addressed. As Raymond famously noted, "Given enough eyeballs, all bugs are shallow." (3) Technological innovation. In OSS projects, technological innovation encompasses not only the invention or breakthrough of entirely new technologies but also improvements and optimizations of existing technologies. Such innovation may involve introducing new algorithms, protocols, or architectures; employing new innovation tools (e.g., automated testing tools, continuous integration systems); and utilizing asynchronous communication tools to enable collaboration across time zones, among other possibilities. (4) Market Innovation. Market innovation refers to a project's capacity to expand markets, improve user experience, or create commercial value by introducing novel market models, business models, or user-engagement practices. Compared with technological innovation, market innovation is more focused on market acceptance of open-source software, commercial model innovation, and strategies for attracting and retaining users and developers.

In the China Automotive Operating System Open-Source Project, in terms of the impacts of complementary behaviors, R&D efficiency improvement is reflected in the reduction of repetitive R&D costs through unified standards, while kernel–hardware adaptation enables the rapid deployment of existing applications. R&D quality enhancement relies on the "mass production experience + community iteration" model, which ensures that code quality and system security are fully validated and continuously optimized. Technological innovation emerges from cross-industry collaboration between automakers and chip manufacturers, accelerating breakthroughs in domestic hardware–software integration in areas such as intelligent driving and smart cockpits, as well as promoting the transformation of original kernel technologies from R&D to commercial use. Market innovation is achieved through the construction of an indigenous automotive operating system ecosystem, which not only enhances the market competitiveness of domestic enterprises but also drives the large-scale adoption of domestic hardware and software in mass-produced vehicles, opening up new business models and market opportunities.

In the FastCAE Project, in terms of the impacts of complementary behaviors, R&D efficiency improvement is realized as the open-source community reduces time spent on repetitive development and model building, significantly shortening the cycle from new feature development to application. R&D quality benefits from a dual mechanism of "community feedback + engineering verification," which continuously optimizes the stability and accuracy of the software. Technological innovation is driven by the introduction of new capabilities such as multi-physics coupling, parallel computing, and cloud-based simulation, enabling breakthroughs on domestic simulation platforms in high-performance computing and complex engineering problem solving. Market innovation is achieved through the formation of a domestic simulation software ecosystem centered on FastCAE, which not only reduces the industry's dependence on foreign commercial software but also gives rise to commercialized services and supporting products for niche markets, providing new pathways for domestic industrial software to compete in the international market.

4.4. Analytical Framework for Complementary Collaboration in Industrial Software Innovation Catch-Up Under the Digital–Real Integration Approach

Building on the above theoretical proposition, this study constructs an analytical framework to illustrate the mechanisms through which open-source collaboration strategies, mediated by complementary collaboration behaviors, influence collaborative performance in the context of industrial software innovation catch-up under digital–real integration. In the context of digital–real integration, these two cases demonstrate that open-source collaboration strategies in industrial software projects can, through multi-dimensional linkages of formal coordination (behavioral coordination, outcome coordination) and informal coordination (relationship coordination, empowerment coordination), foster high-level complementary cooperation between enterprises and communities in terms of data, technology, institutions, and human resources. These complementary behaviors enhance the efficiency and quality of R&D and application, accelerate the iteration and implementation of new technologies, and promote market-oriented applications and ecosystem development. As a result, they significantly improve the performance of complementary cooperation, achieving a positive cycle from resource integration to value creation. The conceptual analytical framework derived from the case analysis is shown in Figure 2.

