Analyzing and Evaluating today's Power of Open Source: The Open Source Value Canvas

Barbara Steffen
TU Dortmund University, Germany, barbara.steffen@tu-dortmund.de

Frederik Möller
TU Dortmund University, Germany;Fraunhofer ISST, Germany, frederik.moeller@tu-dortmund.de

Analyzing and Evaluating today's Power of Open Source: The Open Source Value Canvas

Barbara Steffen[1] and Frederik Möller[1,2]

[1] TU Dortmund University, Chair for Industrial Information Management, Dortmund, Germany
{barbara.steffen, frederik.moeller}@tu-dortmund.de
[2] Fraunhofer ISST, Dortmund, Germany
{frederik.moeller}@isst.fraunhofer.de

Abstract. The drastically progressing digitalization of society and economy shines a new light on the open-source paradigm. Previously, open-source was merely a developer paradigm to share code openly and make it available to others. However, given the need for innovation and optimization, companies can leverage open-source components to use out of the box, build services on top, or replace commodifiable services. Subsequently, there is great potential to create new value in companies using open-source components. To assist companies and researchers in achieving this, the paper presents the Open Source Value Canvas for companies' collaborative and interdisciplinary identification of open-source value. It particularly aims at analyzing and aligning the open-source potentials from the business and IT perspectives. We draw on rich insights from an ongoing research project providing tailored open-source components for the European logistics sector.

Keywords: Open Source Strategy, Open Source Integration, Open Source Value Canvas, Stakeholder Alignment

1 Introduction

The continuously progressing digitalization of society and economy poses new challenges for companies [1], such as identifying and designing new business opportunities based on data and, in general, tackling digital transformation [2]. One way to generate digital innovation is using *open-source* (OS) components that can be used *out-of-the-box*, to replace commodifiable services or build new services on top [3–5]. Generally, OS refers to all software components with an open code base and allows for inspection of source codes [6]. Today, 90% of software integrates OS components somehow [7], leading companies to be "(…) both heavy users of OSS products and contributors to their development" [6 p. 649]. Fuerstenau et al. [8] report on the utilization power of OS and the corresponding crowd knowledge of a community exemplified through the case of *Otto.de*. The company made parts of its platform open to leverage external input through third-party developers. That case is a prime example of OS as a vehicle to transcend organizational boundaries, leverage knowledge, share costs, and improve existing products and services [9].

17th International Conference on Wirtschaftsinformatik,
February 2022, Nürnberg, Germany

However, there are a plethora of determinants for companies to decide when, how, and from whom one should use OS components (e.g., in terms of governance and approval processes [10]). These include, e.g., assessing the need for OS as well as trustworthiness and reputation of the OS component's community based on its activities and quality [9, 11, 12]. Considering these aspects is vital when using OS components, given the risk of potentially integrating OS components of sub-par quality, reducing the overall quality of a company's offers/software solutions [13].

Today, integrating OS components is not a singular issue of a company's IT department as it has strategic business implications. From a company's management perspective, the cost-benefit ratio of OS integration must be analyzed and evaluated to ensure, e.g., quality/offer enhancements and/or internal cost/effort reductions compared to sole in-house solutions. Only this way it can be ensured that the company's why, it's long-term goals, are aligned with its move towards OS [14]. Thus, the aim is to identify hybrid potentials, a mixture of OS use and in-house developments, e.g., development alleviations, business model integration, or continuous innovation [15, 16]. Identifying and assessing these potentials requires an interdisciplinary approach [3, 16–18].

This paper designs and validates the *Open Source Value Canvas (OSVC),* which analyzes and assesses OS potentials from multiple perspectives. We chose a *visual inquiry tool* (also referred to as canvas, the term we use in this paper) as an artifact to guide the *interdisciplinary* and innovative OS strategy endeavor. A canvas is a structured and collaborative tool to design something of interest (e.g., a business model in the *Business Model Canvas (BMC)* [19, 20]) based on pre-defined building-blocks [21]. It is characterized by a dedicated interdisciplinary approach, which should generate a shared understanding of a visualized problem domain and enable its users to produce solutions creatively [22, 23]. Recently, scholars have proposed canvases in multiple fields, e.g., to find business model potentials based on data (e.g., see [24]), to design research (e.g., see [25, 26]), or to support ideation (e.g., see [27]) and platform design [28]. Given the new opportunities and risks surrounding OS, we see the design of the OSVC for shared innovation as an endeavor with high merit to assist practitioners and researchers in conceptualizing and assessing the potentials of OS integration.

Given that *canvases* are artifacts (e.g., see [19]), we develop the OSVC in a *Design Science Research (DSR)* study based on Hevner's three-cycle view [29] embedded in a project setting. The OSVC addresses our project, an ongoing large-scale OS-driven project in the European logistics industry, facing the challenge of assisting logistics companies in finding the opportunities and risks of OS usage and integration. The project develops a repository of dedicated OS components for the European logistics industry to foster industry-wide innovation, generate standards, ensure compatibility, and allows logistics companies to enter the corresponding platform economy. Today, almost no logistics-specific OS components exist and thus this project opens a new strategic opportunity addressing the companies' core competencies. So, the OSVC supports companies new to the business strategic use of OS to identify OS potentials and select the most suitable OS components. Thus, this paper pursues the following research question:

Research Question: How to guide companies in defining OS strategies generating value from OS integration by identifying and selecting suitable OS components?

The paper's structure: In the next section, we briefly introduce the value mechanisms of open-source. In Section 3, we outline our research approach. Section 4 shows the final result of our paper, the OSVC, its validation, and its discussion. Section 5 finalizes the paper with its contributions, limitations, and avenues for further research.

2 Open Source

Fundamentally, OS refers to the source code of software being open, giving others the means to inspect it [6, 30]. Perhaps the most famous OS license is the General Public License (GPL), used, for example, by Linux, which waives typical copyright restrictions regarding the use of the source codes and derivates with the restriction that they, in turn, must also be published under the GPL license [31]. In contrast, proprietary solutions only enable the use of software as an executable 'black box' program [11]. The proprietary software is then licensed to the customer, with the company usually providing updates and maintenance [32]. OS is highly different from 'closed-source' or proprietary software solutions [33]. For example, OS can be a vehicle for high-quality standards, yet, it also includes projects with little, none, or sub-par documentation, complicating the work on and with the OS software [33].

Often, the OS paradigm is conflated with free software, such as 'shareware' or 'freeware' [34]. Subsequently, OS might be required to be 'open', but it does not have to be free of charge, leading various scholars to analyze business models and commercialization of OS [34, 35]. For example, Okoli & Nguyen [35] propose eight FOSS business models based on a Delphi study, including *Advertising*, *Memberships/Donations*, or *Update Subscriptions*. *Free*, in that regard, means the freedom to run, redistribute, modify, and redistribute modified versions of the source code [36]. Subsequently, OS has many implications for companies, as it offers a variety of potential advantages. For example, opening access to 3rd party developers could leverage external innovation to provide continuous innovation [37] and ensure better security given the codebase's disclosure [38]. While the decision to go 'open-source' is riddled with various design options, advantages include faster adoption of users or a potentially emerging active community that contributes to product improvement [39].

Interestingly, OS is understood and perceived quite differently from, e.g., a company's business and IT perspectives. Both views emphasize different opportunities and threats and thus normally do not reach a shared OS understanding. In the following, we briefly address the OS benefits, risks, and common misunderstandings based on project observations and experiences and conversations with IT and OS experts.

Generally, OS is perceived as readily available and integrateable software components [40]. Compared to sole in-house developments, it is viewed as the lever to reduce the need for in-house expertise and thus as an outsource mechanism, to reduce development times and costs and ease continuous innovation [41, 42]. In addition, the source code is generally of great quality due to high OS community standards [41, 43]. The OS code can be viewed and thus must be "beautiful", well tested (ideally by a significant user group of which all can report and propose bug fixes) [40, 42], and easily understandable to allow for OS integration and modifications. Here, OS experts underlined that OS components are only valuable if they fulfill these criteria.

Otherwise, too many modifications and too long OS code familiarization periods are necessary as it gets harder to understand and contribute to OS projects the later one enters the OS projects. Further, the independence of individual vendors and thus the easy modular integration and modification are important assets [40–43]. That enables know-how transfers and collaboration on a global scale, making business models possible the companies could not have developed independently [41].

However, OS usage and contribution also come with risks [43]. Users of OS components do not get warranty rights, guarantees regarding the promised component functionalities, and long-term maintenance and support services [41–43]. Here, IT experts emphasized the dependency on in-house experts to assess the quality of the OS components and select and integrate the most suitable ones. Additionally, they need to be aware of all potential risks and ensure that all services can be compensated internally in case of stopped community activity. Thus, OS usage does not make in-house expertise redundant but instead requires companies to employ (rare) excellent IT experts who act as IT alignment experts and cover all important IT functions from development and operations to (continuous) deployment [41, 42]. To achieve this, IT employees are already OS experts, receive the required training, or are supported by consultants. All three options are scarce, costly, and enter a dependency [42, 43]).

To conclude, OS usage can be very valuable, but its integration is never free [44]. With the right in-house expertise, OS usage and contribution are great additions to the internal workforce and a competitive solution to commercial software products [44]. However, if this in-house expertise is missing, companies are entering a very risky dependency for which they cannot compensate. Generally, business experts tend to overestimate the quality and benefits of the OS component usage while overlooking the abovementioned in-house tasks and responsibilities. Thus, the envisioned cost savings and reduced development times are typically smaller than expected [44].

Nevertheless, OS usage can complement a company's current offers and innovation potentials [40]. Many IT experts favor OS integration due to the OS components' high quality and significant customization potentials. These provide them with great latitudes compared to commercial software use. To achieve a more realistic OS understanding and guide companies in their OS evaluation and potentially integration, the OSVC acts as an educator and aligner of the company's different stakeholders. Its main focus lies on aligning the business and IT perspectives. This support is particularly relevant if companies are considering OS for the first time or are new to its strategic lever and thus interdisciplinary context. The first OS integration steps could be the hybrid development model or hybrid business model. Here, companies can use the OSVC to design the in-house and OS capability alignment [15].

3 Research Design

We pursue a DSR approach as our intended result is an artifact (i.e., a model) [45, 46]. It enables users to understand a complex issue through abstraction and focuses on the most relevant parts of a problem under investigation [46]. Our research emerges from a case in the European logistics industry that explicitly focuses on providing OS components to logistics companies (these include, e.g., *logistics service providers*). The

adoption, use, and commercialization of logistics-specific OS components are recognized as essential but are still a field left untapped (based on our experience working in the project environment). Due to the missing and too rudimentary understanding of OS potentials and risks (in company practice), we derive the necessity and requirements for a corresponding canvas support enabling interdisciplinary (e.g., IT and business) and creative collaboration on an ill-structured problem [21, 23].