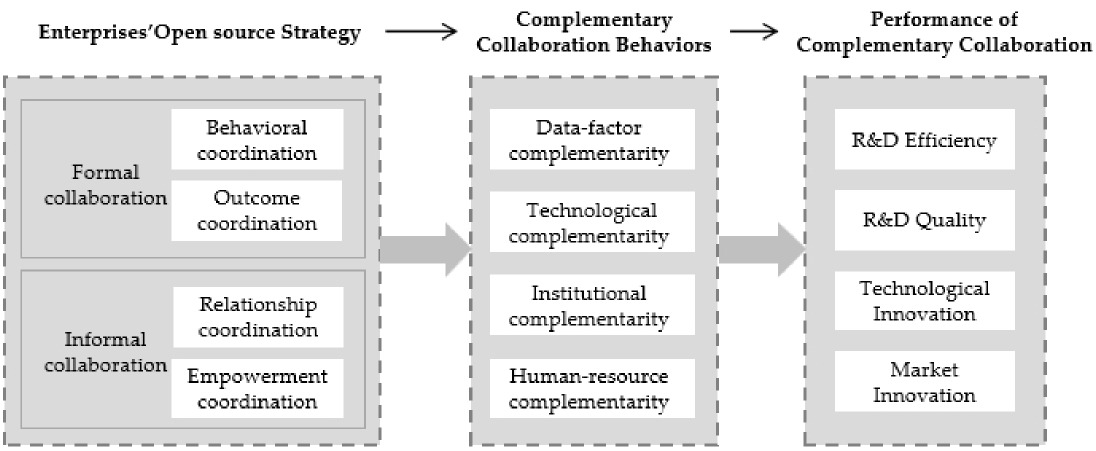

Figure 2. Analytical framework for complementary collaboration in industrial software innovation catch-up under the digital–real integration approach.

On the one hand, the open-source collaboration strategy within digital-real integration—employed by the enterprise and the OSS community—impacts their complementary collaboration behaviors. Classic strategic management theory posits that enterprises do not merely rely on internal decisions for success but also shape behaviors through interacting and cooperating with external stakeholders, leveraging resource allocation, and building relationship networks.

On the other hand, complementary collaboration behaviors between enterprises and OSS communities influence innovation performance in joint projects. When an enterprise participates in an OSS community and collaborates on development, whether in the form of shared code, documentation, audio/video resources, frameworks, technology, project standards, or human resources, it significantly shortens the development cycle and improves R&D efficiency. Furthermore, complementary collaboration between enterprises and virtual OSS communities often manifests as multiple concurrent projects, creating a multi-node, parallel coordination network structure. Such a setup markedly enhances the network's robustness, and implementing security measures at key nodes can substantially raise the overall security of the collaborative project [50]. For instance, in Ant Group's Kata Containers project, the enterprise engaged in the OSS community and established a Vulnerability Management Team (VMT) to provide security isolation, fault isolation, and performance isolation using lightweight virtualization technologies. The project lead noted: "What makes me most proud of Kata Containers is its ability, through the open-source community, to galvanize other major stakeholders and raise the overall security standard across the cloud-native landscape, harnessing open collaboration to enhance security performance." Thus, complementary collaboration between OSS communities and enterprises can enhance R&D quality, including security features of collaborative projects.

5. Conclusions and Implications

5.1. Research Conclusions

This study addresses the research question of how open-source collaboration mechanisms, under the lens of digital–real integration, influence industrial software innovation catch-up. It found that enterprise open-source collaboration strategies are the key determinants of complementary collaboration between physical enterprises and virtual open-source communities. These strategies, manifested as formal coordination (behavioral coordination and outcome coordination) and informal coordination (relationship coordination and empowerment coordination), shape the scope, intensity, and governance of joint activities.

Such strategies influence the development of four types of complementary behaviors: data complementarity, technological complementarity, institutional complementarity, and human resource complementarity. These complementarities constitute the operational pathways through which digital–real integration advances innovation. Strengthening these pathways significantly improves R&D efficiency, R&D quality, technological innovation capacity, and market innovation outcomes, forming a coherent mechanism in which strategy drives behavior and behavior enhances performance, thereby enabling industrial software innovation catch-up.

These findings provide a clear theoretical basis for deriving targeted and actionable recommendations to guide enterprises and communities in optimizing collaboration for sustained innovation catch-up.

5.2. Practical Implications

The findings of this research carry important implications for enterprises, open-source communities, and policymakers seeking to improve the performance of industrial software innovation through digital–real integration. From a strategic perspective, enterprises should match their choice of coordination modes to the stage of the project, the maturity of the technology, and the characteristics of the community. Projects in the early stages of development or those operating under high levels of uncertainty may benefit from a greater emphasis on informal coordination, which facilitates trust-building and rapid iteration, whereas more mature, market-oriented projects require stronger formal coordination to ensure alignment, quality assurance, and predictability. A dynamic adjustment of the balance between formal and informal coordination over the course of collaboration can yield optimal results.

The enhancement of data complementarity should be treated as a foundational task. This involves not only expanding the volume of shareable non-sensitive data but also improving its quality through classification, grading, and standardization, and establishing clear rules for ownership, licensing, and use. Such measures enable seamless integration across organizational boundaries and increase the economic value of data assets. Similarly, technological complementarity can be deepened by selecting high-potential innovation domains, establishing joint research and testing pipelines, and aligning enterprise technical roadmaps with community development priorities to ensure mutual reinforcement. Institutional complementarity requires the alignment of contribution agreements, intellectual property policies, and compliance mechanisms, reconciling corporate governance requirements with community norms. This reduces friction, strengthens trust, and facilitates sustained collaboration. Human resource complementarity can be strengthened through cross-training programs, short-term exchanges, and joint workshops that promote mutual understanding and skill transfer between enterprise engineers and community contributors, thereby building a sustainable talent base for the long term.

Finally, the integration of risk management into all stages of the collaboration process is essential. This includes the identification and mitigation of risks related to open-source compliance, dependence on external contributors, and the potential disruption of supply chains. By systematically aligning strategic coordination choices with the targeted enhancement of data, technological, institutional, and human resource complementarities, stakeholders can transform open-source collaboration from a loosely organized interaction into a deliberate, structured mechanism for achieving industrial software innovation catch-up in the era of digital–real integration.

5.3. Research Limitations

As an exploratory case-based analysis, this study is constrained by two main limitations. First, the proprietary nature and limited transparency of China's industrial software sector restricted access to comprehensive and longitudinal data, which may affect the breadth and generalizability of the findings. Second, the proposed conceptual framework has not been quantitatively validated, leaving the identified causal relationships theoretically grounded but empirically untested. Future research could broaden the case scope, including cross-country comparisons, as well as operationalizing the framework for quantitative testing to strengthen its explanatory and predictive power.

Author Contributions: Conceptualization, X.C.; methodology, X.C. and Y.L.; software, X.C.; validation, X.C.; formal analysis, X.C., Q.Z. and Y.L.; investigation, X.C. and Q.Z.; resources, X.C., Q.Z. and Y.L.; data curation, X.C., Q.Z. and Y.L.; writing—original draft preparation, X.C.; writing—review and editing, X.C., Q.Z. and Y.L.; visualization, X.C., Q.Z. and Y.L.; supervision, Q.Z. and Y.L.; project administration, X.C.; funding acquisition, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by the Humanities and Social Science Youth Fund of Ministry of Education of China (Grant: 23YJC630014); the Project of the State Administration of Foreign Experts Affairs of Ministry of Human Resources and Social Security of China (Grant: H20240930); and the Fundamental research Funds for the Central Universities (FRF-TP-25-049).

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huang, R.; Wang, H. How is the development of industrial digital-real integration progressing?—Evidence from China's cultural industries. *Humanit. Soc. Sci. Commun.* **2025**, *12*, 427. [CrossRef]
2. Zhao, X.F.; Zhang, Y.M. Continuously expanding the breadth and depth of digital-real integration. *Economic Daily*, 2 April 2024. Available online: http://theory.people.com.cn/n1/2024/0402/c40531-40208206.html (accessed on 29 May 2025).
3. Bukht, R.; Heeks, R. Defining, conceptualising and measuring the digital economy. *Dev. Inform. Work. Pap.* **2017**, *68*. [CrossRef]
4. Pirounakis, N.G. Aspects of the 'Real' Economy. In *The Greek Economy: Past, Present and Future*; Pal-grave Macmillan: London, UK, 1997; pp. 170–205.
5. Baldwin, C.; Von Hippel, E. Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. *Organ. Sci.* **2011**, *22*, 1399–1417. [CrossRef]