We follow Hevner's three-cycle view as it is highly applicable for creating novel artifacts and has been applied in a recent paper developing a canvas (e.g., see [25]). The framework demands cycling between the relevance, design, and rigor cycles, enabling us to incorporate, e.g., case findings, practitioners, experts, literature, and theory.

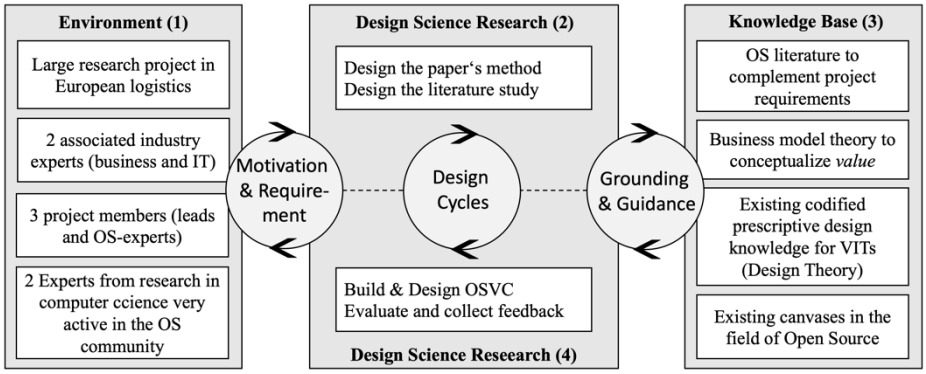

Figure 1. Research Method based on Hevner's three cycle view [29]

In terms of the **Environment (1)**, our motivation stems from a large European project in logistics as outlined above. As we are part of the project, we could derive the need for such a canvas through interactions (i.e., informal talks, regular meetings in different combinations, project documentation, and presentations) with project leads, project members, and associated industry partners from the logistics industry. Notably, the goal is to generate an intuitive tool that assists practitioners in decision-making regarding whether OS should be integrated and, if so, which OS components to integrate. To accommodate for the interdisciplinarity of strategically leveraging OS components, we also included external experts from computer science to provide a distinctive IT-heavy perspective covering technological aspects.

In the **Design Science Research (2)** phase, we designed the paper's method and aim and its literature review study to develop the first version of the OSVC. While the literature review is not meant to be comprehensive, our goal was to identify initial building blocks to complement the requirements we faced in our project setting. We used these initial building blocks as a basis for discussion amongst the authors and to facilitate a shared understanding of the OS domain. These initial building blocks served as a starting point to iteratively develop the building blocks of the final version.

The **Knowledge Base (3)** for our design is fourfold. First, we collected insights from the existing literature corpus searching for papers referring to the use of OS components or OS business models and the associated value. For that, we collected papers from *AISeL*, where we searched in *Titles* and *Abstracts* for "Open Source" with "Business

Model" and "Value" respectively. We searched for papers referring to OS value dimensions, for instance, in terms of their business model (e.g., [35]) or general business value (e.g., [47]) and constructed a preliminary concept matrix [48]. These keywords helped us to identify business motivations for using OS software and components. The search produced a total of 84 papers, which we reduced to a sample of 32 papers for deeper analysis [48]. This literature review only covers the IS domain as it is just intended to complement our case and workshop design iterations. However, an in-depth analysis of the literature corpus and the collection of additional literature from other databases (e.g., IEEE) is a task yet to do.

Second, since we strive to realize business value, we adopt the *business model* as a theoretical lens [49] to integrate and align our findings into the design project. In particular, we draw typical business model concepts, such as the *value proposition*, *value assessment*, and *value capture,* and combine them with our OS-specific findings to ensure that the OS strategy fits the company's overall strategy (see Section 4.1).

Third, we use existing prescriptive design knowledge codified as design principles in a design theory for Visual Inquiry Tools (canvases) by Avdiji et al. [21, 50]. Using existing design knowledge is paramount in design projects to reduce redundancy and create successful designs [51]. Here, we reuse the design principles for canvases by instantiating them in our canvas [52].

Fourth, a Google Search identified four existing canvases covering OS. All four were included in the concept matrix. The analysis showed that none of the canvases explicitly questions and challenges the use of OS nor addresses OS components' value-oriented utilization, thus supporting our motivation to develop the OSVC.

In the second **Design Science Research (4)** phase, we iteratively improve our initial design of the OSVC and its concept table consisting of guiding questions for each of the OSVC's 11 building blocks (see **Figure 2** and **Table 1)** via four workshops: 1. two leaders of the project, 2. two project members, 3. two OS experts and 4. two representatives from a logistics leader (business and IT expert). In IS research and DSR, workshops are valuable to collect insights from practitioners in an interactive setting, e.g., for the evaluation and iteration of a new artifact's design [53, 54].

Based on the knowledge base's findings, e.g., the literature and our case experiences, we designed the initial version of the OSVC. Then, we evaluated and iterated the OSVC (design) in the successive workshops [53]. Reasons for selecting the four workshops' participants: First and second, the project leads/members could challenge the OSVC from their project expertise covering industry needs, requirements, and the difficulty to communicate the OS value potentials to users. Third, the IT (especially OS) experts ensure a distinct and realistic integration of the IT perspective covering, e.g., a developer's typical decision-making process when selecting and participating in OS communities/components. Lastly, the industry experts used the OSVC in the intended interdisciplinary setting allowing us to collect sophisticated industry feedback.

4 The Open Source Value Canvas

In this section, we introduce the OSVC. It assists users in designing creative solutions on how to leverage the potentials of OS and prepare for and reduce potential pitfalls.

4.1 Open Source Value Canvas Perspective

The *OSVC* is structured in 11 building blocks. Together they design a company's intended OS strategy and support the selection of the OS components based on their quality and applicability. The building blocks can be assigned to three inductively derived strategic company-level categories using existing terminology from business model research. This is valuable to demonstrate the interdependence of company strategy and OS usage. These are:

- **Value Proposition:** Describes the company's underlying motivation and usage options of OS in potential business models and their envisioned benefits [55].
- **Value Assessment:** Describes important considerations to ensure an educated evaluation and assessment of OS projects and components for potential use.
- **Value Capture:** Describes the incurred costs and monetarization potentials and thus direct risks and benefits of OS [56].

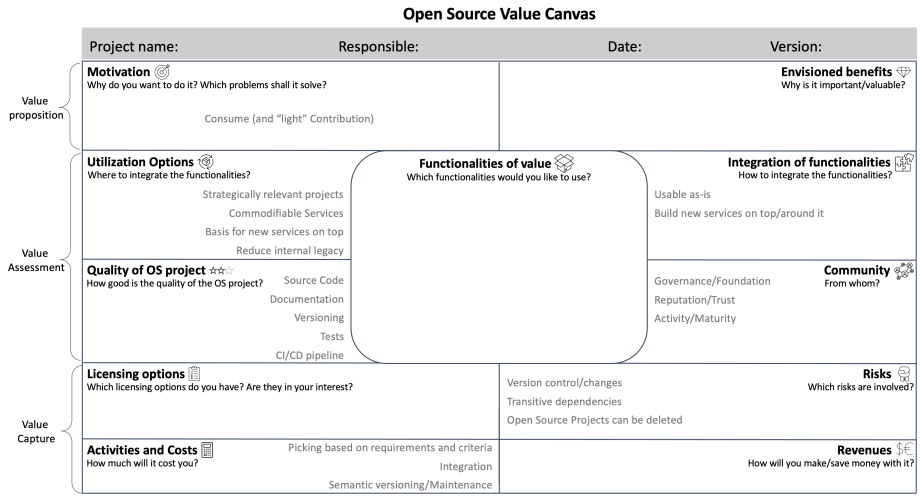

Figure 2. The final version of the OSVC

Most of the building blocks are further refined via two to five relevant subcategories (see **Figure 2** and **Table 1**). Drawing from the design principles of Avdiji et al. [21] and complementing the OSVC, we derived 'directions for use' [21] to ease the utilization of the canvas. We codified these directions in guiding questions supporting users to apply the OSVC correctly in the following concept table. We derived the guiding questions from the building blocks' overarching theme and each workshops' comments, insights, and reflections. Subsequently, the guiding questions are an iterative design artifact that emerged and matured over time.

4.2 Value Proposition

The *Value Proposition* covers two building blocks addressing the company's overall motivation and envisioned benefits when integrating OS (components).

Motivation: It represents the company's underlying motivation to use OS (components) and/or contribute to OS projects. Generally, users of the OSVC should think about 'why' and 'what' to operate in terms of OS. Workshop participants named several reasons: the software snippet already exists, the codebase remains open, or the goal is to integrate or leverage an existing OS community. Complementarily, the literature names the engagement in open collaboration and innovation through OS, such as sharing resources, leveraging external ideas, saving costs, or synergizing expertise [4, 9]. Thus, these benefits even lead to advantages for light contributors. They can influence further developments and features of the OS component and have a better and more detailed understanding of the component and potential usages.

Envisioned Benefits: It requires the users to define OS integration benefits which later enable them to objectively measure whether the OS usage was a success. These benefits can range from cost-free and maintained availability of the codebase to quick integration of existing components. These might reduce development costs or provide access to external innovation outside of the company's boundaries [57].

4.3 Value Assessment

The *Value Assessment* consists of five building blocks that support the user in assessing and selecting OS components based on several criteria. Three of the five building blocks focus on choosing, using, and integrating OS components, including assessing *how to utilize an OS component*, what *functionalities are of value* to the company, and *how to integrate* it properly. The other two building blocks guide the user in assessing the *quality of the OS project* and the *community* built around it.

Functionalities of Value: It emphasizes identifying OS projects and/or OS component functionalities that add value to a company's processes, products, and services. These can range from domain-specific services to general applications (e.g., OS databases). In our project services such as *track and trace* are useful OS components. It is a commodified service for which customers are unwilling to pay.

Utilization Options: It depicts utilization options for OS components. The underlying question is 'what to do' with the OS component. The dominant and most direct utilization option is integrating existing OS components into software during development. The company should consider internal reasons for integrating OS and its benefits for replacing currently used software and/or developing new software.

Integration of Functionalities: Next, the question is 'where' to integrate the OS component. Where does it add value? And how easy is it to integrate? The user must assess the OS component's integratability and analyze the cost-benefit ratio surrounding it. The main question is whether it costs or gains time for the developers to integrate it vs. developing the functionality themselves. For example, it might be necessary to hire full time employees responsible for managing OS tasks.

Quality of OS Project: A significant issue in selecting OS components for reuse and integration is the OS project's quality. Here, the participants of the workshops highlighted that they usually screen an OS project regarding its promised functionality (e.g., does it solve the problem) and the quality based on different metrics, such as the star rating on *GitHub,* the reliability, number of committers, the ratio between open vs. closed tickets, and the project fork frequency. Other metrics are, arguably, hard to

assess, e.g., the quality of code or documentation. Evaluating the quality of OS might be more complicated than that of proprietary software [38].