6. Fitzgerald, M.; Kruschwitz, N.; Bonnet, D.; Welch, M. Embracing digital technology: A new strategic imperative. *MIT Sloan Manag. Rev.* **2014**, *55*, 1–12.
7. Chen, X.H.; Zhou, Y. Research on the open source software innovation policy and governance in the digital economy era. *Sci. Manag. Res.* **2022**, *40*, 16–23.
8. German, D.M. The GNOME project: A case study of open source, global software development. *Softw. Process Improv. Pract.* **2003**, *8*, 201–215. [CrossRef]
9. Xu, Z. The Mysteries of Large Language Models: Tracing the Evolution of Transparency for OpenAI's GPT Models. Ph.D Thesis, Wellesley College, Wellesley, MA, USA, 2024.
10. Zhou, X.; Xiao, Y.; Yang, X. Industrial software innovation breakthrough path and incentive mechanism for domestic replacement. *Sci. Technol. Rev.* **2023**, *41*, 34–46.
11. Barney, J.B.; Clark, D.N. *Resource-Based Theory: Creating and Sustaining Competitive Advantage*; Oxford University Press: Oxford, UK, 2007.
12. Haefliger, S.; Von Krogh, G.; Spaeth, S. Code reuse in open source software. *Manag. Sci.* **2008**, *54*, 180–193. [CrossRef]
13. Dahlander, L.; Wallin, M. A man on the inside: Unlocking communities as complementary assets. *Res. Policy* **2006**, *35*, 1243–1259. [CrossRef]
14. Teece, D.J. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Res. Policy* **1986**, *15*, 285–305. [CrossRef]
15. Monteiro, F.R.; Pereira, P.A.; Cordeiro, L.C.; Filho, C.F.F.C.; Costa, M. Complementary training programme for electrical and computer engineering students through an industrial-academic collaboration. In Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 12–15 October 2016; IEEE: New York, NY, USA, 2016; pp. 1–9.
16. Kula, R.G.; Hata, H.; Matsumoto, K. FLOSS!= GitHub: A Case Study of Linux/BSD Perceptions from Microsoft's Acquisition of GitHub. *arXiv* **2021**, arXiv:2102.01325.
17. Zhou, T.; Yuan, Q. Examining users' contribution in open source software communities. *J. Comput. Inf. Syst.* **2023**, *63*, 1382–1393. [CrossRef]
18. Amit, R.; Zott, C. Value creation in e-business. *Strateg. Manag. J.* **2001**, *22*, 493–520. [CrossRef]
19. Okong'o, W.; Ndiege, J.R.A. Knowledge sharing in open-source software development communities: A review and synthesis. *VINE J. Inf. Knowl. Manag. Syst.* **2023**, *55*, 622–649. [CrossRef]
20. Koo, H.M.; Ko, I.Y. An analysis of problem-solving patterns in open source software. *Int. J. Softw. Eng. Knowl. Eng.* **2015**, *25*, 1077–1103. [CrossRef]
21. Chen, G.; Wei, J.; Li, T. Open source community: Research context, knowledge framework and research prospects. *Foreign Econ. Manag.* **2021**, *43*, 84–102.
22. Wei, J.; Chen, G. How to innovate with open source communities: Based on isomorphism—spawned cognitive legitimacy. *Stud. Sci. Sci.* **2021**, *39*, 1860–1869.
23. Wang, Y.; Chen, Y.; Koo, B. Open to your rival: Competition between open source and proprietary software under indirect network effects. *J. Manag. Inf. Syst.* **2020**, *37*, 1128–1154. [CrossRef]
24. Shaikh, M.; Levina, N. Selecting an open innovation community as an alliance partner: Looking for healthy communities and ecosystems. *Res. Policy* **2019**, *48*, 103766. [CrossRef]
25. Alami, A.; Pardo, R.; Cohn, M.L.; Wasowski, A. Pull request governance in open source communities. *IEEE Trans. Softw. Eng.* **2021**, *48*, 4838–4856. [CrossRef]
26. Porter, M.E. Technology and competitive advantage. *J. Bus. Strategy* **1985**, *5*, 60–78. [CrossRef]
27. Qian, F.; Hong, J.; Fang, T.; She, Y. Global value chain embeddedness and innovation efficiency in China. *Technol. Anal. Strateg. Manag.* **2022**, *34*, 1050–1064. [CrossRef]
28. Morgan, L.; Finnegan, P. Beyond free software: An exploration of the business value of strategic open source. *J. Strateg. Inf. Syst.* **2014**, *23*, 226–238. [CrossRef]
29. Kochhar, P.S.; Kalliamvakou, E.; Nagappan, N.; Zimmermann, T.; Bird, C. Moving from closed to open source: Observations from six transitioned projects to GitHub. *IEEE Trans. Softw. Eng.* **2019**, *47*, 1838–1856. [CrossRef]
30. Dahlander, L.; Magnusson, M. How do firms make use of open source communities? *Long Range Plan.* **2008**, *41*, 629–649. [CrossRef]
31. Parmentier, G. How to innovate with a brand community. *J. Eng. Technol. Manag.* **2015**, *37*, 78–89. [CrossRef]
32. Ocasio, W. Towards an attention-based view of the firm. *Strateg. Manag. J.* **1997**, *18*, 187–206. [CrossRef]
33. Eisenhardt, K.M. Building theories from case study research. *Acad. Manag. Rev.* **1989**, *14*, 532–550. [CrossRef]
34. Yu, C.; Xu, H.; Wang, Y.J. Research on innovation diffusion mechanism of platform ecosystem: Comparative analysis between COSMOPlat and iFLYTEK Platform. *Nankai Bus. Rev.* **2023**, *26*, 15–29.
35. Chen, X.H.; Zhou, Y. The essence of innovative collaboration and theoretical evolution of open source software from cooperation and competition perspectives. *Sci. Sci. Manag. S. T.* **2024**, *45*, 1243–1259.