Community: A central element of OS components is the community they originate from or are embedded in. Assessing an OS community from the company's point of view should include the activity/maturity and trustworthiness of the community's governance. For example, a common pitfall of using OS components is that the OS project can disappear or be abandoned [58]. However, usually, it is possible to continue working with the source code as-is, leaving maintenance and development to the users. An active community is also the nexus for continuous innovation in OS projects [16].

4.4 Value Capture

The category *Value Capture* encompasses four building blocks that support the user in assessing an OS component's usage rights, risks, costs, and benefits. That step is particularly important to assess the business fit of a component, especially in relation to the already defined value proposition (see Section 4.2).

Licensing: Licensing is an essential part of OS distribution and settles the use of OS components [17]. Various licenses are available, coming with different rights, obligations, and prohibitions that require compliance, as there can be legal consequences [7, 17, 59]. For example, the *copyleft* license requires the user of OS components to adhere to the same license agreement that the initial OS components hold (e.g., GNU) [34, 58]. The careful analysis of the license (options) is essential. One industry professional told us that some licenses are simply out of bounds, as they cannot be used 'freely.' This is echoed in the literature as 'unacceptable license clauses' [59].

Risks: Leveraging OS components comes across a variety of risks. For example, while OS gives companies a reasonable road to acquire new software components, relying on various projects can potentially lead to *transitive dependencies* that entangle dependencies connected to root libraries in a hierarchy of dependencies [60]. Other risks include that OS projects can be abandoned or discontinued, meaning that the user must secure the capabilities to control and maintain the OS component itself. Another issue is that users must ensure compliance with license agreements and prevent the illegal use of OS components and concerns about liability.

Activities and Costs: OS components are available cost-free; nevertheless, selecting, integrating, and maintaining these components in the company's products/ services takes time and costs resources of the IT department. Also, one of our workshop participants recalled an unsuccessful bid to establish an OS budget for contributing to the available OS projects and becoming part of the community.

Revenues: Using OS only makes business sense if the company earns or saves money with the OS-enabled products or services [55] or benefits from long-term innovation potentials. If a company wishes to generate revenue based on an OS component (instead of 'only' saving costs), outlining potential revenue models is essential. Naturally, that also includes questions, such as deciding on whether part of the software built with an OS component will also be available as an OS component/project. If so, Rajala [34] outlines various OS revenue models, such as *Loss-leader*, *Accessorizing*, or *Support Selling*. If the software is offered proprietarily, typical revenue models of software applications apply.

Table 1: Guiding Questions to use the OSVC

Building Block		Guiding questions
Value Proposition	Motivation	▪ Why do you want to use open source? ▪ Which problems shall it solve (e.g., missing expertise/resources)? ▪ Why do you want to contribute to existing OS projects (e.g., setting own standards)?
	Envisioned Benefits	▪ Which benefits do you gain (e.g., reduced time, saved costs, quality)? ▪ Why is it valuable to you?
Value Assessment	Functions of Value	▪ Which functionalities do you want to use?

	Utilization Options	▪ Where do you integrate the functionalities (e.g., in products)?
		▪ Are these utilization options strategically relevant?
		  ▪ Does it support strategic projects?
		  ▪ Is it the basis for new offers?
		  ▪ Does it automatize commodities?
		  ▪ Does it reduce internal legacies?

	Integration of Functions	▪ How to integrate the functionalities?
		  ▪ Are they usable as they are?
		  ▪ Will you build services on top/around it (e.g. vertical integration)?

	Quality of OS project	▪ How is the quality of the functionalities?
		  ▪ How is the quality of the source code?
		  ▪ How well does it perform?
		  ▪ Is it secure/safe?
		  ▪ Are tests integrated?
		  ▪ Is continuous integration and deployment included?
		  ▪ Is the versioning transparent?
		  ▪ How well is the documentation?

	Community	▪ From whom (includes transitive dependencies of OS projects)?
		  ▪ Is the community/OS project foundation trustworthy?
		  ▪ Are there potential conflicts of interest?
		▪ How is the community governed?
		  ▪ Who decides what to integrate/accept?
		  ▪ How do they ensure quality?
		▪ Is the community active/mature?

Value Capture	Licensing Options	▪ Which licensing options do you have? ▪ Are they in your interest (e.g., utilization)?

	Risks	▪ Which risks are involved?
		  ▪ Are there dependencies from other OS projects? (transitive dependencies)?
		  ▪ Can you handle the version control/changes?
		  ▪ If the OS project gets deleted: Can you maintain/develop it?
		▪ Are you prepared to mitigate these risks?

	Activities and Costs	▪ What activities are needed to make it work?
		  ▪ Search for the needed functionalities
		  ▪ Pick OS project based on, e.g., quality & community trustworthiness
		▪ How much will it cost you?
		▪ Do you have an OS budget (e.g., invested into the used communities)?

	Revenues	▪ How will you make money with it? ▪ How will you save money with it?

4.5 Evaluation

The OSVC was evaluated at the end of all four workshops, leading to continuous in-between iterations resulting in the final version depicted in **Figure 2**. We asked all participants about their overall impression, whether the building blocks are understandable, whether they cover all essential aspects and whether building blocks and/or subcategories are superfluous. All participants liked the OSVC combined with the concept table due to their easy understandability and intuitive look and feel, allowing for fast adoption. None of the building blocks was perceived as unnecessary, nor did the participants propose additional building blocks. Regarding the OSVC's applicability, each workshop analyzed a specific OS project to use and evaluate the OSVC to challenge its structure, building blocks, and guidance. The OS scenarios demonstrated that the OSVC works, is helpful, and does not show any logical insufficiencies. The workshop participants' diversity of backgrounds (computer science, business, research, and practitioners) ensured that the business and IT perspectives on OS usage and integrations got evaluated. Here, we validated and added new input covering the necessary aspects to consider when selecting OS components.

The project internal workshops (workshops 1 and 2) highlighted the benefits of making OS tangible. This transparency eases the industry partner collaboration and education and makes a shared understanding possible. The third workshop with the two IT/OS experts detailed and complemented the building blocks by additional 12 subcategories. Here, we added, e.g., the three additional *component quality* assessment criteria "versioning", "tests" and "CI/CD pipeline" and added all subcategories of the building blocks *risks* and *activities and costs*. These additions are particularly valuable as they ensure that non-OS experts "at least" analyze these criteria in the evaluation and selection process of OS components, reducing the risks of OS usage. Especially, subcategories like "transitive dependencies" and "open source projects can be deleted" are essential to consider as OS usage leads to a component dependency. Given the worst-case scenario of stopped OS development and support, the company must cover these tasks and responsibilities internally. Thus, these criteria ensure that the OS integration does not get taken too lightly. The fourth workshop with the logistics industry's IT, and business expertise of a market leader confirmed the final subcategories.

To conclude, the OSVC was perceived as professional, understandable, and applicable. No further building block proposals were made, and the final subcategories were assessed as sufficient for the first OS usage/component screenings. The main feature was to see, design, and align the most relevant aspects of OS strategy and OS integration at one glance, ensuring cross-expertise alignment (especially of the business and IT perspectives).

5 Contributions, Limitations, and Future Work

The paper develops the OSVC to assess the business value of integrating and/or contributing to OS and selecting the most applicable OS projects for both. In the offset, we strived to design a canvas that helps interdisciplinary teams (especially focusing on

aligning the business and IT perspectives on OS) to discuss and decide on OS integration potentials regarding its fit to the company strategy.

The OSVC has a variety of contributions to research and practice. Our work provides **researchers** with a tool guiding the structured analysis and evaluation of OS components. Subsequently, researchers may use our canvas or the underlying structure of building blocks to extend, refine, or adapt them to specific industries. For example, the OSVC could be tailored explicitly for specific licensing options, industrial sectors, or technologies (e.g., OS components for Industry 4.0 or Internet of Things devices). The canvas provides an overview of the value components of OS, extending the current state of research with a tool guiding and asking for interdisciplinary collaboration to prevent one-sided OS understanding.

Practitioners can use our tool to assess the value of OS, the integration and selection of OS components, and the decision to "lightly" contribute to the project/community. A significant advantage of a canvas is the availability of condensed information on 'one sheet of paper'. Using the OSVC, practitioners will not overlook essential aspects of OS evaluation and OS component selection and integration. For example, the OSVC includes vital OS aspects, such as *OS component quality* and *licensing,* and forces users to consider the underlying *motivation*, *necessary activities*, and *associated costs*. Also, since we have deliberately looked at OS elements from an interdisciplinary point of view, meaning IT and business, the OSVC acts as a bridge between disciplines.

Naturally, our work is subject to **limitations**. First and foremost, we only considered a snapshot of the existing literature corpus to guide it in designing the OSVC rigorously. Yet, incorporating more databases and a broader literature sample could provide insights for new building blocks or further building block refinements. Additionally, we have drawn knowledge from industry professionals and computer scientists of our project environment. The integration of further expert feedback allows for assessments regarding the tool's general applicability.

Planned **future work** includes additional testing of the OSVC and continuously checking for its applicability and appropriateness of scope. For example, with more practitioners entering the project, the OSVC could be applied in more scenarios with an ever-broader field of industry experts. Correspondingly, future work should consider the triangulation of building blocks through additional research methods. While, at this time, the OSVC is built with a narrow literature sample and workshops, collecting expert feedback through in-depth interviews or even collecting feedback quantitatively through questionnaires could drastically improve the validity of each building block in the future. In addition, the OSVC is envisioned to be complemented by additional OS component evaluation support addressing the subcategories in more detail and allowing for their quantitative assessment. This shall support practitioners in making the right OS component evaluation and selection decisions. Additionally, the aim is to integrate the OSVC and the add-on analysis feature as IT tool-supported *living canvas* [61].

Acknowledgment

The project "Silicon Economy Logistics Ecosystem" is funded by the Federal Ministry of Transport and Digital Infrastructure.