36. McCarthy, S.; O'Raghallaigh, P.; Li, Y.; Adam, F. Control enactment in context: Understanding the interaction of controlee and controller perceptions in inter-organisational project teams. *Inf. Syst. J.* **2023**, *33*, 1029–1084. [CrossRef]
37. Keil, M.; Rai, A.; Liu, S. How user risk and requirements risk moderate the effects of formal and informal control on the process performance of IT projects. *Eur. J. Inf. Syst.* **2013**, *22*, 650–672. [CrossRef]
38. Ghori, A. *Red Hat® Certified Technician & Engineer (RHCT and RHCE) Training Guide and Administrator's Reference*; Endeavor Technologies Inc.: Saint Charles, IL, USA, 2009.
39. Wang, C.; Brunswicker, S.; Majchrzak, A. Knowledge search breadth and depth and OI projects performance: A moderated mediation model of control mechanism. *J. Knowl. Manag.* **2021**, *25*, 847–870. [CrossRef]
40. Bhuvana, R.; Aithal, P.S. Blockchain based service: A case study on IBM blockchain services & hyperledger fabric. *Int. J. Case Stud. Bus. IT Educ.* **2020**, *4*, 94–102.
41. Securities Times. Donating to the openEuler Community and Launching New Open-Source Products, TongTech Actively Contributes to the openEuler Ecosystem. 17 December 2023. Available online: https://baijiahao.baidu.com/s?id=1785527184042498920&wfr=spider&for=pc (accessed on 13 June 2025).
42. Dale, R.; Marshall-Pescini, S.; Range, F. What matters for cooperation? The importance of social relationship over cognition. *Sci. Rep.* **2020**, *10*, 11778. [CrossRef] [PubMed]
43. Kala'Padang, F.I.; Daromes, F.E. Participation decision making, psychological empowerment, job relevant in-formation and managerial performance. *Din. Akunt. Keuang. Dan Perbank.* **2023**, *12*, 104–119.
44. Barney, J.B. Organizational culture: Can it be a source of sustained competitive advantage? *Acad. Manag. Rev.* **1986**, *11*, 656–665. [CrossRef]
45. Carmona-Lavado, A.; Cuevas-Rodríguez, G.; Cabello-Medina, C. Does open innovation always work? The role of complementary assets. *Technol. Forecast. Soc. Change* **2021**, *162*, 120316. [CrossRef]
46. Butler, S.; Gamalielsson, J.; Lundell, B.; Brax, C.; Mattsson, A.; Gustavsson, T.; Feist, J.; Kvarnström, B.; Lönroth, E. Considerations and challenges for the adoption of open source components in software-intensive businesses. *J. Syst. Softw.* **2022**, *186*, 111152. [CrossRef]
47. Microsoft is advancing the open-sourcing of WinUI in four phases to enhance development transparency and community collaboration. *Zhongguancun Online*, 5 August 2025. Available online: https://news.zol.com.cn/1026/10261983.html (accessed on 19 June 2025).
48. Boadu, F.; Du, Y.; Xie, Y.; Fokuo, E.D. Is the correlation between knowledge sharing and firm innovation performance contingent on network trust and hierarchical culture? Evidence from the Chinese high-tech sector. *Int. J. Technol. Manag.* **2023**, *92*, 206–228. [CrossRef]
49. Zhang, N.; Zhao, W.; Pang, Z.; He, L.; Cheng, X.; Fan, W. Leveraging user ideas for product innovation in open innovation communities: A study of two stages of the idea adoption. *J. Glob. Inf. Manag.* **2023**, *31*, 1–30. [CrossRef]
50. Zhang, X.; Lei, S.; Sun, J.; Kou, W. Robustness of Multi-Project Knowledge Collaboration Network in Open Source Community. *Entropy* **2023**, *25*, 108. [CrossRef] [PubMed]