References

1. Bharadwaj, A., El Sawy, O.A., Pavlou, P.A., Venkatraman, N.: Digital Business Strategy: Toward a Next Generation of Insights. MIS Quarterly: Management Information Systems, vol. 37, 471–482 (2013). doi: 10.25300/MISQ/2013/37:2.3
2. Veit, D., Clemons, E., Benlian, A., Buxmann, P., Hess, T., Kundisch, D., Leimeister, J.M., Loos, P., Spann, M.: Business Models. An Information Systems Research Agenda. Business and Information Systems Engineering, vol. 6, 45–53 (2014). doi: 10.1007/s12599-013-0308-y
3. Harutyunyan, N., Riehle, D., Sathya, G.: Industry Best Practices for Corporate Open Sourcing. In: Proceedings of the 53rd Hawaii International Conference on System Sciences (2020). doi: 10.24251/HICSS.2020.716
4. Link, G., Qureshi, S.: The Role of Open Source Communities in Development: Policy Implications for Governments. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018). doi: 10.24251/HICSS.2018.302
5. Imre, Ö.: Constructing an open source ERP Sales Pitch – In search for interpretative repertoires. In: Proceedings of the 8th Mediterranean Conference on Informaiton Systems (2014)
6. Krogh, G. von, Haefliger, S., Spaeth, S., Wallin, M.W.: Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. MIS Quarterly: Management Information Systems, vol. 36, 649–676 (2012). doi: 10.2307/41703471
7. Harutyunyan, N.: Managing Your Open Source Supply Chain-Why and How? Computer, vol. 53, 77–81 (2020). doi: 10.1109/MC.2020.2983530
8. Fuerstenau, D., Rothe, H., Baiyere, A., Schulte-Althoff, M., Masak, D., Schewina, K., Anisimova, D.: Growth, Complexity, and Generativity of Digital Platforms: The Case of Otto.de. In: Proceedings of the 40th International Conference on Information Systems (2019)
9. Morgan, L., Feller, J., Finnegan, P.: Open Source Innovation Networks: Exploring High and Low-desnity Models. In: PACIS 2012 (2012)
10. Harutyunyan, N., Riehle, D.: Industry best practices for component approval in FLOSS Governance. In: Proceedings of the EuroPloP 2020 (2020)
11. Atiq, A., Tripathi, A.: Monetary rewards for open source software developers. In: Proceedings of the 35th International Conference on Information Systems (2014)
12. Pitts, C.: How are hybrid open source firms' products commercially successful? In: Proceedings of the Southern Association for Information Systems Conference (2014)
13. Maki-Asiala, P., Matinlassi, M.: Quality Assurance of Open Source Components: Integrator Point of View. In: 30th Annual International Computer Software and Applications Conference (COMPSAC'06), vol. 2, pp. 189–194 (2006). doi: 10.1109/COMPSAC.2006.153
14. Steffen, B., Steffen, B.: Asking Why. In: 10th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2021 (2021)
15. Shaikh, M.: Digital Hybrids in Open Source: Hybridization Crises in Linux. In: Nagle, F., Wissel, J., Zaggl, M., Lifshitz-Assaf, H., Shaikh, M., Tambe, P., Wang, S. (eds.) Open Source Software Development and Organizations (2019)

16. Riehle, D.: The Commercial Open Source Business Model. Conference: Proceedings of the 15th Americas Conference on Information Systems, vol. (2009)
17. Riehle, D.: The Innovations of Open Source. Computer, vol. 52, 59–63 (2019). doi: 10.1109/MC.2019.2898163
18. Lakka, S., Varoutas, D., Martakos, D.: Impact Of Oss On Software Markets- An Evaluation. In: MCIS 2009 Proceedings (2009)
19. Osterwalder, A.: The Business Model Ontology. A Proposition in a Design Science Approach (2004)
20. Osterwalder, A., Pigneur, Y.: Business Model Generation. A Handbook for Visionaries, Game Changers, and Challengers. Wiley (2013)
21. Avdiji, H., Elikan, D.A., Missonier, S., Pigneur, Y.: A Design Theory for Visual Inquiry Tools. Journal of the Association for Information Systems, vol. 21, 695–734 (2020). doi: 10.17705/1jais.00617
22. Reijnen, C., Overbeek, S., Wijers, G., Sprokholt, A.: A Shared Vision for Digital Transformation: Codification of The Operating Model Canvas Approach. In: Proceedings of the 26th European Conference on Information Systems (2018)
23. Elikan, D.A., Pigneur, Y.: A visual inquiry tool for brand identity. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
24. Hunke, F., Seebacher, S., Thomsen, H.: Please Tell Me What to Do - Towards a Guided Orchestration of Key Activities in Data-Rich Service Systems. In: Hofmann, S., Müller, O., Rossi, M. (eds.) Designing for Digital Transformation. Co-Creating Services with Citizens and Industry, pp. 426–437. Springer International Publishing, Cham (2020)
25. Schoormann, T., Behrens, D., Fellmann, M., Knackstedt, R.: On Your Mark, Ready, Search: A Framework for Structuring Literature Search Strategies in Information Systems. In: Proceedings of the 16th International Conference on Wirtschaftsinformatik (2021)
26. Möller, F., Schoormann, T., Otto, B.: 'Caution - Principle Under Construction' - A Visual Inquiry Tool for Developing Design Principles. In: Chandra Kruse, L., Seidel, S. (ed.) DESRIST 2021, 223-235 (2021)
27. Lecuna, A., Thoring, K., Mueller, R.: The Idea Arc: Designing a Visual Canvas for Fuzzy Ideas. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019). doi: 10.24251/HICSS.2019.068
28. Steffen, B., Möller, F., Nowak, L.: Transformer(s) of the Logistics Industry - Enabling Logistics Companies to excel with Digital Platforms. In: Proceedings of the 55th Hawaii International Conference on System Sciences (2022)
29. Hevner, A.: A Three Cycle View of Design Science Research. Scandinavian journal of information systems, vol. 19, 87–92 (2007)
30. Schryen, G., Kadura, R.: Open source vs. closed source software: Towards measuring security. In: Shin, S.Y., Ossowski, S. (eds.) Proceedings of the 2009 ACM symposium on Applied Computing - SAC '09, pp. 2016–2023. ACM Press, New York, New York, USA (2009). doi: 10.1145/1529282.1529731
31. Fitzgerald, B.: The Transformation of Open Source Software. MIS Quarterly, vol. 30, 587–598 (2006). doi: 10.2307/25148740
32. Sarrab, M., Rehman, O.M.H.: Empirical study of open source software selection for adoption, based on software quality characteristics. Advances in Engineering Software, vol. 69, 1–11 (2014). doi: 10.1016/j.advengsoft.2013.12.001

33. Aberdour, M.: Achieving Quality in Open-Source Software. IEEE Softw., vol. 24, 58–64 (2007). doi: 10.1109/MS.2007.2
34. Rajala, R., Nissilä, J., Westerlund, M.: Determinants of OSS revenue model choices. In: Proceedings of the 14th European Conference on Information Systems, pp. 1839–1850 (2006)
35. Okoli, C., Nguyen, J.: Business Models for Free and Open Source Software: Insights from a Delphi Study. In: Proceedings of the 21st Americas Conference on Information Systems (2015)
36. Stallman, R.: The GNU Operating System and the Free Software Movement. In: DiBona, C., Ockman, S., Stone, M. (eds.) Open Sources. Voices from the Open Source Revolution, pp. 31–38. O'Reilly & Associates, Sebastobol (1999)
37. Tiwana, A.: Evolutionary Competition in Platform Ecosystems. Information Systems Research, vol. 26, 266–281 (2015). doi: 10.1287/isre.2015.0573
38. Asundi, J., Carare, O., Dogan, K.: Why Do Commercial Firms Open The Source Code Of Their Products? In: MCIS 2009 Proceedings (2009)
39. Riehle, D.: Single-vendor open source firms. Computer, vol. 53, 68–72 (2020)
40. Sherlock, J., Muniswamaiah, M., Clarke, L., Cicoria, S.: Open Source Software Opportunities and Risks. arXiv preprint arXiv:1812.11697, vol. (2018)
41. Morgan, L., Finnegan, P.: Beyond free software: An exploration of the business value of strategic open source. The Journal of Strategic Information Systems, vol. 23, 226–238 (2014)
42. Gentemann, L., Termer, F.: Open Source Monitor. Studienbericht 2019 (2019)
43. Renner, T., Vetter, M., Rex, S. and Kett, H.: Open Source Software. Einsatzpotentiale und Wirtschaftlichkeit. Eine Studie der Fraunhofer Gesellschaft. Stuttgart 2005, vol. (2005)
44. Ven, K., Verelst, J., Mannaert, H.: Should you adopt open source software? IEEE Softw., vol. 25, 54–59 (2008)
45. Poeppelbuss, J., Lubarski, A.: Modularity Canvas - A Framework for Visualizing Potentials of Service Modularity. In: Proceedings of the 14th International Conference on Wirtschaftsinformatik (2019)
46. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems, vol. 15, 251–266 (1995). doi: 10.1016/0167-9236(94)00041-2
47. Chengalur-Smith, I., Nevo, S., Demertzoglou, P.: An empirical analysis of the business value of open source infrastructure technologies. Journal of the Association of Information Systems, vol. 11, 3 (2010)
48. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly: Management Information Systems, vol. 26, xiii–xxiii (2002)
49. Niederman, F., March, S.: The "Theoretical Lens" Concept: We All Know What it Means, but do We All Know the Same Thing? Commun. Assoc. Inf. Syst., vol. 44, 1–33 (2019). doi: 10.17705/1CAIS.04401
50. Avdiji, H., Elikan, D.A., Missonier, S., Pigneur, Y.: Designing Tools for Collectively Solving Ill-Structured Problems. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018). doi: 10.24251/HICSS.2018.053
51. Iivari, J., Rotvit Perlt Hansen, M., Haj-Bolouri, A.: A proposal for minimum reusability evaluation of design principles. European Journal of Information Systems, vol. , 1–34 (2020). doi: 10.1080/0960085X.2020.1793697

52. Schoormann, T., Möller, F., Hansen, M.R.P.: How do Researchers (Re-)Use Design Principles: An Inductive Analysis of Cumulative Research. In: Chandra Kruse, L., Seidel, S. (ed.) DESRIST 2021, 188-194 (2021)
53. Thoring, K., Mueller, R., Badke-Schaub, P.: Workshops as a Research Method: Guidelines for Designing and Evaluating Artifacts Through Workshops Proceedings of the 53rd Hawai Conference on System Sciences. Hawaii: USA (2020). doi: 10.24251/HICSS.2020.620
54. Ørngreen, R., Levinsen, K.: Workshops as a Research Methodology. Electronic Journal of E-learning, vol. 15, 70–81 (2017)
55. Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying Business Models: Origins, Present, and Future of the Concept. Communications of the Association for Information Systems, vol. 16, 1–25 (2005). doi: 10.17705/1CAIS.01601
56. Aversa, P., Haefliger, S., Rossi, A., Baden-Fuller, C.: From business model to business modelling: Modularity and manipulation. Business Models and Modelling, vol. 33, 151–185 (2015). doi: 10.1108/S0742-332220150000033022
57. Feller, J., Finnegan, P., Hayes, J.: Open source networks: an exploration of business model and agility issues. In: Proceedings of the 14th European Conference on Information Systems (2006)
58. Colazo, J.A., Fang, Y., Neufeld, D.: Development success in open source software projects: exploring the impact of copylefted licenses. In: Proceedings of the 11th Americas Conference on Information Systems (2005)
59. Gangadharan, G.R., Paoli, S. de, D'Andrea, V., Weiss, M.: License compliance issues in free and open source software. MCIS 2008 Proceedings, vol. , 2 (2008)
60. Pashchenko, I., Plate, H., Ponta, S.E., Sabetta, A., Massacci, F.: Vulnerable open source dependencies: Counting those that matter. In: ESEM '18: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (2018)
61. Steffen, B., Möller, F., Rotgang, A., Ryan, S., Margaria, T.: Towards Living Canvases. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation, pp. 95–116. Springer International Publishing, Cham (2021)